

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Елабужский институт (филиал) федерального государственного автономного
образовательного учреждения высшего образования
«Казанский (Приволжский) федеральный университет»



УТВЕРЖДАЮ
Заместитель директора
по образовательной деятельности
И.П. Михайлова
И.П. Михайлова
МП
« 1 » марта 20 24 г.

**РАБОЧАЯ ПРОГРАММА ПРОФЕССИОНАЛЬНОГО МОДУЛЯ
ПМ.01 Разработка модулей программного обеспечения для компьютерных систем**

Рабочая программа профессионального модуля разработана в соответствии с Федеральным государственным образовательным стандартом среднего профессионального образования (далее ФГОС СПО) по специальности 09.02.07 «Информационные системы и программирование».

Разработчик(и):

Анисимова Т.И., доцент Елабужского института (филиала) Федерального государственного автономного образовательного учреждения высшего образования «Казанский (Приволжский) федеральный университет», кафедра математики и прикладной информатики

РАССМОТРЕНО:

Учебно-методическая комиссия Елабужского института КФУ

Протокол заседания УМК № 1 от «1» марта 20 24 г.

СОДЕРЖАНИЕ

- 1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**
- 2. СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**
- 3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**
- 4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**

1 ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ ПМ.01 РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ

1.1 Место профессионального модуля в программе подготовки специалистов среднего звена

Рабочая программа профессионального модуля (далее – программа ПМ) – является частью программы подготовки специалистов среднего звена в соответствии с ФГОС СПО по специальности 09.02.07 «Информационные системы и программирование» укрупненной группы направлений подготовки и специальностей 09.00.00 «Информатика и вычислительная техника».

1.2 Цели и задачи профессионального модуля (требования к результатам освоения профессионального модуля)

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся в ходе освоения ПМ.01 Разработка модулей программного обеспечения для компьютерных систем должен:

Иметь практический опыт:

- в разработке кода программного продукта на основе готовой спецификации на уровне модуля;

- использовании инструментальных средств на этапе отладки программного продукта;
- проведении тестирования программного модуля по определенному сценарию;
- использовании инструментальных средств на этапе отладки программного продукта;
- разработке мобильных приложений.

Уметь:

- осуществлять разработку кода программного модуля на языках низкого и высокого уровней;

- создавать программу по разработанному алгоритму как отдельный модуль;
- выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;
- уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства.

Знать:

- основные этапы разработки программного обеспечения;
- основные принципы технологии структурного и объектно-ориентированного программирования;
- способы оптимизации и приемы рефакторинга;
- основные принципы отладки и тестирования программных продуктов.

Требования к результатам освоения профессионального модуля:

Индекс компетенции	Расшифровка приобретаемой компетенции
--------------------	---------------------------------------

ОК 01.	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.
ОК 02.	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности
ОК 03	Планировать и реализовывать собственное профессиональное и личностное развитие
ОК 04	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста
ОК 06	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения
ОК 07	Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях
ОК 08	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
ОК 09	Использовать информационные технологии в профессиональной деятельности
ОК 10	Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 1.1	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием
ПК 1.3	Выполнять отладку программных модулей с использованием специализированных программных средств
ПК 1.4	Выполнять тестирование программных модулей
ПК 1.5	Осуществлять рефакторинг и оптимизацию программного кода
ПК 1.6	Разрабатывать модули программного обеспечения для мобильных платформ

Количество часов, отводимое на освоение профессионального модуля

всего – **933** часов, в том числе:

максимальной учебной нагрузки обучающегося – **753** часов, включая:

обязательной аудиторной учебной нагрузки обучающегося – **705** часов,

самостоятельной работы обучающегося – **23** часов;

промежуточная аттестация - **15** часов;

консультация – **10** часов;

учебной практики – **72** часов,

производственной практики – **108** часов.

2. СТРУКТУРА И СОДЕРЖАНИЕ ПМ.01 «РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ»

2.1 Учебно-тематический план профессионального модуля

Индекс общих и профессиональных компетенций	Наименования разделов профессионального модуля*	Всего часов (макс. учебная нагрузка и практики)	Объем времени, отведенный на освоение междисциплинарного курса (курсов)					Консультация	Промежуточная аттестация	Практика	
			Обязательная аудиторная учебная нагрузка обучающегося			Самостоятельная работа обучающегося				Учебная, часов	Производственная (по профилю специальности), часов
			Всего, часов	в т.ч. лабораторные работы и практические занятия, часов	в т.ч., курсовая работа (проект), часов	Всего, часов	в т.ч., курсовая работа (проект), часов				
1	2	3	4	5	6	7	8	9	10	11	12
ОК 01. ОК 02. ОК 03. ОК 04. ОК 05. ОК 06. ОК 07. ОК 08. ОК 09. ОК 10.	МДК.01.01 Разработка программных модулей	259	243	129	20	8	-	4	4	-	-

*Раздел профессионального модуля – часть примерной программы профессионального модуля, которая характеризуется логической завершенностью и направлена на освоение одной или нескольких профессиональных компетенций. Раздел профессионального модуля может состоять из междисциплинарного курса или его части и соответствующих частей учебной и производственной практик. Наименование раздела профессионального модуля должно начинаться с отлагательного существительного и отражать совокупность осваиваемых компетенций, умений и знаний.

ПК 1.1. ПК 1.2.											
ОК 01. ОК 02. ОК 03. ОК 04. ОК 05. ОК 06. ОК 07. ОК 08. ОК 09. ОК 10. ПК 1.3. ПК 1.4. ПК 1.5.	МДК.01.02 Под- держка и тестиро- вание програм- ных модулей	173	163	93	-	5	-	2	3	-	-
ОК 01. ОК 02. ОК 03. ОК 04. ОК 05. ОК 06. ОК 07. ОК 08. ОК 09. ОК 10. ПК 1.2. ПК 1.6	МДК.01.03 Разра- ботка мобильных приложений	172	163	103	-	6	-	3	-	-	-
ОК 01. ОК 02. ОК 03. ОК 04. ОК 05. ОК 06. ОК 07. ОК 08. ОК 09. ОК 10. ПК 1.2. ПК 1.3	МДК.01.04 Си- стемное програм- мирование	141	136	85	-	4	-	1	-	-	-
ОК 01. ОК 02. ОК 03. ОК 04. ОК 05. ОК 06. ОК 07. ОК 08. ОК 09. ОК 10. ПК 1.1. ПК 1.2. ПК 1.3. ПК 1.4. ПК 1.5. ПК 1.6.	Учебная практика	72	-	-	-	-	-	-	-	72	-
ОК 01. ОК 02. ОК 03. ОК 04. ОК 05. ОК 06. ОК 07. ОК 08. ОК 09. ОК 10.	Производственная практика	108									108

ПК 1.1. ПК 1.2. ПК 1.3. ПК 1.4. ПК 1.5. ПК 1.6.											
	Итого	925	705	410	20	23	-	10	7	72	108
	Экзамен по модулю	8									
	Всего по модулю	933									

2.2. Тематический план и содержание профессионального модуля (ПМ)

Наименование разделов и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся, курсовая работа (проект) (если предусмотрены)	Объем часов лек/ лаб. зан/с/р	Уровень освоения*
1	2	3	4
МДК. 01.01 Разработка программных модулей		231 (94/129/8)	
Тема 1.1 Жизненный цикл ПО	Содержание учебного материала	2 (2/0/0)	
	1 Понятие ЖЦ ПО. Этапы ЖЦ ПО.	2	1
	Самостоятельная работа обучающихся:	0	
Тема 1.2. Структурное программирование	Содержание учебного материала	41,5 (16/24/1,5)	1
	1 Технология структурного программирования.		
	2 Инструментальные средства оформления и документирования алгоритмов программ	16	1
	3 Оценка сложности алгоритма: классификация, классы алгоритмов, неразрешимые задачи		
	Лабораторные работы		
	Оценка сложности алгоритмов сортировки.		
	Оценка сложности алгоритмов поиска.	24	2
	Оценка сложности рекурсивных алгоритмов.		
	Оценка сложности эвристических алгоритмов.		
	Самостоятельная работа обучающихся		
Отчет по лабораторным работам 1. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва: ФО-РУМ: ИНФРА-М, 2022. — 400 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0812-9. - Текст: электронный. - URL: https://znanium.com/catalog/product/1794453 (дата обращения: 09.09.2023). – Режим доступа: по подписке. 3. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отрасле-вой направленности: учебное пособие / Г. Н. Федорова. — Москва: КУРС: ИНФРА-М, 2021. — 336 с. —	1,5	3	

	(Среднее профессиональное образование). - ISBN 978-5-906818-41-6. - Текст: электронный. - URL: https://znanium.com/catalog/product/1138896 (дата обращения: 09.09.2023). – Режим доступа: по подписке. 4. Исаченко, О. В. Программное обеспечение компьютерных сетей : учебное пособие / О.В. Исаченко. — 2-е изд., испр. и доп. — Москва : ИНФРА-М, 2021. — 158 с. — (Среднее профессиональное образование). - ISBN 978-5-16-015447-3. - Текст: электронный. - URL: https://znanium.com/catalog/product/1189344 (дата обращения: 09.09.2023). – Режим доступа: по подписке.		
Тема 1.3. Объектно-ориентированное программирование	Содержание учебного материала	39,5 (14/24/1,5)	
	1 Основные принципы объектно-ориентированного программирования. Классы: основные понятия. Перегрузка методов.	14	1
	2 Операции класса. Иерархия классов.		
	3 Синтаксис интерфейсов. Интерфейсы и наследование.		
	4 Структуры. Делегаты. Регулярные выражения		
	5 Коллекции. Параметризованные классы.		
	6 Указатели. Операции со списками		
	Лабораторные работы	24	2
	1. Работа с классами. 2. Перегрузка методов. 3. Определение операций в классе. 4. Создание наследованных классов 5. Работа с объектами через интерфейсы. 6. Использование стандартных интерфейсов. 7. Работа с типом данных структура. 8. Коллекции. Параметризованные классы. 9. Использование регулярных выражений 10. Операции со списками.	24	2
	Самостоятельная работа обучающихся	1,5	3
Отчет по лабораторным работам 1. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва: ФО-РУМ: ИНФРА-М, 2022. — 400 с. — (Среднее профессиональное образование). – ISBN 978-5-8199-0812-9. – Текст: электронный. – URL: https://znanium.com/catalog/product/1794453 (дата обращения: 09.09.2023).	1,5	3	

	<p>– Режим доступа: по подписке.</p> <p>3. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: учебное пособие / Г. Н. Федорова. — Москва: КУРС: ИНФРА-М, 2021. — 336 с. — (Среднее профессиональное образование). — ISBN 978-5-906818-41-6. — Текст: электронный. — URL: https://znanium.com/catalog/product/1138896 (дата обращения: 09.09.2023). — Режим доступа: по подписке.</p> <p>4. Исаченко, О. В. Программное обеспечение компьютерных сетей: учебное пособие / О.В. Исаченко. — 2-е изд., испр. И доп. — Москва: ИНФРА-М, 2021. — 158 с. — (Среднее профессиональное образование). — ISBN 978-5-16-015447-3. — Текст: электронный. — URL: https://znanium.com/catalog/product/1189344 (дата обращения: 09.09.2023). — Режим доступа: по подписке.</p>		
Тема 1.4. Паттерны проектирования	Содержание учебного материала	32 (10/21/1)	
	1 Назначение и виды паттернов.	10	
	2 Основные шаблоны.		
	3 Порождающие шаблоны.		1
	4 Структурные шаблоны.		
	5 Поведенческие шаблоны.		
	Лабораторные работы	21	2
	1. Использование основных шаблонов. 2. Использование порождающих шаблонов. 3. Использование структурных шаблонов. 4. Использование поведенческих шаблонов.	21	2
	Самостоятельная работа обучающихся	1	3
	Отчет по лабораторным работам 1. Гагарина, Л. Г. Технология разработки программного обеспечения: учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва: ФО-РУМ: ИНФРА-М, 2022. — 400 с. — (Среднее профессиональное образование). — ISBN 978-5-8199-0812-9. — Текст: электронный. — URL: https://znanium.com/catalog/product/1794453 (дата обращения: 09.09.2023). — Режим доступа: по подписке. 3. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой	1	3

	<p>направленности: учебное пособие / Г. Н. Федорова. — Москва: КУРС: ИНФРА-М, 2021. — 336 с. — (Среднее профессиональное образование). — ISBN 978-5-906818-41-6. — Текст: электронный. — URL: https://znanium.com/catalog/product/1138896 (дата обращения: 09.09.2023). — Режим доступа: по подписке.</p> <p>4. Исаченко, О. В. Программное обеспечение компьютерных сетей : учебное пособие / О.В. Исаченко. — 2-е изд., испр. И доп. — Москва : ИНФРА-М, 2021. — 158 с. — (Среднее профессиональное образование). — ISBN 978-5-16-015447-3. — Текст: электронный. — URL: https://znanium.com/catalog/product/1189344 (дата обращения: 09.09.2023). — Режим доступа: по подписке.</p>		
Тема 1.5. Событийно-управляемое программирование	Содержание учебного материала	43 (18/24/1)	
	1 Событийно-управляемое программирование		1
	2 Элементы управления. Диалоговые окна. Обработчики событий.	18	
	3 Введение в графику		
	Лабораторные работы	24	2
	1. Разработка приложения с использованием текстовых компонентов 2. Разработка приложения с несколькими формами. 3. Разработка приложения с не визуальными компонентами. 4. Разработка игрового приложения. 5. Разработка приложения с анимацией.	24	2
	Самостоятельная работа обучающихся	1	3
<p>Отчет по лабораторным работам</p> <p>1. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва: ФО-РУМ: ИНФРА-М, 2022. — 400 с. — (Среднее профессиональное образование). — ISBN 978-5-8199-0812-9. — Текст: электронный. — URL: https://znanium.com/catalog/product/1794453 (дата обращения: 09.09.2023). — Режим доступа: по подписке.</p> <p>3. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отрасле-вой направленности: учебное пособие / Г. Н. Федорова. — Москва: КУРС: ИНФРА-М, 2021. — 336 с. — (Среднее профессиональное образование). — ISBN 978-5-906818-41-6. — Текст: электронный. — URL: https://znanium.com/catalog/product/1138896 (дата обращения: 09.09.2023). — Режим доступа: по подписке.</p> <p>4. Исаченко, О. В. Программное обеспечение компьютерных сетей : учебное пособие / О.В. Исаченко. — 2-е изд., испр. И доп. — Москва : ИНФРА-М, 2021. — 158 с. — (Среднее профессиональное образование). — ISBN 978-5-16-015447-3. — Текст: электронный. — URL:</p>	1	3	

	https://znanium.com/catalog/product/1189344 (дата обращения: 09.09.2023). – Режим доступа: по подписке.		
Тема 1.6. Оптимизация и рефакторинг кода	Содержание учебного материала	43 (18/24/1)	
	1 Методы оптимизации программного кода.	18	1
	2 Цели и методы рефакторинга.		
	Лабораторные работы	24	2
	1. Оптимизация и рефакторинг кода	24	
	Самостоятельная работа обучающихся	1	3
Отчет по лабораторным работам 1. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва: ФО-РУМ: ИНФРА-М, 2022. — 400 с. — (Среднее профессиональное образование). — ISBN 978-5-8199-0812-9. — Текст: электронный. — URL: https://znanium.com/catalog/product/1794453 (дата обращения: 09.09.2023). — Режим доступа: по подписке. 3. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отрасле-вой направленности: учебное пособие / Г. Н. Федорова. — Москва: КУРС: ИНФРА-М, 2021. — 336 с. — (Среднее профессиональное образование). — ISBN 978-5-906818-41-6. — Текст: электронный. — URL: https://znanium.com/catalog/product/1138896 (дата обращения: 09.09.2023). — Режим доступа: по подписке. 4. Исаченко, О. В. Программное обеспечение компьютерных сетей : учебное пособие / О.В. Исаченко. — 2-е изд., испр. И доп. — Москва : ИНФРА-М, 2021. — 158 с. — (Среднее профессиональное образование). — ISBN 978-5-16-015447-3. — Текст: электронный. — URL: https://znanium.com/catalog/product/1189344 (дата обращения: 09.09.2023). — Режим доступа: по подписке.	1	3	
Тема 1.7 Разработка пользовательского интерфейса.	Содержание учебного материала	15 (8/6/1)	
	1 Правила разработки интерфейсов пользователя.	8	1
	Лабораторные работы	6	2
	1. Разработка интерфейса пользователя.	6	
Самостоятельная работа обучающихся	1	3	

	<p>Отчет по лабораторным работам</p> <p>1. Гагарина, Л. Г. Технология разработки программного обеспечения: учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва: ФО-РУМ: ИНФРА-М, 2022. — 400 с. — (Среднее профессиональное образование). — ISBN 978-5-8199-0812-9. — Текст: электронный. — URL: https://znanium.com/catalog/product/1794453 (дата обращения: 09.09.2023). — Режим доступа: по подписке.</p> <p>3. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отрасле-вой направленности: учебное пособие / Г. Н. Федорова. — Москва: КУРС: ИНФРА-М, 2021. — 336 с. — (Среднее профессиональное образование). — ISBN 978-5-906818-41-6. — Текст: электронный. — URL: https://znanium.com/catalog/product/1138896 (дата обращения: 09.09.2023). — Режим доступа: по подписке.</p> <p>4. Исаченко, О. В. Программное обеспечение компьютерных сетей : учебное пособие / О.В. Исаченко. — 2-е изд., испр. И доп. — Москва : ИНФРА-М, 2021. — 158 с. — (Среднее профессиональное образование). — ISBN 978-5-16-015447-3. — Текст: электронный. — URL: https://znanium.com/catalog/product/1189344 (дата обращения: 09.09.2023). — Режим доступа: по подписке.</p>	1	3
Тема 1.8 Основы ADO.Net	Содержание учебного материала	15 (8/6/1)	
	1 Работа с базами данных	8	1
	2 Доступ к данным		
	3 Создание таблицы, работа с записями.		
	4 Способы создания команд		
	Лабораторные работы	6	2
	1. Создание приложения с БД 2. Создание запросов к БД 3. Создание хранимых процедур	6	2
	Самостоятельная работа обучающихся	1	3
	<p>Отчет по лабораторным работам</p> <p>1. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва: ФО-РУМ: ИНФРА-М, 2022. — 400 с. — (Среднее профессиональное образование). — ISBN 978-5-8199-0812-9. — Текст: электронный. — URL: https://znanium.com/catalog/product/1794453 (дата обращения: 09.09.2023).</p>	1	3

	<p>– Режим доступа: по подписке.</p> <p>3. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отрасле-вой направленности: учебное пособие / Г. Н. Федорова. — Москва: КУРС: ИНФРА-М, 2021. — 336 с. — (Среднее профессиональное образование). — ISBN 978-5-906818-41-6. — Текст: электронный. — URL: https://znanium.com/catalog/product/1138896 (дата обращения: 09.09.2023). — Режим доступа: по подписке.</p> <p>4. Исаченко, О. В. Программное обеспечение компьютерных сетей : учебное пособие / О.В. Исаченко. — 2-е изд., испр. И доп. — Москва : ИНФРА-М, 2021. — 158 с. — (Среднее профессиональное образование). — ISBN 978-5-16-015447-3. — Текст: электронный. — URL: https://znanium.com/catalog/product/1189344 (дата обращения: 09.09.2023). — Режим доступа: по подписке.</p>		
Итого		231	
<p>Курсовой проект в 5 семестре Тематика курсовых проектов: Автоматизация деятельности учебной части Автоматизация работы салона видеопроката Автоматизация работы физиотерапевтического отделения поликлиники Автосправочник автомобилей АРМ библиотекаря АРМ диспетчера такси АРМ диспетчера трамвайного депо АРМ документооборота библиотеки АРМ заведующей детского сада АРМ кассира АРМ кассира парикмахерской АРМ кассира техникума АРМ компании грузоперевозок АРМ менеджера по продаже мобильных телефонов АРМ продавца АРМ продавца магазина офисной техники АРМ продажи билетов в ЖД кассе АРМ работника аэропорта АРМ работника зоомагазина</p>		20	

АРМ работника кадрового агентства				
АРМ работника СТО				
АРМ работы кассира в кинотеатре				
АРМ секретаря научно-производственного центра				
АРМ слесаря СТО				
АРМ сотрудника упаковочного цеха завода КАЛИБР				
АРМ учета выполненных работ сотрудника фирмы по грузоперевозкам				
АРМ учета оборудования				
Консультация		4		
Экзамен в 5, дифференцированный зачет в 4 семестре		4		
Всего по модулю		259		
МДК.01.02 Поддержка и тестирование программных модулей		168 (70/93/5)		
Тема 2.1 Отладка и тестирование программного обеспечения	Содержание учебного материала		118 (46/69/3)	
	1	Тестирование как часть процесса верификации программного обеспечения.	46	1
	2	Виды ошибок. Методы отладки.		
	3	Методы тестирования.		
	4	Классификация тестирования по уровням.		
	5	Тестирование производительности		
	6	Регрессионное тестирование.		
	Лабораторные работы		69	2
	1.	Тестирование «белым ящиком»	69	2
	2.	Тестирование «черным ящиком»		
3.	Модульное тестирование			
4.	Интеграционное тестирование			
Самостоятельная работа обучающихся		3	3	

	<p>Отчет по лабораторным работам</p> <p>1. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва: ФО-РУМ: ИНФРА-М, 2022. — 400 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0812-9. - Текст: электронный. - URL: https://znanium.com/catalog/product/1794453 (дата обращения: 09.09.2023). – Режим доступа: по подписке.</p> <p>3. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отрасле-вой направленности: учебное пособие / Г. Н. Федорова. — Москва: КУРС: ИНФРА-М, 2021. — 336 с. — (Среднее профессиональное образование). - ISBN 978-5-906818-41-6. - Текст: электронный. - URL: https://znanium.com/catalog/product/1138896 (дата обращения: 09.09.2023). – Режим доступа: по подписке.</p> <p>4. Исаченко, О. В. Программное обеспечение компьютерных сетей : учебное пособие / О.В. Исаченко. — 2-е изд., испр. и доп. — Москва : ИНФРА-М, 2021. — 158 с. — (Среднее профессиональное образование). - ISBN 978-5-16-015447-3. - Текст: электронный. - URL: https://znanium.com/catalog/product/1189344 (дата обращения: 09.09.2023). – Режим доступа: по подписке.</p>	3	3	
Тема 2.2 Документирование	Содержание учебного материала	50 (24/24/2)		
	1	Средства разработки технической документации. Технологии разработки документов.		
	2	Документирование программного обеспечения в соответствии с Единой системой программной документации.	24	1
	3	Автоматизация разработки технической документации Автоматизированные средства оформления документации		
	Лабораторные работы		24	2
	Оформление документации на программные средства с использованием инструментальных средств.		24	2
	Самостоятельная работа обучающихся		2	3
	<p>Отчет по лабораторным работам</p> <p>1. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва: ФО-РУМ: ИНФРА-М, 2022. — 400 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0812-9. - Текст: электронный. - URL: https://znanium.com/catalog/product/1794453 (дата обращения: 09.09.2023). – Режим доступа: по подписке.</p> <p>3. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отрасле-вой направленности: учебное пособие / Г. Н. Федорова. — Москва: КУРС: ИНФРА-М, 2021. — 336 с. —</p>	2	3	

	(Среднее профессиональное образование). - ISBN 978-5-906818-41-6. - Текст: электронный. - URL: https://znanium.com/catalog/product/1138896 (дата обращения: 09.09.2023). – Режим доступа: по подписке. 4. Исаченко, О. В. Программное обеспечение компьютерных сетей : учебное пособие / О.В. Исаченко. — 2-е изд., испр. и доп. — Москва : ИНФРА-М, 2021. — 158 с. — (Среднее профессиональное образование). - ISBN 978-5-16-015447-3. - Текст: электронный. - URL: https://znanium.com/catalog/product/1189344 (дата обращения: 09.09.202). – Режим доступа: по подписке.		
Итого		168	
Консультация		2	
Экзамен в 5 семестре		3	
Всего по модулю		173	
МДК.01.03 Разработка мобильных приложений		169 (60/103/6)	
Тема 3.1 Основные платформы и языки разработки мобильных приложений	Содержание учебного материала	87 (34/51/2)	
	1 Основные платформы мобильных приложений, сравнительная характеристика		
	2 Нативные приложения, веб-приложения, гибридные и кроссплатформенные приложения, их области применения	34	1
	3 Основные языки для разработки мобильных приложений (Java, Objective-C и др.)		
	4 Инструменты разработки мобильных приложений (JDK/ AndroidStudio/ WebView/ Phonegap и др.)		
	Лабораторные работы	51	2
	1. Установка инструментария и настройка среды для разработки мобильных приложений 2. Установка среды разработки мобильных приложений с применением виртуальной машины	51	2
	Самостоятельная работа обучающихся	2	3
	Отчет по лабораторным работам 1. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва: ФО-РУМ: ИНФРА-М, 2022. — 400 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0812-9. - Текст: электронный. - URL: https://znanium.com/catalog/product/1794453 (дата обращения: 09.09.2023).	2	3

	<p>– Режим доступа: по подписке.</p> <p>3. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отрасле-вой направленности: учебное пособие / Г. Н. Федорова. — Москва: КУРС: ИНФРА-М, 2021. — 336 с. — (Среднее профессиональное образование). - ISBN 978-5-906818-41-6. - Текст: электронный. - URL: https://znanium.com/catalog/product/1138896 (дата обращения: 09.09.2023). – Режим доступа: по подписке.</p> <p>4. Исаченко, О. В. Программное обеспечение компьютерных сетей : учебное пособие / О.В. Исаченко. — 2-е изд., испр. и доп. — Москва : ИНФРА-М, 2021. — 158 с. — (Среднее профессиональное образование). - ISBN 978-5-16-015447-3. - Текст: электронный. - URL: https://znanium.com/catalog/product/1189344 (дата обращения: 09.09.2023). – Режим доступа: по подписке.</p>		
Тема 3.2 Создание и тестирование модулей для мобильных приложений	Содержание учебного материала	82 (26/52/4)	
	1 Инструментарий среды разработки мобильных приложений	26	1
	2 Структура типичного мобильного приложения		
	3 Элементы управления и контейнеры		
	4 Работа со списками		
	5 Способы хранения данных		
	Лабораторные работы	52	2
	1. Создание эмуляторов и подключение устройств» 2. Настройка режима терминала» 3. Создание нового проекта» 4. Изучение и комментирование кода» 5. Лабораторная работа «Изменение элементов дизайна» 6. Обработка событий: подсказки» 7. Обработка событий: цветовая индикация» 8. Подготовка стандартных модулей» 9. Обработка событий: переключение между экранами» 10. Передача данных между модулями» 11. Тестирование и оптимизация мобильного приложения»	52	2
Самостоятельная работа обучающихся	4	3	

	<p>Отчет по лабораторным работам</p> <p>1. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва: ФО-РУМ: ИНФРА-М, 2022. — 400 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0812-9. - Текст: электронный. - URL: https://znanium.com/catalog/product/1794453 (дата обращения: 09.09.2023). – Режим доступа: по подписке.</p> <p>3. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отрасле-вой направленности: учебное пособие / Г. Н. Федорова. — Москва: КУРС: ИНФРА-М, 2021. — 336 с. — (Среднее профессиональное образование). - ISBN 978-5-906818-41-6. - Текст: электронный. - URL: https://znanium.com/catalog/product/1138896 (дата обращения: 09.09.2023). – Режим доступа: по подписке.</p> <p>4. Исаченко, О. В. Программное обеспечение компьютерных сетей : учебное пособие / О.В. Исаченко. — 2-е изд., испр. и доп. — Москва : ИНФРА-М, 2021. — 158 с. — (Среднее профессиональное образование). - ISBN 978-5-16-015447-3. - Текст: электронный. - URL: https://znanium.com/catalog/product/1189344 (дата обращения: 09.09.2023). – Режим доступа: по подписке.</p>	4		
Итого		169		
Консультация		3		
Дифференцированный зачет в 7 семестре		0		
Всего по модулю		172		
МДК.01.04 Системное программирование		140 (51/85/4)		
Тема 4.1 Программирование на языке низкого уровня	Содержание учебного материала	51	<i>1</i>	
	1	Подсистемы управления ресурсами.		
	2	Управление процессами.		
	3	Управление потоками.		
	4	Параллельная обработка потоков.	51	<i>1</i>
	5	Создание процессов и потоков.		
	6	Обмен данными между процессами. Передача сообщений.		

	7	Анонимные и именованные каналы.		
	8	Сетевое программирование сокетов.		
	9	Динамически подключаемые библиотеки DLL		
	10	Сервисы.		
	11	Виртуальная память. Выделение памяти процессам.		
	12	Работа с буфером экрана.		
	Лабораторные работы		85	
	1.	Использование потоков.	85	2
	2.	Обмен данными.		
	3.	Сетевое программирование сокетов.		
	4.	Работы с буфером экрана.		
	Самостоятельная работа обучающихся		4	
	Отчет по лабораторным работам		4	3
	1. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва: ФО-РУМ: ИНФРА-М, 2022. — 400 с. — (Среднее профессиональное образование). — ISBN 978-5-8199-0812-9. — Текст: электронный. — URL: https://znanium.com/catalog/product/1794453 (дата обращения: 09.09.2023). — Режим доступа: по подписке.			
	3. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отрасле-вой направленности: учебное пособие / Г. Н. Федорова. — Москва: КУРС: ИНФРА-М, 2021. — 336 с. — (Среднее профессиональное образование). — ISBN 978-5-906818-41-6. — Текст: электронный. — URL: https://znanium.com/catalog/product/1138896 (дата обращения: 09.09.2023). — Режим доступа: по подписке.			
	4. Исаченко, О. В. Программное обеспечение компьютерных сетей: учебное пособие / О.В. Исаченко. — 2-е изд., испр. И доп. — Москва : ИНФРА-М, 2021. — 158 с. — (Среднее профессиональное образование). — ISBN 978-5-16-015447-3. — Текст: электронный. — URL: https://znanium.com/catalog/product/1189344 (дата обращения: 09.09.2023). — Режим доступа: по подписке.			
Итого			140	
Консультация			1	

Дифференцированный зачет в 6 семестре	0	
Всего по модулю	141	
Учебная практика Дифференцированный зачет в 5 семестре	72	
Вводный инструктаж, инструктаж по технике безопасности. Ознакомление со структурой и с трудовым распорядком организации. Разрабатывать алгоритм решения поставленной задачи и реализовывать его средствами автоматизированного проектирования Разрабатывать код программного продукта на основе готовой спецификации на уровне модуля Использовать инструментальные средства на этапе отладки программного продукта Проводить тестирование программного модуля по определенному сценарию Использовать инструментальные средства на этапе тестирования программного продукта Анализировать алгоритмы, в том числе с применением инструментальных средств Осуществлять рефакторинг и оптимизацию программного кода Разрабатывать мобильные приложения Оформление документов по практике: путевки студента-практиканта, дневника, характеристики, аттестационного листа, отчета. Дифференцированный зачет.	72	
Производственная практика Дифференцированный зачет в 7 семестре	108	
Вводный инструктаж, инструктаж по технике безопасности. Ознакомление со структурой и с трудовым распорядком организации. Изучение предметной области работы предприятия Анализ и проектирование программного обеспечения с использованием объектно-ориентированного подхода Диаграммы вариантов использования Диаграммы деятельности Диаграммы последовательности Проектирование базы данных приложения (метод нормальных форм или метод ER-диаграмм) Обзор и выбор инструментальных средств создания приложений Обзор и выбор языков программирования и СУБД для создания приложения Создание объектов базы данных в конкретной СУБД Создание макета приложения Подключение базы данных к макету и реализация функций приложения Разграничение прав доступа для пользователей приложения Тестирование и отладка приложения Создание справочной системы приложения Обзор и выбор языков программирования и инструментальных средств создания мобильных приложений Создание макета мобильного приложения Реализация функционала мобильного приложения Отладка и тестирование мобильного приложения на эмуляторе Тестирование мобильного приложения на реальном устройстве.	108	

Оформление документов по практике: путевки студента-практиканта, дневника, характеристики, аттестационного листа, отчета. Дифференцированный зачет		
Всего:	925	
Экзамен по модулю в 7 семестре	8	
Всего по профессиональному модулю.	933	

*Для характеристики уровня освоения учебного материала используются следующие обозначения:

1. – ознакомительный (узнавание ранее изученных объектов, свойств);
2. – репродуктивный (выполнение деятельности по образцу, инструкции или под руководством)
3. – продуктивный (планирование и самостоятельное выполнение деятельности, решение проблемных задач)

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

МДК. 01.01 Разработка программных модулей	<p>Лаборатория «Программного обеспечения и сопровождения компьютерных систем». Автоматизированные рабочие места обучающихся: системный блок (процессор Intel Core i7, оперативная память 16 Гб), автоматизированное рабочее место преподавателя: системный блок (процессор Intel Core i7, оперативная память 16 Гб), комплект мебели (посадочных мест), комплект мебели (посадочных мест) для преподавателя, проектор и экран (интерактивная доска), трибуна, кондиционер, настенные полки, шкаф двухстворчатый с полками, веб-камера, маркерная доска, выход в Интернет, внутривузовская компьютерная сеть, доступ в электронную информационно-образовательную среду. Программное обеспечение общего и профессионального назначения: Office Professional Plus 2010, GIMP, Inkscape, Notepad ++, Python, Lazarus</p>
	<p>Студия «Разработки дизайна веб-приложений». Автоматизированные рабочие места обучающихся с конфигурацией: процессор Intel Core i7 10700K, дискретная видеокарта 6 Гб, оперативная память ОЗУ - 16 Гб, 2 монитора 23.8, 1 мышь, 1 клавиатура, автоматизированное рабочее место преподавателя с конфигурацией (процессор Intel Core i7 10700K, дискретная видеокарта 6 Гб, оперативная память ОЗУ - 16 Гб, монитор 23.8, мышь, клавиатура), специализированная эргономичная мебель для работы за компьютером: компьютерный стол, комплект мебели (посадочных мест), комплект мебели (посадочных мест) для преподавателя проектор и экран (Интерактивная доска), трибуна, кондиционер, настенные полки, шкаф двухстворчатый с полками, веб-камера, маркерная доска, принтер А3, цветной, Многофункциональное устройство (МФУ) формата А4, выход в Интернет, внутривузовская компьютерная сеть, доступ в электронную информационно-образовательную среду. Программное обеспечение общего и профессионального назначения: Office Professional Plus 2010, GIMP, Inkscape, Notepad ++, Python, Lazarus, EclipseIDEforJavaEEDevelopers, NETFrameworkJDK 8, MicrosoftSQLServerExpressEdition, MySQLInstallerforWindows, NetBeans, SQLServerManagementStudio, MicrosoftSQLServerJavaConnector, AndroidStudio</p>
	<p>Лаборатория «Разработка веб-приложений». Автоматизированные рабочие места обучающихся с конфигурацией: (процессор Intel Core i7, дискретная видеокарта 6 Гб, оперативная память ОЗУ - 16 Гб, монитор 23.8, мышь, клавиатура), автоматизированное рабочее место преподавателя с конфигурацией: (процессор Intel Core i7, дискретная видеокарта 6 Гб, оперативная память ОЗУ - 16 Гб, монитор 23.8, мышь, клавиатура) , специализированная эргономичная мебель для работы за компьютером: компьютерный стол, комплект мебели (посадочных мест), комплект мебели (посадочных мест) для преподавателя, проектор и экран (Интерактивная доска), трибуна, кондиционер, настенные полки, шкаф двухстворчатый с полками, веб-камера, маркерная доска, принтер</p>

	<p>А4, черно-белый, лазерный, выход в Интернет, внутривузовская компьютерная сеть, доступ в электронную информационно-образовательную среду. Программное обеспечение общего и профессионального назначения: Office Professional Plus 2010, GIMP, Inkscape, Notepad ++, Python, Lazarus. Виртуальный сервер на базе преподавательского рабочего места: (EclipseIDEforJavaEEDevelopers, NETFrameworkJDK 8, MicrosoftSQLServerExpressEdition, MySQLInstallerforWindows, NetBeans, SQLServerManagementStudio, MicrosoftSQLServerJavaConnector, AndroidStudio)</p>
<p>МДК.01.02 Поддержка и тестирование программных модулей</p>	<p>Лаборатория «Программного обеспечения и сопровождения компьютерных систем». Автоматизированные рабочие места обучающихся: системный блок (процессор Intel Core i7, оперативная память 16 ГБ), автоматизированное рабочее место преподавателя: системный блок (процессор Intel Core i7, оперативная память 16 ГБ), комплект мебели (посадочных мест), комплект мебели (посадочных мест) для преподавателя, проектор и экран (интерактивная доска), трибуна, кондиционер, настенные полки, шкаф двухстворчатый с полками, веб-камера, маркерная доска, выход в Интернет, внутривузовская компьютерная сеть, доступ в электронную информационно-образовательную среду. Программное обеспечение общего и профессионального назначения: Office Professional Plus 2010, GIMP, Inkscape, Notepad ++, Python, Lazarus</p> <p>Студия «Разработки дизайна веб-приложений». Автоматизированные рабочие места обучающихся с конфигурацией: процессор Intel Core i7 10700К, дискретная видеокарта 6 ГБ, оперативная память ОЗУ - 16 ГБ, 2 монитора 23.8, 1 мышшь, 1 клавиатура, автоматизированное рабочее место преподавателя с конфигурацией (процессор Intel Core i7 10700К, дискретная видеокарта 6 ГБ, оперативная память ОЗУ - 16 ГБ, монитор 23.8, мышшь, клавиатура), специализированная эргономичная мебель для работы за компьютером: компьютерный стол, комплект мебели (посадочных мест), комплект мебели (посадочных мест) для преподавателя проектор и экран (Интерактивная доска), трибуна, кондиционер, настенные полки, шкаф двухстворчатый с полками, веб-камера, маркерная доска, принтер А3, цветной, Многофункциональное устройство (МФУ) формата А4, выход в Интернет, внутривузовская компьютерная сеть, доступ в электронную информационно-образовательную среду. Программное обеспечение общего и профессионального назначения: Office Professional Plus 2010, GIMP, Inkscape, Notepad ++, Python, Lazarus, EclipseIDEforJavaEEDevelopers, NETFrameworkJDK 8, MicrosoftSQLServerExpressEdition, MySQLInstallerforWindows, NetBeans, SQLServerManagementStudio, MicrosoftSQLServerJavaConnector, AndroidStudio</p> <p>Лаборатория «Разработка веб-приложений». Автоматизированные рабочие места обучающихся с конфигурацией: (процессор Intel Core i7, дискретная видеокарта 6 ГБ, оператив-</p>

	<p>ная память ОЗУ - 16 ГБ, монитор 23,8, мышь, клавиатура), автоматизированное рабочее место преподавателя с конфигурацией: (процессор Intel Core i7, дискретная видеокарта 6 ГБ, оперативная память ОЗУ - 16 ГБ, монитор 23.8, мышь, клавиатура) , специализированная эргономичная мебель для работы за компьютером: компьютерный стол, комплект мебели (посадочных мест), комплект мебели (посадочных мест) для преподавателя, проектор и экран (Интерактивная доска), трибуна, кондиционер, настенные полки, шкаф двухстворчатый с полками, веб-камера, маркерная доска, принтер А4, черно-белый, лазерный, выход в Интернет, внутривузовская компьютерная сеть, доступ в электронную информационно-образовательную среду. Программное обеспечение общего и профессионального назначения: Office Professional Plus 2010, GIMP, Inkscape, Notepad ++, Python, Lazarus. Виртуальный сервер на базе преподавательского рабочего места: (EclipseIDEforJavaEEDevelopers, NETFrameworkJDK 8, MicrosoftSQLServerExpressEdition, MySQLInstallerforWindows, NetBeans, SQLServerManagementStudio, MicrosoftSQLServerJavaConnector, AndroidStudio)</p>
<p>МДК.01.03 Разработка мобильных приложений</p>	<p>Лаборатория «Программного обеспечения и сопровождения компьютерных систем». Автоматизированные рабочие места обучающихся: системный блок (процессор Intel Core i7, оперативная память 16 ГБ), автоматизированное рабочее место преподавателя: системный блок (процессор Intel Core i7, оперативная память 16 ГБ), комплект мебели (посадочных мест), комплект мебели (посадочных мест) для преподавателя, проектор и экран (интерактивная доска), трибуна, кондиционер, настенные полки, шкаф двухстворчатый с полками, веб-камера, маркерная доска, выход в Интернет, внутривузовская компьютерная сеть, доступ в электронную информационно-образовательную среду. Программное обеспечение общего и профессионального назначения: Office Professional Plus 2010, GIMP, Inkscape, Notepad ++, Python, Lazarus</p> <p>Студия «Разработки дизайна веб-приложений». Автоматизированные рабочие места обучающихся с конфигурацией: процессор Intel Core i7 10700К, дискретная видеокарта 6 ГБ, оперативная память ОЗУ - 16 ГБ, 2 монитора 23.8, 1 мышь, 1 клавиатура, автоматизированное рабочее место преподавателя с конфигурацией (процессор Intel Core i7 10700К, дискретная видеокарта 6 ГБ, оперативная память ОЗУ - 16 ГБ, монитор 23.8, мышь, клавиатура), специализированная эргономичная мебель для работы за компьютером: компьютерный стол, комплект мебели (посадочных мест), комплект мебели (посадочных мест) для преподавателя проектор и экран (Интерактивная доска), трибуна, кондиционер, настенные полки, шкаф двухстворчатый с полками, веб-камера, маркерная доска, принтер А3, цветной, Многофункциональное устройство (МФУ) формата А4, выход в Интернет, внутривузовская компьютерная сеть, доступ в электронную информационно-образовательную среду. Программное обеспече-</p>

	<p>ние общего и профессионального назначения: Office Professional Plus 2010, GIMP, Inkscape, Notepad ++, Python, Lazarus, EclipseIDEforJavaEEDevelopers, NETFrameworkJDK 8, MicrosoftSQLServerExpressEdition, MySQLInstallerforWindows, NetBeans, SQLServerManagementStudio, MicrosoftSQLServerJavaConnector, AndroidStudio</p> <p>Лаборатория «Разработка веб-приложений». Автоматизированные рабочие места обучающихся с конфигурацией: (процессор Intel Core i7, дискретная видеокарта 6 ГБ, оперативная память ОЗУ - 16 ГБ, монитор 23,8, мышь, клавиатура), автоматизированное рабочее место преподавателя с конфигурацией: (процессор Intel Core i7, дискретная видеокарта 6 ГБ, оперативная память ОЗУ - 16 ГБ, монитор 23.8, мышь, клавиатура) , специализированная эргономичная мебель для работы за компьютером: компьютерный стол, комплект мебели (посадочных мест), комплект мебели (посадочных мест) для преподавателя, проектор и экран (Интерактивная доска), трибуна, кондиционер, настенные полки, шкаф двухстворчатый с полками, веб-камера, маркерная доска, принтер А4, черно-белый, лазерный, выход в Интернет, внутривузовская компьютерная сеть, доступ в электронную информационно-образовательную среду. Программное обеспечение общего и профессионального назначения: Office Professional Plus 2010, GIMP, Inkscape, Notepad ++, Python, Lazarus. Виртуальный сервер на базе преподавательского рабочего места: (EclipseIDEforJavaEEDevelopers, NETFrameworkJDK 8, MicrosoftSQLServerExpressEdition, MySQLInstallerforWindows, NetBeans, SQLServerManagementStudio, MicrosoftSQLServerJavaConnector, AndroidStudio)</p>
МДК.01.04 Системное программирование	<p>Лаборатория «Программного обеспечения и сопровождения компьютерных систем». Автоматизированные рабочие места обучающихся: системный блок (процессор Intel Core i7, оперативная память 16 ГБ), автоматизированное рабочее место преподавателя: системный блок (процессор Intel Core i7, оперативная память 16 ГБ), комплект мебели (посадочных мест), комплект мебели (посадочных мест) для преподавателя, проектор и экран (интерактивная доска), трибуна, кондиционер, настенные полки, шкаф двухстворчатый с полками, веб-камера, маркерная доска, выход в Интернет, внутривузовская компьютерная сеть, доступ в электронную информационно-образовательную среду. Программное обеспечение общего и профессионального назначения: Office Professional Plus 2010, GIMP, Inkscape, Notepad ++, Python, Lazarus</p> <p>Помещение для практической подготовки. Комплект мебели для преподавателя, посадочные места для учащихся, компьютеры, интерактивная панель, меловая доска настенная, выход в интернет, внутривузовская компьютерная сеть, доступ в электронную информационно-образовательную среду. ПО: Microsoft office 365</p>

3.2. Информационное обеспечение обучения (основная и дополнительная литература, интернет-ресурсы)

Учебно-методическая литература для данного профессионального модуля имеется в наличии в электронно-библиотечной системе "ZNANIUM.COM" и электронно-библиотечной системе «Консультант студента», доступ к которым предоставлен обучающимся. ЭБС "ZNANIUM.COM" «Консультант студента» содержат произведения крупнейших российских учёных, руководителей государственных органов, преподавателей ведущих вузов страны, высококвалифицированных специалистов в различных сферах бизнеса. Фонд библиотеки сформирован с учетом всех изменений образовательных стандартов и включает учебники, учебные пособия, монографии, авторефераты, диссертации, энциклопедии, словари и справочники, законодательно-нормативные документы, специальные периодические издания и издания, выпускаемые издательствами вузов. В настоящее время ЭБС ZNANIUM.COM и «Консультант студента» соответствует всем требованиям федеральных государственных образовательных стандартов среднего профессионального образования нового поколения.

Учебно-методическая литература для данного профессионального модуля имеется в наличии в электронно-библиотечной системе Издательства "Лань", доступ к которой предоставлен обучающимся. ЭБС Издательства "Лань" включает в себя электронные версии книг издательства "Лань" и других ведущих издательств учебной литературы, а также электронные версии периодических изданий.

Учебно-методическое и информационное обеспечение модуля ПМ.01

Основная литература:
Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2023. — 400 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0812-9. - Текст : электронный. - URL: https://znanium.com/catalog/product/1895679 (дата обращения: 02.11.2023). – Режим доступа: по подписке.
Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности : учебное пособие / Г.Н. Федорова. — Москва : КУРС : ИНФРА-М, 2023. — 336 с. — (Среднее профессиональное образование). - ISBN 978-5-906818-41-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/1896457 (дата обращения: 02.11.2023). – Режим доступа: по подписке.
Исаченко, О. В. Программное обеспечение компьютерных сетей : учебное пособие / О.В. Исаченко. — 2-е изд., испр. и доп. — Москва : ИНФРА-М, 2024. — 158 с. — (Среднее профессиональное образование). - ISBN 978-5-16-015447-3. - Текст : электронный. - URL: https://znanium.com/catalog/product/2111926 (дата обращения: 02.11.2023). – Режим доступа: по подписке.
Дополнительная литература:
Голицына, О. Л. Программное обеспечение : учебное пособие / О. Л. Голицына, Т. Л. Партыка, И. И. Попов. - 4-е изд., перераб. и доп. - Москва : ФОРУМ : ИНФРА-М, 2021. - 448 с. : ил. - (Профессиональное образование). - ISBN 978-5-91134-711-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/1189345 (дата обращения: 02.11.2023). – Режим доступа: по подписке.
Лисьев, Г. А. Программное обеспечение компьютерных сетей и web-серверов : учебное пособие / Г.А. Лисьев, П.Ю. Романов, Ю.И. Аскерко. — Москва : ИНФРА-М, 2023. — 145 с. — (Среднее профессиональное образование). - ISBN 978-5-16-014514-3. - Текст : электронный. - URL: https://znanium.com/catalog/product/1878635 (дата обращения: 02.11.2023). – Режим доступа: по подписке.
Черников, Б. В. Управление качеством программного обеспечения : учебник / Б.В. Черников.

— Москва : ФОРУМ : ИНФРА-М, 2022. — 240 с. — (Среднее профессиональное образование).
- ISBN 978-5-8199-0902-7. - Текст : электронный. -
URL: <https://znanium.com/catalog/product/1850732> (дата обращения: 02.11.2023). – Режим
доступа: по подписке.

Колдаев, В. Д. Архитектура ЭВМ : учебное пособие / В.Д. Колдаев, С.А. Лупин. — Москва :
ФОРУМ : ИНФРА-М, 2023. — 383 с. — (Среднее профессиональное образование). - ISBN 978-
5-8199-0868-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1896460> (дата
обращения: 02.11.2023). – Режим доступа: по подписке.

Белугина, С. В. Разработка программных модулей программного обеспечения для
компьютерных систем. Прикладное программирование / С. В. Белугина. — 4-е изд., стер. —
Санкт-Петербург : Лань, 2023. — 312 с. — ISBN 978-5-507-46061-8. — Текст : электронный //
Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/296975> (дата
обращения: 02.11.2023). — Режим доступа: для авториз. пользователей.

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

3.3. Общие требования к организации образовательного процесса

Организация образовательного процесса должна способствовать формированию об-
щих и профессиональных компетенций, освоение которых является результатом обучения по
профессиональному модулю.

В образовательном процессе используются активные и интерактивные формы прове-
дения занятий (компьютерные симуляции, деловые и ролевые игры, разбор конкретных си-
туаций, групповые дискуссии и т.д.) в сочетании с внеаудиторной работой с целью формиро-
вания и развития общих и профессиональных компетенций обучающихся.

Теоретические и практические занятия проводятся в лаборатории образовательного
учреждения преподавателями данных междисциплинарных курсов.

Организация учебной практики и производственной практики (по профилю специаль-
ности) по профессиональному модулю является неотъемлемой составляющей при реализа-
ции ППССЗ.

Учебная практика и производственная практика направлены на формирование у
обучающихся общих компетенций и профессиональных компетенций, приобретение практи-
ческого опыта.

Учебная и производственная практика проводятся на профильных предприятиях (ор-
ганизациях), оснащённых современной техникой, применяющих новейшие технологии, со-
временные программные продукты и современную организацию труда, располагающих вы-
сококвалифицированным персоналом и реальными возможностями предприятия (организа-
ции) производственного обучения студентов. Учебная практика может проводиться в лабо-
раториях и мастерских Елабужского института (филиала) ФГАОУ ВО «Казанский (При-
волжский) федеральный университет» или профильных предприятиях (организациях).

3.4. Кадровое обеспечение образовательного процесса:

Требования к квалификации педагогических кадров, обеспечивающих обучение по
междисциплинарным курсам: высшее образование, соответствующее профилю преподавае-
мой дисциплины (модуля). Опыт деятельности в организациях соответствующей профессио-
нальной сферы является обязательным для преподавателей, отвечающих за освоение обуча-
ющимися профессионального цикла. Преподаватели получают дополнительное профессио-
нальное образование по программам повышения квалификации, в том числе в форме стажир-
овки в профильных организациях не реже одного раза в три года.

ОК 01. Выбирать способы решения	Уметь: осуществлять разработку кода программного модуля на языках низ-	МДК 01.01 – дифференци- рованный зачет (задания 1-
------------------------------------	---	---

задач профессиональной деятельности, применительно к различным контекстам.	кого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	30), экзамен (вопросы 1-35) тестовые задания МДК 01.02 – экзамен (вопросы 1-20), тестовые задания МДК 01.03 – дифференцированный зачет (вопросы 1-20), практические задания (1-4) МДК 01.04 – дифференцированный зачет (вопросы 1-25), практические задания (1-12) Учебная практика- дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике) Производственная практика -
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике)
ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	МДК 01.01 – дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35), тестовые задания МДК 01.02 – экзамен (вопросы 1-20), тестовые задания МДК 01.03 – дифференцированный зачет (вопросы 1-20), практические задания (1-4) МДК 01.04 – дифференцированный зачет (вопросы 1-25), практические задания (1-12) Учебная практика- дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике) Производственная практика -
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике)

		рактеристика, дневник по практике)
ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	МДК 01.01 – дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35), тестовые задания МДК 01.02 – экзамен (вопросы 1-20), тестовые задания МДК 01.03 – дифференцированный зачет (вопросы 1-20), практические задания (1-4) МДК 01.04 – дифференцированный зачет (вопросы 1-25), практические задания (1-12)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Учебная практика- дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике) Производственная практика - дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике)
ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	МДК 01.01 – дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35), тестовые задания МДК 01.02 – экзамен (вопросы 1-20), тестовые задания МДК 01.03 – дифференцированный зачет (вопросы 1-20), практические задания (1-4) МДК 01.04 – дифференцированный зачет (вопросы 1-25), практические задания (1-12)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Учебная практика- дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике) Производственная практика

		- дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике)
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	МДК 01.01 – дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35), тестовые задания МДК 01.02 – экзамен (вопросы 1-20), тестовые задания МДК 01.03 – дифференцированный зачет (вопросы 1-20), практические задания (1-4) МДК 01.04 – дифференцированный зачет (вопросы 1-25), практические задания (1-12)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Учебная практика- дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике) Производственная практика - дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике)
ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	МДК 01.01 – дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35), тестовые задания МДК 01.02 – экзамен (вопросы 1-20), тестовые задания МДК 01.03 – дифференцированный зачет (вопросы 1-20), практические задания (1-4) МДК 01.04 – дифференцированный зачет (вопросы 1-25), практические задания (1-12)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптими-	Учебная практика- дифференцированный зачет (индивидуальное задание, аттестационный лист, харак-

	зации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	теристика, дневник по практике) Производственная практика - дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике)
ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	МДК 01.01 – дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35), тестовые задания МДК 01.02 – экзамен (вопросы 1-20), тестовые задания МДК 01.03 – дифференцированный зачет (вопросы 1-20), практические задания (1-4) МДК 01.04 – дифференцированный зачет (вопросы 1-25), практические задания (1-12)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Учебная практика- дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике) Производственная практика - дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике)
ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	МДК 01.01 – дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35), тестовые задания МДК 01.02 – экзамен (вопросы 1-20), тестовые задания МДК 01.03 – дифференцированный зачет (вопросы 1-20), практические задания (1-4) МДК 01.04 – дифференцированный зачет (вопросы 1-25), практические задания (1-12)
	Знать: основные этапы разработки программного обеспечения; основ-	

	ные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Учебная практика- дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике) Производственная практика - дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике)
ОК. 09. Использовать информационные технологии в профессиональной деятельности	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	МДК 01.01 – дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35), тестовые задания МДК 01.02 – экзамен (вопросы 1-20), тестовые задания МДК 01.03 – дифференцированный зачет (вопросы 1-20), практические задания (1-4) МДК 01.04 – дифференцированный зачет (вопросы 1-25), практические задания (1-12)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Учебная практика- дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике) Производственная практика - дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике)
ОК.10. Пользоваться профессиональной документацией на государственном и иностранном языках	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию	МДК 01.01 – дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35), тестовые задания МДК 01.02 – экзамен (вопросы 1-20), тестовые задания МДК 01.03 – дифференцированный зачет (вопросы 1-20), практические задания (1-4) МДК 01.04 – дифференци-

	<p>на программные средства</p> <p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	<p>рованный зачет (вопросы 1-25), практические задания (1-12)</p> <p>Учебная практика- дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике)</p> <p>Производственная практика -</p> <p>дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике)</p>
<p>ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием</p>	<p>Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений</p> <p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p> <p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	<p>МДК 01.01 – дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35), тестовые задания</p> <p>МДК 01.02 – экзамен (вопросы 1-20), тестовые задания</p> <p>МДК 01.03 – дифференцированный зачет (вопросы 1-20), практические задания (1-4)</p> <p>МДК 01.04 – дифференцированный зачет (вопросы 1-25), практические задания (1-12)</p> <p>Учебная практика- дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике)</p> <p>Производственная практика -</p> <p>дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике)</p>
<p>ПК 1.2. Разрабатывать программные</p>	<p>Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на</p>	<p>МДК 01.01 – дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35),</p>

<p>модули в соответствии с техническим заданием</p>	<p>уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений</p> <p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p> <p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	<p>тестовые задания МДК 01.02 – экзамен (вопросы 1-20), тестовые задания МДК 01.03 – дифференцированный зачет (вопросы 1-20), практические задания (1-4) МДК 01.04 – дифференцированный зачет (вопросы 1-25), практические задания (1-12) Учебная практика- дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике) Производственная практика - дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике)</p>
<p>ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств</p>	<p>Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений</p> <p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оп-</p>	<p>МДК 01.01 – дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35), тестовые задания МДК 01.02 – экзамен (вопросы 1-20), тестовые задания МДК 01.03 – дифференцированный зачет (вопросы 1-20), практические задания (1-4) МДК 01.04 – дифференцированный зачет (вопросы 1-25), практические задания (1-12) Учебная практика- дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике) Производственная практика</p>

	<p>тимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p> <p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	<p>- дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике)</p>
<p>ПК 1.4. Выполнять тестирование программных модулей</p>	<p>Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений</p> <p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p> <p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	<p>МДК 01.01 – дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35), тестовые задания МДК 01.02 – экзамен (вопросы 1-20), тестовые задания МДК 01.03 – дифференцированный зачет (вопросы 1-20), практические задания (1-4) МДК 01.04 – дифференцированный зачет (вопросы 1-25), практические задания (1-12) Учебная практика- дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике) Производственная практика - дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике)</p>
<p>ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода</p>	<p>Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию;</p>	<p>МДК 01.01 – дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35), тестовые задания МДК 01.02 – экзамен (вопросы 1-20), тестовые задания МДК 01.03 – дифференци-</p>

	использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений	рованный зачет (вопросы 1-20), практические задания (1-4) МДК 01.04 – дифференцированный зачет (вопросы 1-25), практические задания (1-12)
	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	Учебная практика- дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике) Производственная практика - дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	
ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ	Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений	МДК 01.01 – дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35), тестовые задания МДК 01.02 – экзамен (вопросы 1-20), тестовые задания МДК 01.03 – дифференцированный зачет (вопросы 1-20), практические задания (1-4) МДК 01.04 – дифференцированный зачет (вопросы 1-25), практические задания (1-12)
	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	Учебная практика- дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по практике) Производственная практика - дифференцированный зачет (индивидуальное задание, аттестационный лист, характеристика, дневник по
	Знать: основные этапы разработки программного обеспечения; основ-	

	ные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	практике)
--	--	-----------

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Елабужский институт (филиал) федерального государственного автономного
образовательного учреждения высшего образования
«Казанский (Приволжский) федеральный университет»

УТВЕРЖДАЮ

Заместитель директора по
образовательной деятельности
Елабужского института ЕИ КФУ
И.П. Михайлова
«01» марта 2024 г.



ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

МДК.01.01 Разработка программных модулей

(наименование дисциплины)

09.02.07 Информационные системы и программирование

(код и наименование специальности)

Программист

(квалификация выпускника)

г.Елабуга, 2024

**Паспорт
фонда оценочных средств по
МДК. 01.01 Разработка программных модулей**
(наименование междисциплинарного курса)

Результаты (освоенные профессиональ- ные компетенции)	Основные показатели оценки результата	Формы и методы кон- троля и оценки
ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;</p> <p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;</p>	Отчет по лабораторным работам. Тестовые задания Дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35)
ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;</p> <p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;</p>	Отчет по лабораторным работам. Тестовые задания Дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35)
ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;</p> <p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структур-</p>	Отчет по лабораторным работам. Тестовые задания Дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35)

	ного и объектно-ориентированного программирования;	
ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	Отчет по лабораторным работам. Тестовые задания Дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	Отчет по лабораторным работам. Тестовые задания Дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	Отчет по лабораторным работам. Тестовые задания Дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффектив-	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выпол-	Отчет по лабораторным работам. Тестовые задания Дифференцированный зачет (задания 1-30), экзамен

но действовать в чрезвычайных ситуациях	нять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	(вопросы 1-35)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	Отчет по лабораторным работам. Тестовые задания Дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ОК. 09. Использовать информационные технологии в профессиональной деятельности	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	Отчет по лабораторным работам. Тестовые задания Дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ОК.10. Пользоваться профессиональной документацией на государственном и иностранном языках	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	Отчет по лабораторным работам. Тестовые задания Дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35)

	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на уровне модуля;	Отчет по лабораторным работам. Тестовые задания Дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35)
	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием	Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на уровне модуля;	Отчет по лабораторным работам. Тестовые задания Дифференцированный зачет (задания 1-30), экзамен (вопросы 1-35)
	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Елабужский институт (филиал) федерального государственного автономного образовательного учреждения высшего образования
«Казанский (Приволжский) федеральный университет»

**Отчет по лабораторным работам
по МДК.01.01
(ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2)**

Тема 1.2.

Структурное программирование (ОК 01-10, ПК 1.1-1.2)

Оценка сложности алгоритмов сортировки.
Оценка сложности алгоритмов поиска.
Оценка сложности рекурсивных алгоритмов.
Оценка сложности эвристических алгоритмов.

Тема 1.3.

Объектно-ориентированное программирование (ОК 01-10, ПК 1.1-1.2)

Работа с классами.
Перегрузка методов.
Определение операций в классе.
Создание наследованных классов
Работа с объектами через интерфейсы.
Использование стандартных интерфейсов.
Работа с типом данных структура.
Коллекции. Параметризованные классы.
Использование регулярных выражений
Операции со списками.

Тема 1.4. Паттерны проектирования (ОК 01-10, ПК 1.1-1.2)

Использование основных шаблонов.
Использование порождающих шаблонов.
Использование структурных шаблонов.
Использование поведенческих шаблонов

Тема 1.5. Событийно-управляемое программирование (ОК 01-10, ПК 1.1-1.2)

Разработка приложения с использованием текстовых компонентов
Разработка приложения с несколькими формами.
Разработка приложения с не визуальными компонентами.
Разработка игрового приложения.
Разработка приложения с анимацией

Тема 1.6. Оптимизация и рефакторинг кода (ОК 01-10, ПК 1.1-1.2)

Оптимизация и рефакторинг кода.

Тема 1.7 Разработка пользовательского интерфейса (ОК 01-10, ПК 1.1-1.2)

Разработка интерфейса пользователя

Тема 1.8 Основы ADO.Net (ОК 01-10, ПК 1.1-1.2)

Создание приложения с БД

Создание запросов к БД
Создание хранимых процедур

Индекс и расшифровка компетенции	Результаты обучения	Критерии оценивания результатов обучения (баллы)			
		2	3	4	5
ОК-1	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p>	<p>Не умеет Демонстрирует частичные умения, допуская грубые ошибки</p>	<p>Демонстрирует частичные умения без грубых ошибок</p>	<p>Умеет применять знания на практике в базовом объеме</p>	<p>Демонстрирует высокий уровень умений</p>
	<p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отлад-</p>	<p>Не знает Допускает грубые ошибки</p>	<p>Демонстрирует частичные знания без грубых ошибок</p>	<p>Знает достаточно в базовом объеме</p>	<p>Демонстрирует высокий уровень знаний</p>

	ки и тестирования программных продуктов				
ОК-2	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
ОК-3	Уметь: осуществлять разработку	Не умеет Демонстри-	Демонстрирует частичные	Умеет применять зна-	Демонстрирует высо-

	<p>кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p>	<p>рует частичные умения, допуская грубые ошибки</p>	<p>умения без грубых ошибок</p>	<p>ния на практике в базовом объеме</p>	<p>кий уровень умений</p>
	<p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	<p>Не знает Допускает грубые ошибки</p>	<p>Демонстрирует частичные знания без грубых ошибок</p>	<p>Знает достаточно в базовом объеме</p>	<p>Демонстрирует высокий уровень знаний</p>
<p>ОК-4</p>	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать про-</p>	<p>Не умеет Демонстрирует частичные умения, допуская грубые ошибки</p>	<p>Демонстрирует частичные умения без грубых ошибок</p>	<p>Умеет применять знания на практике в базовом объеме</p>	<p>Демонстрирует высокий уровень умений</p>

	грамму по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства				
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
ОК-5	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений

	тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства				
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
ОК- 6	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода про-	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений

	граммного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства				
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
ОК- 7	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять опти-	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений

	<p>мизацию и рефакторинг программного кода; оформлять документацию на программные средства</p>				
	<p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	<p>Не знает Допускает грубые ошибки</p>	<p>Демонстрирует частичные знания без грубых ошибок</p>	<p>Знает достаточно в базовом объеме</p>	<p>Демонстрирует высокий уровень знаний</p>
ОК- 8	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на про-</p>	<p>Не умеет Демонстрирует частичные умения, допуская грубые ошибки</p>	<p>Демонстрирует частичные умения без грубых ошибок</p>	<p>Умеет применять знания на практике в базовом объеме</p>	<p>Демонстрирует высокий уровень умений</p>

	граммные средства				
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
ОК- 9	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений
	Знать: основные этапы разработки программного	Не знает Допускает грубые	Демонстрирует частичные знания без	Знает достаточно в базовом объё-	Демонстрирует высокий уровень

	обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	ошибки	грубых ошибок	ме	знаний
ОК-10	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний

	ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов				
ПК- 1.1	Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений	Не владеет Демонстрирует низкий уровень владения, допуская грубые ошибки	Демонстрирует частичные владения без грубых ошибок	Владеет базовыми приемами	Демонстрирует владения на высоком уровне
	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разра-	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений

	ботку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства				
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
ПК 1.2	Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе	Не владеет Демонстрирует низкий уровень владения, допуская грубые ошибки	Демонстрирует частичные владения без грубых ошибок	Владеет базовыми приёмами	Демонстрирует владения на высоком уровне

	отладки программного продукта; разработке мобильных приложений				
	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p>	<p>Не умеет Демонстрирует частичные умения, допуская грубые ошибки</p>	<p>Демонстрирует частичные умения без грубых ошибок</p>	<p>Умеет применять знания на практике в базовом объеме</p>	<p>Демонстрирует высокий уровень умений</p>
	<p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	<p>Не знает Допускает грубые ошибки</p>	<p>Демонстрирует частичные знания без грубых ошибок</p>	<p>Знает достаточно в базовом объеме</p>	<p>Демонстрирует высокий уровень знаний</p>

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Елабужский институт (филиал) федерального государственного автономного образовательного учреждения высшего образования
«Казанский (Приволжский) федеральный университет»

Тестовые задания
по МДК.01.01 Разработка программных модулей
(ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2)

Вариант 1

1. В ассемблере команда вычитания

- 1) add
- 2) sub
- 3) mul
- 4) div
- 5) xor

Ответ: 2 (ОК 01-10, ПК 1.1-1.2)

2. Разрядность eax

- 1) 4
- 2) 8
- 3) 16
- 4) 24
- 5) 32

Ответ: 5 (ОК 01-10, ПК 1.1-1.2)

3. Результат работы фрагмента программы:

```
mov dl,8  
xor dl,9
```

- 1) 1
- 2) 17
- 3) 8
- 4) 9
- 5) 254

Ответ: 1 (ОК 01-10, ПК 1.1-1.2)

4. В микропроцессоре 80386 указатель стека

- 1) ax
- 2) ip
- 3) bx
- 4) sp
- 5) ah

Ответ: 4 (ОК 01-10, ПК 1.1-1.2)

5. Комментарии в ассемблере отделяются символами:

- 1) ;
- 2) *)
- 3) /*
- 4) {

5) \

Ответ: 1 (ОК 01-10, ПК 1.1-1.2)

6. В ассемблере команда умножения

1) add

2) sub

3) mul

4) div

5) xor

Ответ: 3 (ОК 01-10, ПК 1.1-1.2)

7. Результат работы фрагмента программы:

```
mov dx,401
```

```
shr dx
```

1) 400

2) 402

3) 200

4) 802

5) 15983

Ответ: 1 (ОК 01-10, ПК 1.1-1.2)

8. jne в ассемблере

1) нет такой команды

2) переход, если не равно

3) переход к подпрограмме

4) конец программы

5) начало блока описания параметров подпрограммы

Ответ: 2 (ОК 01-10, ПК 1.1-1.2)

9. Системная программа:

1) Microsoft Word

2) Paint

3) Дефрагментация

4) Corel Draw

5) Пасьянс

Ответ: 3 (ОК 01-10, ПК 1.1-1.2)

10. Сколько байт определяет команда DQ:

1) 1

2) 2

3) 4

4) 8

5) 10.

Ответ: 4 (ОК 01-10, ПК 1.1-1.2)

11. МП 8086 является:

1) 8-разрядным;

2) 16-разрядным;

3) 24-разрядным;

4) 32-разрядным

Ответ: 2 (ОК 01-10, ПК 1.1-1.2)

12. Процессор Pentium имеет:

- 1) 16-разрядную магистраль;
- 2) 32-разрядную магистраль;
- 3) 64-разрядную магистраль;
- 4) 128-разрядную магистраль.

Ответ: 3 (ОК 01-10, ПК 1.1-1.2)

13. Для программиста доступна:

- 1) вся рабочая память процессора;
- 2) внутренняя память процессора недоступна;
- 3) внутренняя память доступна через регистры;
- 4) внутренняя память доступна через информационную магистраль.

Ответ: 3 (ОК 01-10, ПК 1.1-1.2)

14. Говоря о 16-разрядной ЭВМ, имеют в виду:

- 1) разрядность шины данных 16 бит;
- 2) разрядность шины адреса 16 бит;
- 3) размер слова 16 бит;
- 4) размер внутренних регистров памяти 16 бит.

Ответ: 4 (ОК 01-10, ПК 1.1-1.2)

15. Минимальной адресной единицей является:

- 1) бит;
- 2) байт;
- 3) слово;
- 4) двойное слово.

Ответ: 2 (ОК 01-10, ПК 1.1-1.2)

16. Если адрес находится в самой команде, то мы имеем дело:

- 1) с косвенной адресацией;
- 2) с основной адресацией;
- 3) с прямой адресацией;
- 4) с двойной косвенной адресацией.

Ответ: 3 (ОК 01-10, ПК 1.1-1.2)

17. Базовый адрес является:

- 1) начальной точкой массива данных;
- 2) конечной точкой массива данных;
- 3) промежуточной точкой массива данных;
- 4) массивом данных.

Ответ: 1 (ОК 01-10, ПК 1.1-1.2)

18. ROM-BIOS – часть какой памяти?

- 1) ПЗУ;
- 2) ОЗУ;
- 3) Дисковой.

Ответ: 1 (ОК 01-10, ПК 1.1-1.2)

19. Какие имена обозначают сегментные регистры ЦП?

- 1) CS;
- 2) IP;
- 3) SS;

- 4) AL;
- 5) DS;
- 6) ES

Ответ: 1,3,5,6 (ОК 01-10, ПК 1.1-1.2)

20. Получение EXE-файла из исходного модуля AA.asm:

- 1) tasm aa.asm
tlink aa.obj
- 2) tasm aa.asm
tlink aa.obj /t
- 3) tasm aa.asm /t
tlink aa.obj
- 4) tlink aa.obj
tasm aa.asm /t

Ответ: 1 (ОК 01-10, ПК 1.1-1.2)

21. Какой цвет определяет байт атрибут, имеющий значение 00011110b?

- 1) Чёрный фон, белый текст;
- 2) Белый фон, чёрный текст;
- 3) Тёмно синий фон, жёлтый текст;
- 4) Тёмно зелёный фон, белый текст.

Ответ: 3 (ОК 01-10, ПК 1.1-1.2)

22. Какой цвет определяет байт атрибут, имеющий значение 00101111b?

- 1) Чёрный фон, белый текст;
- 2) Белый фон, чёрный текст;
- 3) Тёмно синий фон, жёлтый текст;
- 4) Тёмно зелёный фон, белый текст.

Ответ: 4 (ОК 01-10, ПК 1.1-1.2)

23. Флаг (переполнение) регистра флагов, отвечающий за переполнение при выполнении арифметических операций:

- 1) O
- 2) D
- 3) I
- 4) T

Ответ: 1 (ОК 01-10, ПК 1.1-1.2)

24. Флаг (прерывание) регистра флагов, который разрешает / запрещает внешние прерывания

- 1) O
- 2) D
- 3) I
- 4) T

Ответ: 3 (ОК 01-10, ПК 1.1-1.2)

25. Флаг (знак) регистра флагов, отвечающий за знак результата выполненной команды (0 – плюс, 1 – минус)

- 1) S

- 2) Z
- 3) A
- 4) P

Ответ: 1 (ОК 01-10, ПК 1.1-1.2)

Вариант 2

1. В микропроцессоре 80386 нет регистра

- 1) bp
- 2) al
- 3) bl
- 4) el
- 5) sp

Ответ: 4 (ОК 01-10, ПК 1.1-1.2)

2. Ассемблер – язык

- 1) формальный
- 2) низкого уровня
- 3) высокого уровня
- 4) визуального программирования
- 5) автокодов

Ответ: 2 (ОК 01-10, ПК 1.1-1.2)

3. В ассемблере команда сложения

- 1) add
- 2) sub
- 3) mul
- 4) div
- 5) xor

Ответ: 1 (ОК 01-10, ПК 1.1-1.2)

4. Результат работы фрагмента программы:

```
mov dl,201
```

```
rol dl,1
```

- 1) 200
- 2) 202
- 3) 147
- 4) 228
- 5) 178

Ответ: 3 (ОК 01-10, ПК 1.1-1.2)

5. Что означает строка на ассемблере «sl db 5 dup(?)»:

- 1) Присвоение регистру sl содержимого ячейки памяти, адресуемой сегментом, определяемым es со смещением, равным сумме содержимого bx и числа 5.
- 2) sl – адрес переменной, хранящей 5 байт неопределенного значения.
- 3) Присвоение регистру sl содержимого ячейки памяти, адресуемой сегментом, определяемым es со смещением 5.
- 4) Арифметический сдвиг влево регистра sl на 5 разрядов, причем справа разряды заполняются произвольными значениями
- 5) sl – переменная, хранящая число 5

Ответ: 2 (ОК 01-10, ПК 1.1-1.2)

6. Нет флага:

- 1) переноса
- 2) нулевого результата
- 3) разрешения прерывания
- 4) регистра адреса
- 5) переполнения

Ответ: 4 (ОК 01-10, ПК 1.1-1.2)

7. Объявление сегмента кода на ассемблере

- 1) CODESG SEGMENT PARA 'Code'
- 2) CS SEGMENT
- 3) program SEGMENT
- 4) program code

Ответ: 1(ОК 01-10, ПК 1.1-1.2)

8. В ассемблере команда деления

- 1) add
- 2) sub
- 3) mul
- 4) div
- 5) imul

Ответ: 4(ОК 01-10, ПК 1.1-1.2)

9. В результате работы фрагмента программы:

```
mov ax,1001b
```

```
mov bx,1110b
```

```
xchg ax,bx
```

в регистрах ax:bx

- 1) 1110b:1001b
- 2) 10111b:1110b
- 3) 111b:1110b
- 4) 1000b:1110b
- 5) 1001b:1111b

Ответ: 1(ОК 01-10, ПК 1.1-1.2)

10. На языке ассемблера выполняет действия, противоположные push

- 1) and
- 2) xor
- 3) xlat
- 4) pop
- 5) mov

Ответ: 4 (ОК 01-10, ПК 1.1-1.2)

11. МП 80386 является:

- 1) 8-разрядным;
- 2) 16-разрядным;
- 3) 24-разрядным;
- 4) 32-разрядным

Ответ: 4 (ОК 01-10, ПК 1.1-1.2)

12. Внутренний КЭШ процессора Pentium разделен на:

- 1) КЭШ команд и КЭШ разрядов;

- 2) КЭШ разрядов и КЭШ данных
- 3) КЭШ команд и КЭШ адресов;
- 4) КЭШ команд и КЭШ данных.

Ответ: 4 (ОК 01-10, ПК 1.1-1.2)

13. Аккумулятор используется:

- 1) для указания на стек;
- 2) для битового сложения;
- 3) как место для проведения операций и сохранения их результатов;
- 4) как регистр приемника.

Ответ: 3(ОК 01-10, ПК 1.1-1.2)

14. В защищенном режиме работы процессора начальные адреса сегментов вычисляются:

- 1) умножением на 16 содержимого сегментных регистров;
- 2) умножением на 32 содержимого сегментных регистров;
- 3) извлекаются из таблиц сегментных дескрипторов;
- 4) извлекаются из таблиц сегментных дескрипторов, индексируемых теми же сегментными регистрами.

Ответ: 4 (ОК 01-10, ПК 1.1-1.2)

15. В методе косвенной адресации адрес памяти содержится:

- 1) в одном из регистров;
- 2) в команде;
- 3) в стеке;
- 4) в ссылке на команду.

Ответ: 1 (ОК 01-10, ПК 1.1-1.2)

16. При индексном доступе памяти адрес равен:

- 1) базовому адресу;
- 2) базовый адрес * смещение;
- 3) базовый адрес + смещение;
- 4) базовый адрес * К, где К — размер страницы.

Ответ: 3 (ОК 01-10, ПК 1.1-1.2)

17. Стек — это:

- 1) неявный способ адресации, при котором информация записывается и считывается только последовательным образом;
- 2) способ адресации, при котором информация записывается и считывается по принципу очереди;
- 3) неявный способ адресации, в котором информация записывается по принципу иерархий;
- 4) способ адресации, при котором информация записывается по старшинству;

Ответ: 1 (ОК 01-10, ПК 1.1-1.2)

18. Максимальное значение 16-разрядного регистра процессора в 16-й ОС?

- 1) 0ffh
- 2) 0ffffh
- 3) 0fffh
- 4) 0ffffffh

Ответ: 2 (ОК 01-10, ПК 1.1-1.2)

19. Назначение регистра IP?

- 1) счетчик команд;
- 2) указывают на сегмент;
- 3) хранение данных при вызовах.

Ответ: 1 (ОК 01-10, ПК 1.1-1.2)

20. Получение COM-файла из исходного модуля AA.asm:

- 1) tasm aa.asm
tlink aa.obj
- 2) tasm aa.asm
tlink aa.obj /t
- 3) tasm aa.asm /t
tlink aa.obj
- 4) tlink aa.obj
tasm aa.asm /t

Ответ: 2 (ОК 01-10, ПК 1.1-1.2)

21. Какой цвет определяет байт атрибут, имеющий значение 00001111b?

- 1) Чёрный фон, белый текст;
- 2) Белый фон, чёрный текст;
- 3) Тёмно синий фон, жёлтый текст;
- 4) Тёмно зелёный фон, белый текст.

Ответ: 1 (ОК 01-10, ПК 1.1-1.2)

22. Какой цвет определяет байт атрибут, имеющий значение 11110000b?

- 1) Чёрный фон, белый текст;
- 2) Белый фон, чёрный текст;
- 3) Тёмно синий фон, жёлтый текст;
- 4) Тёмно зелёный фон, белый текст.

Ответ: 2 (ОК 01-10, ПК 1.1-1.2)

23. Флаг (направление) регистра флагов, отвечающий за направление пересылки данных при выполнении строковых команд:

- 1) O
- 2) D
- 3) I
- 4) T

Ответ: 2 (ОК 01-10, ПК 1.1-1.2)

24. Флаг (пошаговый режим) регистра флагов, отвечающий за останов после выполнения каждой команды (используется отладчиками)

- 1) O
- 2) D
- 3) I
- 4) T

Ответ: 4 (ОК 01-10, ПК 1.1-1.2)

25. Флаг (ноль) регистра флагов, отвечающий за значение результата выполненной команды (0 – ненулевой, 1 – нулевой)

- 1) S

- 2) Z
- 3) A
- 4) P

Ответ: 2 (ОК 01-10, ПК 1.1-1.2)

Критерии оценки при проведении тестирования

«отлично»	«хорошо»	«удовлетворительно»	«неудовлетворительно»
студент выполнил 91-100% и набрал 23-25 баллов.	студент выполнил 76-90% и набрал 19-22 баллов.	студент выполнил 60-75% и набрал 15-18 баллов.	студент выполнил менее 0-59 % и набрал 0-14 баллов.

Задания к дифференцированному зачету

по МДК.01.01

(ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2)

1. Создать форму для ввода массива и кнопкой «Перемешать». При нажатии на кнопку элементы массива перемешиваются (ОК 01-10, ПК 1.1-1.2)
2. Палиндромом называют последовательность символов, которая читается как слева направо, так и справа налево. Создать web-форму для ввода строки и кнопкой, при нажатии на которую определяется, является ли введенная строка палиндромом (ОК 01-10, ПК 1.1-1.2)
3. Создать web-форму с двумя полями для ввода чисел и выбором операции (+, -, *, /). Вычислить соответствующее выражение и вывести результат в браузер (ОК 01-10, ПК 1.1-1.2)
4. Найти наибольшее из трех чисел, введенных пользователем в форму (ОК 01-10, ПК 1.1-1.2)
5. Написать программу нахождения синуса и косинуса угла, выбор действия осуществляется с помощью переключателя (ОК 01-10, ПК 1.1-1.2)
6. Средствами PHP вывести время и дату на странице (ОК 01-10, ПК 1.1-1.2)
7. Создать web-форму для ввода строки и кнопкой, при нажатии на которую строка переворачивается и выводится в браузер (ОК 01-10, ПК 1.1-1.2)
8. Написать программу, которая для введенной строки подсчитывает сумму всех содержащихся в ней цифр (ОК 01-10, ПК 1.1-1.2)
9. Создать класс Point, разработав следующие элементы класса: поля (x, y); конструкторы, позволяющие создать экземпляр класса с нулевыми координатами или с заданными координатами; методы, позволяющие вывести координаты точки на экран, рассчитать расстояние от начала координат до точки, переместить точку на плоскости на вектор (a, b); свойства для получения-установки координаты точки (доступное для чтения и записи), для умножения координаты точки на скаляр (доступное только для записи) (ОК 01-10, ПК 1.1-1.2)
10. Создать класс Triangle, разработав следующие элементы класса: поля (a, b, c); конструктор, позволяющий создать экземпляр класса с заданными длинами сторон; методы, позволяющие вывести длины сторон треугольника на экран, рассчитать периметр треугольника, рассчитать площадь треугольника; свойства для получения-установки длин сторон треугольника (доступное для чтения и записи), для определения, существует ли треугольник с данными длинами сторон (доступное только для чтения) (ОК 01-10, ПК 1.1-1.2)
11. Создать класс Rectangle, разработав следующие элементы класса: поля (a, b); конструктор, позволяющий создать экземпляр класса с заданными длинами сторон; методы, позволяющие вывести длины сторон прямоугольника на экран, рассчитать периметр прямоугольника, рассчитать площадь прямоугольника; свойства для получения-установки длин сторон прямоугольника (доступное для чтения и записи), для определения, является ли данный прямоугольник квадратом (доступное только для чтения) (ОК 01-10, ПК 1.1-1.2)
12. Создать класс Round, разработав следующие элементы класса: поля (r); конструктор, позволяющий создать экземпляр класса с заданным радиусом; методы, позволяющие рассчитать длину окружности, рассчитать площадь круга; свойства для получения-установки радиуса круга (доступное для чтения и записи) (ОК 01-10, ПК 1.1-1.2)
13. Создать абстрактный класс Figure с методами вычисления площади и периметра, а также методом, выводящим информацию о фигуре на экран. Создать производные классы: Triangle (треугольник), Circle (круг) со своими методами вычисления площади и периметра. Создать массив n фигур и вывести полную информацию о фигурах на экран (ОК 01-10, ПК 1.1-1.2)

14. Посчитать сумму элементов побочной диагонали квадратной матрицы (ОК 01-10, ПК 1.1-1.2)
15. Найти среднее арифметическое положительных элементов одномерного массива. Количество элементов массива вводится пользователем, массив заполняется случайными числами (ОК 01-10, ПК 1.1-1.2) (ОК 01-10, ПК 1.1-1.2)
16. Посчитать в строке количество цифр. Строка вводится пользователем с клавиатуры (ОК 01-10, ПК 1.1-1.2)
17. Создать форму Windows. Программа должна предоставлять возможность менять цвет фона формы (ОК 01-10, ПК 1.1-1.2)
18. Создать Windows-приложение, демонстрирующее работу с модальными и немодальными окнами (ОК 01-10, ПК 1.1-1.2)
19. Написать программу на языке C++, которая считает сумму элементов одномерного массива, не используя операцию индексирования []. (ОК 01-10, ПК 1.1-1.2)
20. Написать программу на языке C++, демонстрирующую способы передачи параметров в функции (ОК 01-10, ПК 1.1-1.2)
21. Написать рекурсивную функцию для нахождения факториала числа (ОК 01-10, ПК 1.1-1.2)
22. Написать программу на языке C++ для подсчета суммы элементов одномерного массива. Размерность массива вводится пользователем с клавиатуры (ОК 01-10, ПК 1.1-1.2)
23. Написать программу, демонстрирующую возможность создания односвязного списка и добавления элемента в список (ОК 01-10, ПК 1.1-1.2)
24. Написать программу, демонстрирующую возможность создания стека и добавления элемента в любое место стека (ОК 01-10, ПК 1.1-1.2)
25. Создать класс «Прямоугольник», который имеет два конструктора. Первый конструктор принимает в качестве параметров длину и ширину. Второй конструктор создает квадрат с заданной стороной (ОК 01-10, ПК 1.1-1.2)
26. Написать программу, демонстрирующую работу с абстрактными классами (ОК 01-10, ПК 1.1-1.2)
27. Создать функцию на языке C++, подсчитывающую сумму элементов массива. Массив передается как параметр в функцию (ОК 01-10, ПК 1.1-1.2)
28. Написать программу для нахождения максимального элемента матрицы. (ОК 01-10, ПК 1.1-1.2)
29. Написать программу для сортировки массива методом пузырька. (ОК 01-10, ПК 1.1-1.2)
30. Написать программу для нахождения корней квадратного уравнения $ax^2+bx+c=0$. Коэффициенты a, b, c вводятся пользователем (ОК 01-10, ПК 1.1-1.2)

Критерии оценки при проведении дифференцированного зачета

«отлично»	«хорошо»	«удовлетворительно»	«неудовлетворительно»
студент раскрывает теоретический вопрос билета, практическое задание выполняет без ошибок, уверенно отвечает на дополнительные вопросы	студент раскрывает теоретический вопрос, практическое задание выполняет без ошибок, на дополнительные вопросы отвечает неуверенно, допускает не точности в определениях.	студент раскрывает теоретический вопрос не в полной мере, допускает неточности в формулировках (1-2 ошибки), практическое задание выполнено частично, с допущением ошибок в расчётах	Теоретический вопрос не раскрыт, практическое задание не выполнено.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Елабужский институт (филиал) федерального государственного автономного образовательного учреждения высшего образования
«Казанский (Приволжский) федеральный университет»

**Вопросы к экзамену
по МДК.01.01**

(ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2)

1. Системное программирование. **(ОК 01-10, ПК 1.1-1.2)**

Ответ: Системное программирование (или программирование систем) — род деятельности, заключающийся в работе над системным программным обеспечением.

2. Системное ПО. **(ОК 01-10, ПК 1.1-1.2)**

Ответ: Системное программное обеспечение — это комплекс программ, которые обеспечивают эффективное управление компонентами вычислительной системы, такими как процессор, оперативная память, каналы ввода-вывода, сетевое оборудование, выступая как «межслойный интерфейс» с одной стороны которого аппаратура, а с другой приложения пользователя. В отличие от прикладного программного обеспечения, системное не решает конкретные прикладные задачи, а лишь обеспечивает работу других программ, управляет аппаратными ресурсами вычислительной системы и т.д.

3. Формализация задачи и разработка алгоритма. **(ОК 01-10, ПК 1.1-1.2)**

Ответ: Формализация задачи - это описание задачи формальными средствами (математическими, логическими). Алгоритм - последовательность действий, необходимых для решения задачи. Разработка алгоритмов и программ начинается с постановки задач. На этом этапе из условия задачи определяют входную, выходную и промежуточную информацию. На первой стадии разработки программы были описаны все переменные и их типы. Следующим шагом стала разработка блока (процедуры) ввода исходных данных. Ввод исходных данных включает в себя создание файла для хранения данных и его заполнения.

4. ЭВМ - исполнитель алгоритмов. **(ОК 01-10, ПК 1.1-1.2)**

Ответ: Работа компьютера имитирует (моделирует) информационную деятельность. Но компьютер - техническое устройство, поэтому для того, чтобы он выполнил определенные действия, им нужно управлять. Компьютер действует как автоматический формальный исполнитель алгоритмов обработки информации. Это главное свойство любого компьютера. Автоматизм в его работе означает, что некоторые свои действия он выполняет без вмешательства человека в соответствии со следующими общими принципами: двоичное кодирование информации; программное управление работой компьютера (идея Чарльза Бэббиджа) хранимая программа (по предложению американского математика Джона фон Неймана (1930-1959) наряду с данными каждая команда программы работы компьютера, кодируемая определенной последовательностью из нулей и единиц, помещается как число в одной из ячеек оперативной памяти.).

5. Жизненный цикл ПО. Основные этапы разработки ПО. **(ОК 01-10, ПК 1.1-1.2)**

Ответ: Жизненный цикл программного обеспечения (ПО) — период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации. Процесс разработки программного обеспечения можно разбить на этапы (фазы). Первый этап – постановка задачи. Второй этап - выбор метода решения. Третий этап - разработка алгоритма решения задачи. Четвертый этап – кодирование алгоритма. Пятый этап – трансляция и компиляция программы. Шестой этап – тестирование программы. Седьмой этап – создание документации. Восьмой этап - сопровождение и эксплуатация.

6. Модели жизненного цикла программного средства. **(ОК 01-10, ПК 1.1-1.2)**

Ответ: Каскадная модель - реализует принцип однократного выполнения каждого вида деятельности в виде заранее ограниченных и однозначно упорядоченных во времени стадий и этапов, осуществляемых как бы в их естественных границах.

Эволюционная модель разработки – т.е. разрабатывается первоначальная версия программного продукта, которая передается на испытание пользователям. Затем эта версия дорабатывается с учетом мнения пользователей, и получается новая (промежуточная) версия продукта. Эта версия также проходит «испытание пользователем». Снова дорабатывается, и так несколько раз, пока не будет получен необходимый программный продукт. Отличительной особенностью этой модели является то, что процессы специфицирования, разработки и аттестации ПО выполняются параллельно при постоянном обмене информацией.

Модель формальной разработки - этот подход основан на формальной математической спецификации программного средства и преобразований этой спецификации посредством специальных математических методов в исполняемые программы.

Модель разработки ПС на основе ранее созданных компонентов - эта модель предполагает, что отдельные составные части программной системы уже существуют. В этом случае основное внимание уделяется интеграции отдельных компонентов в общее целое.

7. Постановка задачи на разработку программного средства. (ОК 01-10, ПК 1.1-1.2)

Ответ: Постановка задачи (problem definition) - это точная формулировка результата решения задачи на компьютере с описанием входной и выходной информации. Входная информация определяется совокупностью данных, поступающих на вход задачи и используемых для ее решения. Входной информацией служат первичные данные документов ручного заполнения, информация, хранимая в файлах базы данных (результаты решения других задач, нормативно-справочная информация – классификаторы, кодификаторы, справочники), входные сигналы датчиков. Выходная информация по задаче может быть представлена в виде документов (типа распечатки отчётов, заполненных форм), сформированных кадров (видеограммы на экране монитора), файла базы данных, выходного сигнала для дальнейшего использования в устройстве управления.

8. Составление программы на языке программирования (ОК 01-10, ПК 1.1-1.2)

Ответ: Основные этапы компьютерного моделирования: постановка задачи, алгоритмизация, кодирование, комплексное тестирование и отладка, а также сопровождение.

9. Структура и способы описания языков программирования высокого уровня. (ОК 01-10, ПК 1.1-1.2)

Ответ: Язык высокого уровня - это язык программирования, предназначенный для удовлетворения требований программиста; он не зависит от внутренних машинных кодов компьютера любого типа. Языки высокого уровня используют для решения проблем и поэтому их часто называют проблемно-ориентированными языками. Каждая команда языка высокого уровня эквивалентна нескольким командам в машинных кодах, поэтому программы, написанные на языках высокого уровня, более компактны, чем аналогичные программы в машинных кодах. Всякий язык программирования имеет три основные составляющие: алфавит, синтаксис и семантику. Синтаксис языка описывается путем последовательного усложнения понятий: сначала определяются простейшие (базовые), затем все более сложные, включающие в себя предыдущие понятия в качестве составляющих. Смысловое содержание языковой конструкции называется семантикой.

10. Подпрограмма – процедура.(ОК 01-10, ПК 1.1-1.2)

Ответ: Подпрограмма – относительно самостоятельная часть программы, имеющая свое имя и выполняющая определенные действия. Подпрограммы повышают надежность и наглядность программ, т.к. позволяют разрабатывать и отлаживать каждый блок программы независимо (например, разными людьми).

Структура подпрограммы почти полностью повторяет структуру всей программы и состоит из следующих частей: заголовок подпрограммы, раздел описаний, тело подпрограммы. В Паскаль имеются два вида подпрограмм – процедуры и функции. Они отличаются назначением и способом их использо-

вания. Процедуры служат для выполнения определенной последовательности действий, направленных на изменение программной обстановки (изменение значений переменных, ввод/вывод данных и т.п.). Функции для вычисления значения выражения. Подпрограмма – процедура имеет вид:
procedure <имя> (список формальных параметров);
<раздел описаний>;
begin
<тело процедуры>;
end.

11. Формальные и фактические параметры.(ОК 01-10, ПК 1.1-1.2)

Ответ: Формальные параметры - это это переменные, указанные в заголовке подпрограммы. Их часто называют просто параметры.

Фактические параметры - это список переменных (констант, выражений), указанных в операторе вызова подпрограммы (то есть передаваемых в подпрограмму при вызове). Их также называют аргументами.

12. Локальные и глобальные переменные.(ОК 01-10, ПК 1.1-1.2)

Ответ: Локальные переменные – это переменные, определенные внутри какого-либо блока. Область действия локальной переменной – от места ее описания и до конца блока, в котором она определена, т.е. за пределами этого блока она не существует. Время жизни локальной переменной равно времени выполнения блока – при завершении блока все его локальные переменные уничтожаются и память освобождается. Другими словами, локальные переменные предназначены для временного хранения значений и это время определяется длительностью выполнения некоторого фрагмента программы.

Глобальные переменные – это переменные, определенные вне какого-либо блока, т.е. в глобальном пространстве вне какой-либо функции. Такие переменные действуют с момента их описания и до конца программы (т.е. во всех блоках на этом интервале), а время их жизни равно времени работы программы.

13. Подпрограмма функция.(ОК 01-10, ПК 1.1-1.2)

Ответ: Подпрограмма-функция – это подпрограмма, результат которой представляет собой единственное значение, присваиваемое имени этой функции. Обращение к функции является операндом в выражении. Оно записывается в следующей форме:

<Имя функции> (<Список фактических параметров>)

14. Разработка программного продукта с использованием подпрограммы процедуры.

(ОК 01-10, ПК 1.1-1.2)

Ответ: Прежде чем использовать процедуру, ее нужно описать в разделе описаний основной программы.

В разделе описаний процедуры и функции определяют после раздела описания переменных (после раздела Var) и до начала раздела операторов (до слова Begin). Структура процедуры в языке Паскаль похожа на структуру основной программы: в ней есть заголовок, раздел описаний и раздел операторов. Заголовок процедуры начинается с ключевого слова Procedure за которым следует имя процедуры и необязательный, заключенный в круглые скобки, список формальных параметров с указанием типа каждого параметра.

Общий вид процедуры:

Procedure <имя>[<список формальных параметров с указанием типов>;

<раздел описаний>

Begin

<операторы>

End;

Параметры, перечисленные в скобках после имени процедуры, называются формальными. Их назначение — указать количество и тип переменных, с помощью которых будет осуществляться обмен информацией между подпрограммой и основной программой. Формальные параметры определяют место подстановки фактических параметров, над которыми и производятся действие. Через парамет-

ры-значения в подпрограмму поступает входная информация. Перед ними слово `Var` не ставится и идет передача по значению, то есть передается только копия значений этих параметров. Через параметры- переменные из подпрограммы в основную программу передаются результаты. Перед параметрами-переменными ставится служебное слово `Var`.

15. Модульное программирование. (ОК 01-10, ПК 1.1-1.2)

Ответ: Модульное программирование — это организация программы как совокупности небольших независимых блоков, называемых модулями, структура и поведение которых подчиняются определенным правилам. Использование модульного программирования позволяет упростить тестирование программы и обнаружение ошибок. Аппаратно-зависимые подзадачи могут быть строго отделены от других подзадач, что улучшает мобильность создаваемых программ.

16. Решение задач с использованием стека. (ОК 01-10, ПК 1.1-1.2)

Ответ: Стек - это структура данных, которая хранит элементы упорядоченно и умеет отвечать на две операции за:

- `push(x)` - положить элемент `x` в конец стека
- `pop()` - снять и вернуть элемент, лежащий в конце стека

То есть это структура данных, где действия происходят только с элементом, лежащим в конце. Выполняется принцип `FIFO (First In - Last Out)` - последним вынется тот элемент, который мы положили первым, если сначала положить все элементы, а потом все вынуть. Часто для удобства у стека еще есть операции * `size()` - размер стека * `empty()` - проверка на пустоту * `clear()` - очистить стек. Как стек удобно использовать динамический массив: * в Питоне это `list`, причем `push = append`, `pop = pop` * в `C++` это `vector`, причем `push = push_back`, а операция `pop` заменяется на две - `back` возвращает последний элемент, а `pop_back` вынимает его

17. Реализация процедур и функций работы с бинарным деревом. (ОК 01-10, ПК 1.1-1.2)

Ответ: Бинарное дерево или двоичное дерево — это дерево, в котором у каждого из его узлов не более двух дочерних узлов. При этом каждый дочерний узел тоже представляет собой бинарное дерево. Бинарные деревья поиска отличаются от обычных бинарных деревьев тем, что хранят данные в отсортированном виде. С бинарными деревьями поиска можно выполнять следующие операции: искать узел, вставлять узел, удалять узел, выполнять обход дерева.

18. Разработка программного продукта с использованием модуля. (ОК 01-10, ПК 1.1-1.2)

Ответ: Практически все сложное программное обеспечение имеет модульную структуру, то есть состоит из программных модулей, при этом разработка программных модулей программного обеспечения может быть выполнена:

1. Как составляющая часть процесса разработки ПО.

2. В процессе доработки, усовершенствования или модернизации ранее созданного ПО.

В программном комплексе созданные модули могут представлять собой: законченные компоненты; функциональные группы; пакеты программ; единый программный комплекс – законченный программный продукт конкретного целевого назначения.

Каждый программный модуль может быть спроектирован и разработан отдельно от других модулей. Более того, он может представлять собой самостоятельный программный продукт и быть включенным в структуру разных программ, не обязательно тех, для которых разрабатывался. Такая характеристика показывает, что разработка программных модулей программного обеспечения призвана решать не только задачу упрощения программирования, но и борьбы с дублированием – нет смысла разрабатывать то, что уже создано и может быть использовано в конкретной структуре ПО.

19. Выбор языка программирования. (ОК 01-10, ПК 1.1-1.2)

Ответ: Выбор языка программирования зависит от поставленной задачи в определенной сфере деятельности. Веб-программирование - JavaScript; HTML; CSS. Разработка игр - Unreal Engine и C++; Unity и C#; Godot и Python. Разработка ПО - для Linux или Windows подойдет Си, C++, C#, Java; для

Mac лучше выбрать Си, Swift или Java. В области анализа данных используются преимущественно специализированные языки – Scala или R.

20. Основы языка UML (Unified Modeling Language).(ОК 01-10, ПК 1.1-1.2)

Ответ: UML представляет собой объектно-ориентированный язык моделирования, обладающий следующими основными характеристиками: является языком визуального моделирования, который обеспечивает разработку репрезентативных моделей для организации взаимодействия заказчика и разработчика информационной системы (ИС), различных групп разработчиков ИС; содержит механизмы расширения и специализации базовых концепций языка.

21. Создание абстрактных типов данных. Диаграмма объекта. (ОК 01-10, ПК 1.1-1.2)

Ответ: Принцип абстрагирования предписывает включать в модель только те аспекты проектируемой системы, которые имеют непосредственное отношение к выполнению системой своих функций или своего целевого предназначения. При этом все второстепенные детали опускаются, чтобы чрезмерно не усложнять процесс анализа и исследования полученной модели. Диаграмма объектов — это структурная диаграмма UML, которая является экземпляром диаграммы классов и представляет собой снимок детального состояния системы в конкретный момент времени, позволяет увидеть, какие объекты существуют в системе и как они взаимодействуют друг с другом. Диаграммы объектов предназначена для визуализации объектов, их атрибутов и взаимосвязи между ними. Это позволяет лучше понять структуру системы, определить, какие объекты существуют в системе, и как они взаимодействуют друг с другом.

23. Принципы объектноориентированного анализа: абстрагирование, инкапсуляция, наследование, полиморфизм, модульность, сохраняемость, параллелизм (ОК 01-10, ПК 1.1-1.2)

Ответ: Абстрагирование выделяет существенные характеристики некоторого объекта, отличающие его от всех других видов объектов и, таким образом, четко определяет его концептуальные границы с точки зрения наблюдателя.

Наследование в объектно-ориентированном программировании — это концепция, согласно которой одни классы, называемые родительскими, могут лежать в основе других — дочерних.

Инкапсуляция — это процесс отделения друг от друга элементов объекта, определяющих его устройство и поведение; инкапсуляция служит для того, чтобы изолировать контрактные обязательства абстракции от их реализации.

Полиморфизм в контексте ООП означает, что разные объекты могут реагировать на один и тот же запрос, проявляя разное поведение в зависимости от своего типа.

Модульность. Разделение программы на модули позволяет уменьшить ее сложность, однако гораздо важнее тот факт, что внутри модульной программы создаются множества хорошо определенных и документированных интерфейсов, которые важны для исчерпывающего понимания программы в целом.

Параллелизм – это свойство, отличающее активные объекты от неактивных.

Сохраняемость – это способность объекта существовать во времени, переживая породивший его процесс, и (или) в пространстве, перемещаясь из своего первоначального адресного пространства. Любой программный объект существует в памяти и живет во времени.

24. Структура программы на языке C++. Проект.(ОК 01-10, ПК 1.1-1.2)

Ответ: Программа на языке C/C++ состоит из: директив процессора, описаний, функций

Общую структуру программы на языке C/C++ можно представить следующим образом:

```
директивы препроцессора //1
описание глобальных переменных //2
тип_результата main (параметры) //3
{
тело главной функции
}
тип_результата F1 (параметры 1) {
```

```

тело функции F1;
}
тип_результата F2 (параметры 2) {
тело функции F2;
}
тип_результата FN (параметры N) {
тело функции FN;
}

```

Директивы препроцессора – определяют действия по преобразованию программы перед компиляцией, а также включают инструкции, которым компилятор следует во время компиляции;

Объявления – описания переменных, функций, структур, классов и типов данных;

Описание функций, выполняемых в проекте. В языке C++ существует только один вид подпрограмм – подпрограмма-функция.

25. Компиляция программы и сборка исполняемого модуля. (ОК 01-10, ПК 1.1-1.2)

Ответ: Компиляция (compilation) – это преобразование программы или ее отдельного модуля, текст которых составлен на языке программирования высокого уровня (исходная программа, исходный модуль – это файл с расширением .cpp) в программу или модуль на машинном языке или на языке, близком к машинному (получают объектный модуль – файл с расширением .obj). Компиляцию осуществляет специальная программа – компилятор (compiler), которая является неотъемлемой частью системы программирования. На вход компилятора поступает исходный модуль (файл .cpp), который после компиляции преобразуется в объектный модуль (файл .obj).

Компоновка, редактирование связей (linking, linking editing) – это процесс сборки загрузочного модуля (исполняемого файла) из полученных в результате отдельной компиляции объектных модулей с одновременным автоматическим поиском и присоединением библиотечных подпрограмм и процедур. В процессе компоновки программа собирается в единое целое непосредственно в оперативной памяти в файл, готовый к работе (загрузочный модуль - файл с расширением .exe).

26. Размещение программы и данных в памяти. (ОК 01-10, ПК 1.1-1.2)

Ответ: Данные и программы во время работы ПЭВМ размещаются в оперативной памяти, которая представляет собой последовательность пронумерованных ячеек. По указанному номеру процессор находит нужную ячейку, поэтому номер ячейки называется ее адресом. Минимальная адресованная ячейка (согласно стандарту IBM), с точки зрения программиста, состоит из 8 двоичных позиций, т.е. в каждую позицию могут быть записаны либо 0, либо 1. Объем информации, который помещается в одну двоичную позицию, называется бит. Объем информации равный 8 бит называется байтом. Таким образом, в одной ячейке из 8 двоичных разрядов помещается объем информации в один байт. Поэтому объем памяти принято оценивать количеством байт (1024 Байт = 1 Килобайт, 1024 Килобайт = 1048576 Байт = 1 Мегабайт, 1024 Мегабайт = 1073741824 Байт = 1 Гигабайт.). Составление программ в современных компьютерах автоматизировано. Программист пишет программу на специальном языке высокого уровня, т.е. наиболее удобном для записи алгоритма решения определенного класса задач. С помощью символов клавиатуры вводит ее текст в память компьютера. Далее происходит перевод алгоритма в команды машины (трансляция) под-ключение необходимых стандартных программ (компоновка) и лишь после этого выполнение. Заметим, что полученная программа в кодах машины может быть записана на диск и для ее многократного выполнения уже не требуется этапов трансляции и компоновки.

27. Структура исполняемого модуля. (ОК 01-10, ПК 1.1-1.2)

Ответ: Большие программные системы обычно состоят из нескольких файлов - исходных модулей. Каждый исходный модуль компилируется отдельно, а потом полученные объектные модули объединяются компоновщиком в один исполняемый модуль.

Модуль имеет следующую структуру:

```

Unit <имя>;
interface

```

<интерфейсная часть> implementation
<исполняемая часть> initialization <иницирующая часть> finalization <завершающая часть>
end.

Здесь unit - зарезервированное слово (единица); начинает заголовок модуля; <имя> - имя модуля (правильный идентификатор); interface - зарезервированное слово (интерфейс); начинает интерфейсную часть модуля; implementation - зарезервированное слово (выполнение); начинает исполняемую часть; initialization - зарезервированное слово (инициация); начинает иницирующую часть модуля; finalization - зарезервированное слово (завершение); начинает завершающую часть модуля; end - зарезервированное слово - признак конца модуля. Таким образом, модуль состоит из заголовка и четырех составных частей, любая из которых может быть пустой.

28. Стандартная библиотека функций языка C++.(ОК 01-10, ПК 1.1-1.2)

Ответ: В языке программирования C++ термин Стандартная Библиотека означает коллекцию классов и функций, написанных на базовом языке. Стандартная Библиотека поддерживает несколько основных контейнеров, функций для работы с этими контейнерами, объектов-функции, основных типов строк и потоков (включая интерактивный и файловый ввод-вывод), поддержку некоторых языковых особенностей, и часто используемые функции для выполнения таких задач, как, например, нахождение квадратного корня числа. Стандартная Библиотека языка C++ также включает в себя спецификации стандарта ISO C90 стандартной библиотеки языка Си. Функциональные особенности Стандартной Библиотеки объявляются внутри пространства имен std.

Стандартная библиотека шаблонов (STL) — подмножество стандартной библиотеки C++ и содержит контейнеры, алгоритмы, итераторы, объекты-функции и т. д., хотя некоторые программисты используют термин «STL» вместе (или попеременно) с термином «Стандартная библиотека C++».

29. Компиляция программы и сборка исполняемого модуля. (ОК 01-10, ПК 1.1-1.2)

Ответ: Компиляция (compilation) – это преобразование программы или ее отдельного модуля, текст которых составлен на языке программирования высокого уровня (исходная программа, исходный модуль – это файл с расширением .cpp) в программу или модуль на машинном языке или на языке, близком к машинному (получают объектный модуль – файл с расширением .obj). Компиляцию осуществляет специальная программа –компилятор (compiler), которая является неотъемлемой частью системы программирования. На вход компилятора поступает исходный модуль (файл .cpp), который после компиляции преобразуется в объектный модуль (файл .obj).

Компоновка, редактирование связей (linking, linking editing) – это процесс сборки загрузочного модуля (исполняемого файла) из полученных в результате отдельной компиляции объектных модулей с одновременным автоматическим поиском и присоединением библиотечных подпрограмм и процедур. В процессе компоновки программа собирается в единое целое непосредственно в оперативной памяти в файл, готовый к работе (загрузочный модуль - файл с расширением .exe).

30.Размещение программы и данных в памяти.(ОК 01-10, ПК 1.1-1.2)

Ответ: Данные и программы во время работы ПЭВМ размещаются в оперативной памяти, которая представляет собой последовательность пронумерованных ячеек. По указанному номеру процессор находит нужную ячейку, поэтому номер ячейки называется ее адресом. Минимальная адресованная ячейка (согласно стандарту IBM), с точки зрения программиста, состоит из 8 двоичных позиций, т.е. в каждую позицию могут быть записаны либо 0, либо 1. Объем информации, который помещается в одну двоичную позицию, называется бит. Объем информации равный 8 бит называется байтом. Таким образом, в одной ячейке из 8 двоичных разрядов помещается объем информации в один байт. Поэтому объем памяти принято оценивать количеством байт (1024 Байт = 1 Килобайт, 1024 Килобайт = 1048576 Байт = 1 Мегабайт, 1024 Мегабайт = 1073741824 Байт = 1 Гигабайт.). Составление программ в современных компьютерах автоматизировано. Программист пишет программу на специальном языке высокого уровня, т.е. наиболее удобном для записи алгоритма решения определенного класса задач. С помощью символов клавиатуры вводит ее текст в память компьютера. Далее происходит перевод алгоритма в команды машины (трансляция) под-ключение необходимых стандартных программ (компоновка) и лишь после этого выполнение. Заметим, что полученная программа в кодах

машины может быть записана на диск и для ее многократного выполнения уже не требуется этапов трансляции и компоновки.

31. Виртуальные функции и абстрактные базовые классы. (ОК 01-10, ПК 1.1-1.2)

Ответ: Виртуальная функция — это функция-член, которую предполагается переопределить в производных классах. При ссылке на объект производного класса с помощью указателя или ссылки на базовый класс можно вызвать виртуальную функцию для этого объекта и выполнить версию функции производного класса. Виртуальные функции обеспечивают вызов соответствующей функции для объекта независимо от выражения, используемого для вызова функции.

Абстрактные классы - это классы, которые предназначены для наследования, но избегают реализации конкретных методов, оставляя только сигнатуры методов, которые должны реализовывать подклассы. Абстрактные классы полезны для определения и реализации абстракций классов на высоком уровне, аналогично концепции интерфейсов в типизированных языках, без необходимости реализации метода.

32. Множественное наследование. (ОК 01-10, ПК 1.1-1.2)

Ответ: Множественное наследование — свойство, поддерживаемое частью объектно-ориентированных языков программирования, когда класс может иметь более одного суперкласса (непосредственного класса-родителя), интерфейсы поддерживают множественное наследование во многих языках программирования. Эта концепция является расширением «простого (или одиночного) наследования» (англ. single inheritance), при котором класс может наследоваться только от одного суперкласса. В список языков, поддерживающих множественное наследование, входят: Io, Eiffel, C++, Dylan, Python, некоторые реализации классов JavaScript (например, dojo.declare), Perl 6, Curl, Common Lisp (благодаря CLOS), OCaml, Tcl (благодаря Incremental Tcl)[1], а также Object REXX (за счёт использования классов-примесей).

33. Контейнеры и итераторы в библиотеке STL (Standard Template Library). (ОК 01-10, ПК 1.1-1.2)

Ответ: Итераторы представляют собой объекты, обеспечивающие перемещение по последовательности и доступ к ее элементам (можно сказать, что итератор – это абстракция понятия указателя). Хотя обычно итераторы используются при работе с контейнерами; наиболее простые виды итераторов – итераторы для чтения и итераторы для записи – могут применяться в программе для доступа к потокам чтения/записи, поэтому начать знакомство с итераторами можно без предварительного изучения контейнеров STL (которым посвящены группы STL2Seq, STL4Str и STL5Assoc).

34. Ассоциативные массивы. (ОК 01-10, ПК 1.1-1.2)

Ответ: Ассоциативный массив — абстрактный тип данных (интерфейс к хранилищу данных), позволяющий хранить пары вида «(ключ, значение)» и поддерживающий операции добавления пары, а также поиска и удаления пары по ключу:

INSERT(ключ, значение)

FIND(ключ)

REMOVE(ключ)

35. Разработка кода программного модуля на современных языках программирования (ОК 01-10, ПК 1.1-1.2)

Ответ: При разработке программного модуля целесообразно придерживаться следующего порядка: изучение и проверка спецификации модуля, выбор языка программирования; выбор алгоритма и структуры данных; программирование (кодирование) модуля; шлифовка текста модуля; проверка модуля; компиляция модуля.

Первый шаг разработки программного модуля в значительной степени представляет собой смежный контроль структуры программы снизу: изучая спецификацию модуля, разработчик должен убедиться, что она ему понятна и достаточна для разработки этого модуля. На втором шаге разработки программного модуля необходимо выяснить, не известны ли уже какие-либо алгоритмы для решения поставленной и или близкой к ней задачи. На третьем шаге осуществляется построение текста модуля

на выбранном языке программирования. Следующий шаг разработки модуля связан с приведением текста модуля к завершённому виду в соответствии со спецификацией качества ПС. Шаг проверки модуля представляет собой ручную проверку внутренней логики модуля до начала его отладки (использующей выполнение его на компьютере), реализует общий принцип, сформулированный для обсуждаемой технологии программирования, о необходимости контроля принимаемых решений на каждом этапе разработки ПС. Последний шаг разработки модуля означает завершение проверки модуля (с помощью компилятора) и переход к процессу отладки модуля.

Критерии оценки при проведении экзамена

«отлично»	«хорошо»	«удовлетворительно»	«неудовлетворительно»
студент раскрывает теоретический вопрос билета, практическое задание выполняет без ошибок, уверенно отвечает на дополнительные вопросы	студент раскрывает теоретический вопрос, практическое задание выполняет без ошибок, на дополнительные вопросы отвечает неуверенно, допускает неточности в определениях.	студент раскрывает теоретический вопрос не в полной мере, допускает неточности в формулировках (1-2 ошибки), практическое задание выполнено частично, с допущением ошибок в расчётах	Теоретический вопрос не раскрыт, практическое задание не выполнено.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Елабужский институт (филиал) федерального государственного автономного
образовательного учреждения высшего образования
«Казанский (Приволжский) федеральный университет»

УТВЕРЖДАЮ

Заместитель директора по
образовательной деятельности
Елабужского института ЕИ КФУ

И. П. Михайлова

«01» марта 2024 г.

МП



ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

МДК.01.02 Поддержка и тестирование программных модулей

(наименование дисциплины)

09.02.07 Информационные системы и программирование

(код и наименование специальности)

Программист

(квалификация выпускника)

г.Елабуга, 2024

**Паспорт
фонда оценочных средств по
МДК.01.02 Поддержка и тестирование программных модулей**
(наименование междисциплинарного курса)

Результаты (освоенные профессиональные компетенции)	Основные показатели оценки результата	Формы и методы контроля и оценки
ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;</p> <p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;</p>	Отчет по лабораторным работам. Тестовые задания Экзамен (вопросы 1-18)
ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;</p> <p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;</p>	Отчет по лабораторным работам. Тестовые задания Экзамен (вопросы 1-18)
ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;</p> <p>Знать: основные этапы разработки программного обеспечения; основ-</p>	Отчет по лабораторным работам. Тестовые задания Экзамен (вопросы 1-18)

	ные принципы технологии структурного и объектно-ориентированного программирования;	
ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	Отчет по лабораторным работам. Тестовые задания Экзамен (вопросы 1-18)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	Отчет по лабораторным работам. Тестовые задания Экзамен (вопросы 1-18)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	Отчет по лабораторным работам. Тестовые задания Экзамен (вопросы 1-18)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ОК 07. Содействовать сохранению окружающей среды, ресурсосбере-	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алго-	Отчет по лабораторным работам. Тестовые задания Экзамен (вопросы 1-18)

жению, эффективно действовать в чрезвычайных ситуациях	ритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	Отчет по лабораторным работам. Тестовые задания Экзамен (вопросы 1-18)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ОК. 09. Использовать информационные технологии в профессиональной деятельности	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	Отчет по лабораторным работам. Тестовые задания Экзамен (вопросы 1-18)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ОК.10. Пользоваться профессиональной документацией на государственном и иностранном языках	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	Отчет по лабораторным работам. Тестовые задания Экзамен (вопросы 1-18)

	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств	Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений	Отчет по лабораторным работам. Тестовые задания Экзамен (вопросы 1-18)
	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	
ПК 1.4. Выполнять тестирование программных модулей	Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений	Отчет по лабораторным работам. Тестовые задания Экзамен (вопросы 1-18)
	Уметь: осуществлять разработку кода	

	<p>программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p>	
	<p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	
<p>ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода</p>	<p>Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений</p> <p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p> <p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тести-</p>	<p>Отчет по лабораторным работам. Тестовые задания Экзамен (вопросы 1-18)</p>

	рования программных продуктов	
--	-------------------------------	--

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
 Елабужский институт (филиал) федерального государственного автономного
 образовательного учреждения высшего образования
 «Казанский (Приволжский) федеральный университет»

**Отчет по лабораторным работам
 по МДК.01.02 Поддержка и тестирование программных модулей
 (ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ПК 1.3, ПК 1.4, ПК 1.5)**

**Тема 2.1 Отладка и тестирование программного обеспечения
 (ОК 01-09, ПК 1.3-1.5)**

Тестирование «белым ящиком»
 Тестирование «черным ящиком»
 Модульное тестирование
 Интеграционное тестирование

Тема 2.2 Документирование

Оформление документации на программные средства с использованием инструментальных средств.
 (ОК 01-09, ПК 1.3-1.5)

Индекс и расшифровка компетенции	Результаты обучения	Критерии оценивания результатов обучения (баллы)			
		2	3	4	5
ОК-1	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документа-	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений

	цию на программные средства				
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
ОК-2	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений
	Знать: основные этапы разработки	Не знает Допускает	Демонстрирует частичные	Знает достаточно в ба-	Демонстрирует высо-

	<p>программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	<p>грубые ошибки</p>	<p>знания без грубых ошибок</p>	<p>зовом объеме</p>	<p>кий уровень знаний</p>
<p>ОК- 3</p>	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p>	<p>Не умеет Демонстрирует частичные умения, допуская грубые ошибки</p>	<p>Демонстрирует частичные умения без грубых ошибок</p>	<p>Умеет применять знания на практике в базовом объеме</p>	<p>Демонстрирует высокий уровень умений</p>
	<p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-</p>	<p>Не знает Допускает грубые ошибки</p>	<p>Демонстрирует частичные знания без грубых ошибок</p>	<p>Знает достаточно в базовом объеме</p>	<p>Демонстрирует высокий уровень знаний</p>

	но-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов				
ОК-4	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и при-	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний

	емы рефакторинга; основные принципы отладки и тестирования программных продуктов				
ОК-5	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний

	продуктов				
ОК- 6	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
ОК- 7	Уметь: осуществлять разработку кода программного модуля на язы-	Не умеет Демонстрирует частичные умения,	Демонстрирует частичные умения без грубых оши-	Умеет применять знания на практике в базо-	Демонстрирует высокий уровень умений

	<p>как низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p>	<p>допуская грубые ошибки</p>	<p>бок</p>	<p>вом объёме</p>	
	<p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	<p>Не знает Допускает грубые ошибки</p>	<p>Демонстрирует частичные знания без грубых ошибок</p>	<p>Знает достаточно в базовом объёме</p>	<p>Демонстрирует высокий уровень знаний</p>
<p>ОК- 8</p>	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алго-</p>	<p>Не умеет Демонстрирует частичные умения, допуская грубые ошибки</p>	<p>Демонстрирует частичные умения без грубых ошибок</p>	<p>Умеет применять знания на практике в базовом объёме</p>	<p>Демонстрирует высокий уровень умений</p>

	<p>ритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p>				
	<p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	<p>Не знает Допускает грубые ошибки</p>	<p>Демонстрирует частичные знания без грубых ошибок</p>	<p>Знает достаточно в базовом объеме</p>	<p>Демонстрирует высокий уровень знаний</p>
<p>ОК- 9</p>	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне</p>	<p>Не умеет Демонстрирует частичные умения, допуская грубые ошибки</p>	<p>Демонстрирует частичные умения без грубых ошибок</p>	<p>Умеет применять знания на практике в базовом объеме</p>	<p>Демонстрирует высокий уровень умений</p>

	модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства				
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
ОК- 10	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений

	языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства				
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
ПК- 1.3	Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке	Не владеет Демонстрирует низкий уровень владения, допуская грубые ошибки	Демонстрирует частичные владения без грубых ошибок	Владеет базовыми приемами	Демонстрирует владения на высоком уровне

	мобильных приложений				
	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
ПК 1.4	Иметь практический опыт: разработка кода про-	Не владеет Демонстрирует низкий	Демонстрирует частичные владения без	Владеет базовыми приёмами	Демонстрирует владения на высо-

	граммного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений	уровень владения, допуская грубые ошибки	грубых ошибок		ком уровне
	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений

	<p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	<p>Не знает Допускает грубые ошибки</p>	<p>Демонстрирует частичные знания без грубых ошибок</p>	<p>Знает достаточно в базовом объеме</p>	<p>Демонстрирует высокий уровень знаний</p>
ПК-1.5	<p>Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений</p>	<p>Не владеет Демонстрирует низкий уровень владения, допуская грубые ошибки</p>	<p>Демонстрирует частичные владения без грубых ошибок</p>	<p>Владеет базовыми приёмами</p>	<p>Демонстрирует владения на высоком уровне</p>
	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разра-</p>	<p>Не умеет Демонстрирует частичные умения, допуская грубые ошибки</p>	<p>Демонстрирует частичные умения без грубых ошибок</p>	<p>Умеет применять знания на практике в базовом объеме</p>	<p>Демонстрирует высокий уровень умений</p>

	<p>ботанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p>				
	<p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	<p>Не знает Допускает грубые ошибки</p>	<p>Демонстрирует частичные знания без грубых ошибок</p>	<p>Знает достаточно в базовом объеме</p>	<p>Демонстрирует высокий уровень знаний</p>

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Елабужский институт (филиал) федерального государственного автономного
образовательного учреждения высшего образования
«Казанский (Приволжский) федеральный университет»

Тестовые задания
по МДК.01.02 Поддержка и тестирование программных модулей
(ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ПК 1.3, ПК 1.4, ПК 1.5)

Вариант 1

1. Тестовое окружение может использоваться для:

1. запуска и выполнения тестируемого модуля,
2. передачи входных данных,
3. сбора ожидаемых выходных данных,
4. сравнения реальных выходных данных с ожидаемыми,
5. поддержки отчуждения отдельных модулей системы от всей системы

Ответ: 1, 2, 4, 5 (ОК 01-09, ПК 1.3-1.5)

2. Модульное тестирование проводится для того, чтобы:

1. удостовериться в корректной работе системы в целом,
2. удостовериться в корректной работе набора модулей,
3. удостовериться в корректной работе отдельного модуля

Ответ: 3 (ОК 01-09, ПК 1.3-1.5)

3. Какие основные задачи решаются в ходе модульного тестирования?

1. Поиск и документирование несоответствий требованиям,
2. Поддержка разработки и рефакторинга низкоуровневой архитектуры системы и межмодульного взаимодействия,
3. Рефакторинг модулей,
4. Поддержка рефакторинга модулей,
5. Отладка,
6. Поддержка устранения дефектов и отладки

Ответ: 1, 2, 4, 6 (ОК 01-09, ПК 1.3-1.5)

4. Выберите верные утверждения:

1. Полное покрытие по веткам дает полное покрытие по строкам,
2. Полное покрытие по веткам не дает полного покрытия по строкам,
3. Полное покрытие по строкам без ветвления дает полное покрытие кода по веткам,
4. Полное покрытие по MC\DC не дает полного покрытия по строкам.

Ответ: 1, 3 (ОК 01-09, ПК 1.3-1.5)

5. Согласно методу MC\DC для тестирования логической функции с тремя входами и одним выходом достаточно:

1. 3-х тестовых примеров,
2. 4-х тестовых примеров,
3. 5-х тестовых примеров,
4. 6-х тестовых примеров

Ответ: 2 (ОК 01-09, ПК 1.3-1.5)

6. При использовании какого метода интеграционного тестирования сначала все программные модули, входящие в состав системы, тестируются и только затем объединяются для интеграционного тестирования?

1. Восходящего,
2. монолитного,
3. нисходящего,

4. с поздней интеграцией,
5. с постоянной интеграцией,
6. с регулярной интеграцией

Ответ: 1 (ОК 01-09, ПК 1.3-1.5)

7. Для каких видов интеграционного тестирования нужен драйвер?

1. Восходящего,
2. Монолитного,
3. Нисходящего,
4. С поздней интеграцией,
5. С постоянной интеграцией,
6. С регулярной интеграцией

Ответ: 1, 5, 6 (ОК 01-09, ПК 1.3-1.5)

8. Для каких видов интеграционного тестирования при разработке часто выполняется интеграция?

1. Восходящего,
2. Монолитного,
3. нисходящего,
4. с поздней интеграцией,
5. с постоянной интеграцией,
6. с регулярной интеграцией

Ответ: 3, 5, 6 (ОК 01-09, ПК 1.3-1.5)

9. Тестовое окружение для программного кода на структурных языках программирования состоит из:

1. Драйвера,
2. Тестов,
3. заглушек,
4. исходного кода

Ответ: 1, 3 (ОК 01-09, ПК 1.3-1.5)

10. Модуль — это (с точки зрения наших семинарских занятий):

1. часть программного кода, выполняющая одну функцию с точки зрения функциональных требований,
2. программный модуль, т.е. минимальный компилируемый элемент программной системы
3. задача в списке задач проекта,
4. участок кода, который может уместиться на одном экране или одном листе бумаги,
5. один класс или их множество с единым интерфейсом.

Ответ: 2 (ОК 01-09, ПК 1.3-1.5)

11. Delphi – это...

1. Язык программирования низкого уровня
2. Система объектно-ориентированного визуального программирования
3. Программа обработки видео
4. Программа для перевода кода программы на язык машинных кодов.

Ответ: 2 (ОК 01-09, ПК 1.3-1.5)

12. Язык программирования Delphi

1. Object Pascal
2. Java
3. C#

Ответ: 1 (ОК 01-09, ПК 1.3-1.5)

13. IDE не включает в себя: (несколько вариантов ответа)

1. Редактор кода

2. Высокопроизводительный компилятор в машинный код
3. Объектно-ориентированную модель компонент
4. Эффекты и переходы
5. Визуальное построение приложений
6. Сопровождение ПП
7. Средство для построения баз данных

Ответ: 4,6 (ОК 01-09, ПК 1.3-1.5)

14. Дополнить предложение. Основным окном разрабатываемого приложения является ...

1. Код
2. Форма
3. Библиотека
4. Объект

Ответ: 2 (ОК 01-09, ПК 1.3-1.5)

15. Названия процедур на определенное событие:

1. Компоненты
2. Обработчики событий
3. Редактор кода

Ответ: 2 (ОК 01-09, ПК 1.3-1.5)

16. Назначение панели инструментов:

1. Обработка событий
2. Меню команд быстрого доступа к командам, содержащее набор кнопок, функции которых эквивалентны наиболее часто употребляемым командам Главного меню
3. Построение баз данных

Ответ: 2 (ОК 01-09, ПК 1.3-1.5)

17. Дополнить предложение. Главное окно управляет окнами ... (несколько вариантов ответа)

1. Инспектор объектов
2. Язык ассемблера
3. Редактора кода
4. Проектировщика форм
5. Трансляции программы

Ответ: 1,3,4 (ОК 01-09, ПК 1.3-1.5)

18. Действия которые нельзя выполнить с помощью проектировщика форм

1. Добавить компоненты в форму
2. Сменить язык программирования
3. Модифицировать форму и её компоненты
4. Связать обработчики событий компонента с программой на Object Pascal, содержащейся в редакторе кода
5. Документирование программы

Ответ: 2,5 (ОК 01-09, ПК 1.3-1.5)

19. Дополнить предложение. Инспектор объектов состоит из ... (несколько вариантов ответа)

1. Unit – программного модуля
2. Properties – списка свойств,
3. Events – списка событий.
4. Begin...end.

Ответ: 2,3 (ОК 01-09, ПК 1.3-1.5)

20. Задание метки текста из редактора ввода:

1. Memo1.Caption := Edit1.Text;
2. Label1.Caption := Edit1.Text;
3. Form1.Caption := Edit1.Text;

Ответ: 2 (ОК 01-09, ПК 1.3-1.5)

21. Передача фокуса ввода на редактор ввода

1. Label1.SetFocus;
2. Edit1.Add;
3. Edit1.SetFocus;

Ответ: 3 (ОК 01-09, ПК 1.3-1.5)

22. Для обозначения комментария не используются:

1. (*комментарий*)
2. \комментарий/
3. {комментарий}
4. //комментарий

Ответ: 2 (ОК 01-09, ПК 1.3-1.5)

23. Дополнить предложение. Объект Application определяет ...

1. Свойства и методы программы-приложения для Windows
2. Отображение размещения формы
3. Перемещение по программному коду

Ответ: 1 (ОК 01-09, ПК 1.3-1.5)

24. Контейнер, содержащий все элементы программы:

1. Unit1
2. Form1
3. Edit1
4. Memo1

Ответ: 2 (ОК 01-09, ПК 1.3-1.5)

25. poDesigned:

1. Форма выводится в центр экрана
2. Windows автоматически определяет начальную позицию и размеры формы
3. Форма отображается в той позиции и с теми размерами, которые были установлены при её конструировании

Ответ: 3 (ОК 01-09, ПК 1.3-1.5)

Вариант 2

1. Тестовое окружение для программного кода на структурных языках программирования состоит из:

1. драйвера,
2. тестов,
3. заглушек,
4. исходного кода

Ответ: 1, 3 (ОК 01-09, ПК 1.3-1.5)

2. Модуль — это (с точки зрения наших семинарских занятий):

1. часть программного кода, выполняющая одну функцию с точки зрения функциональных требований,
2. программный модуль, т.е. минимальный компилируемый элемент программной системы,
3. 3. задача в списке задач проекта,
4. участок кода, который может уместиться на одном экране или одном листе бумаги,
5. один класс или их множество с единым интерфейсом.

Ответ: 2 (ОК 01-09, ПК 1.3-1.5)

3. Полная система тестов позволяет утверждать, что:

1. система реализует всю функциональность, указанную в требованиях,
2. система работает корректно,
3. система не реализует функциональность, которая не указана в требованиях,
4. система работает правильно,

5. система реализует функциональность, которая не указана в требованиях,
6. система не реализует функциональность, которая указана в требованиях

Ответ: 1, 3 (ОК 01-09, ПК 1.3-1.5)

4. Какие условия должны быть выполнены для обеспечения полного покрытия по методу MC\DC?

1. должно быть показано зависимое влияние каждой из компонент на значение логического условия,
2. каждое логическое условие должно принимать все возможные значения ,
3. каждая компонента логического условия должна хотя бы один раз принимать все возможные значения,
4. любая часть логического условия должна принимать хотя бы раз все возможные значения,
5. должно быть показано независимое влияние каждой из компонент на значение логического условия

Ответ: 2, 3, 5 (ОК 01-09, ПК 1.3-1.5)

5. Одной из основных задач анализа полноты покрытия кода является:

1. выявление участков кода, которые выполняются при выполнении тестовых примеров,
2. выявление участков кода, которые содержат ошибки,
3. выявление участков кода, которые не выполняются при выполнении тестовых примеров,
4. выявление участков кода, которые не содержат ошибок

Ответ: 3 (ОК 01-09, ПК 1.3-1.5)

6. При использовании какого метода интеграционного тестирования подразумевается, что, как только разрабатывается новый модуль системы, он сразу же интегрируется со всей остальной системой?

1. Восходящего,
2. Монолитного,
3. Нисходящего,
4. с поздней интеграцией,
5. С постоянной интеграцией,
6. С регулярной интеграцией

Ответ: 5 (ОК 01-09, ПК 1.3-1.5)

7. Для каких видов интеграционного тестирования нужны заглушки?

1. Восходящего,
1. Монолитного,
2. нисходящего,
3. с поздней интеграцией,
4. с постоянной интеграцией,
5. с регулярной интеграцией

Ответ: 1, 3, 6 (ОК 01-09, ПК 1.3-1.5)

8. Тестовое окружение может использоваться для:

1. запуска и выполнения тестируемого модуля,
2. передачи входных данных,
3. сбора ожидаемых выходных данных,
4. сравнения реальных выходных данных с ожидаемыми,
5. поддержки отчуждения отдельных модулей системы от всей системы

Ответ: 1, 2, 4 (ОК 01-09, ПК 1.3-1.5)

9. Модульное тестирование проводится для того, чтобы:

1. удостовериться в корректной работе системы в целом,
2. удостовериться в корректной работе набора модулей,

- 3.** удостовериться в корректной работе отдельного модуля
 Ответ: 3 (ОК 01-09, ПК 1.3-1.5)
10. Какие основные задачи решаются в ходе модульного тестирования?
1. Поиск и документирование несоответствий требованиям,
 2. Поддержка разработки и рефакторинга низкоуровневой архитектуры системы и меж-модульного взаимодействия,
 3. 3. Рефакторинг модулей,
 4. Поддержка рефакторинга модулей,
 5. Отладка,
 6. Поддержка устранения дефектов и отладки
- Ответ: 1, 2, 4, 6 (ОК 01-09, ПК 1.3-1.5)
11. RAD – это...
1. Программа записи видео
 2. Средство модульного программирования
 3. Среда быстрой разработки приложений
- Ответ: 3 (ОК 01-09, ПК 1.3-1.5)
12. IDE – это...
1. История развития Delphi
 2. Приложение, разрабатываемое программистом
 3. Интегрированная среда разработки
- Ответ: 3 (ОК 01-09, ПК 1.3-1.5)
13. Объектно-ориентированная программа - ...
1. Программирование, основанное на объектах
 2. Совокупность объектов и способов их взаимодействия
 3. Структура среды программирования
- Ответ: 2 (ОК 01-09, ПК 1.3-1.5)
14. Дополнить предложение. Коды для стандартных окон диалога и кнопок в системе Delphi получены от ...
1. Компилятора
 2. C++
 3. Windows
- Ответ: 3 (ОК 01-09, ПК 1.3-1.5)
15. Назначение главного меню:
1. Осуществление функций управления при разработке программ
 2. Сопровождение программных продуктов
 3. Автоматизирует процесс тестирования программ
- Ответ: 1 (ОК 01-09, ПК 1.3-1.5)
16. Назначение палитры компонентов:
1. содержит пиктограммы, которые представляют компоненты VCL , которые можно включить в приложение
 2. Подделка подписей
 3. Управление Paint
- Ответ: 1 (ОК 01-09, ПК 1.3-1.5)
17. Заготовка разрабатываемого приложения - ...
1. Компилятор
 2. Свойство
 3. Окно проектировщика форм
 4. Главное окно
- Ответ: 3 (ОК 01-09, ПК 1.3-1.5)
18. Дополнить предложение. Инспектор объектов позволяет ... (несколько вариантов ответа)

1. Устанавливать свойства объектов
2. Изменять структуру программного кода
3. Изменять наследование классов объектов
4. Назначать методы обработки событий

Ответ: 1,4 (ОК 01-09, ПК 1.3-1.5)

19. Окно содержащее текст программы на языке Object Pascal, связанный с каждой формой в приложении:

1. Окно компилятора
2. Окно редактора кода
3. Окно проектировщика
4. Главное окно

Ответ: 2 (ОК 01-09, ПК 1.3-1.5)

20. Обнуление строки ввода:

1. Edit1.Text :='';
2. Edit1.Text := TМемо;
3. Edit1.Text :='Закругляемся'

Ответ: 1 (ОК 01-09, ПК 1.3-1.5)

21. Открыть файл проекта Project1 можно, нажав:

1. CTRL+F4
2. CTRL+ALT+F10
3. CTRL+F12
4. ALT+F12

Ответ: 3 (ОК 01-09, ПК 1.3-1.5)

22. Резервированное слово, объявляющее блок подключаемых к проекту модулей:

1. Program
2. Begin..end
3. Uses
4. Forms

Ответ: 3 (ОК 01-09, ПК 1.3-1.5)

23. Метод Application.Run:

1. Создает окно формы, регистрирует входящие в него компоненты
2. Подключает автоматический цикл обработки сообщений Windows к приложению
3. Выполняет подпрограммы раздела инициализации модулей приложения в случае их наличия

Ответ: 2 (ОК 01-09, ПК 1.3-1.5)

24. Дополнить предложение. Свойство Position определяет ...

1. Размещение и размер формы
2. Подключение модулей
3. Цвет шрифта

Ответ: 1 (ОК 01-09, ПК 1.3-1.5)

25. Определяет способ выравнивания компонента внутри контейнера:

1. Свойство Position
2. Свойство Align
3. Свойство Memo
4. Свойство ScrollBars

Ответ: 2 (ОК 01-09, ПК 1.3-1.5)

Критерии оценки при проведении тестирования

«отлично»	«хорошо»	«удовлетворительно»	«неудовлетворительно»
студент выполнил 91-100% и набрал 23-25 баллов.	студент выполнил 76-90% и набрал 19-22 баллов.	студент выполнил 60-75% и набрал 15-18 баллов.	студент выполнил менее 0-59 % и набрал 0-14 баллов.

Вопросы к экзамену
по МДК.01.02 Поддержка и тестирование программных модулей
(ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ПК 1.3, ПК 1.4, ПК 1.5)

1. Валидация(аттестация) и верификация (ОК 01-09, ПК 1.3-1.5)

Ответ: Термин "верификация" (verification) в русскоязычной литературе обычно переводят, как "проверка". Термин "валидация" - как "проверка правильности", "аттестация", "утверждение". Согласно стандарту IEEE 1012-1986, верификация представляет собой процесс оценивания системы или компонента с целью определить, удовлетворяют ли результаты некой фазы условиям, наложенным в начале данной фазы. Валидация в этом же стандарте определяется, как процесс оценивания системы или компонента во время или по окончании процесса разработки с целью определить, удовлетворяет ли она указанным требованиям.

2. Общие вопросы верификации ПО. (ОК 01-09, ПК 1.3-1.5)

Ответ: Методики повышения качества ПО. Контроль качества – планомерная и систематичная программа действий, призванная гарантировать, что система обладает желательными характеристиками. Явное определение целевых характеристик (внутренних и внешних) – эффективная методика. Разработка стратегии тестирования. Выполнить задачи оценки и повышения качества только путем тестирования невозможно. Неформальные и формальные технические обзоры: инспекция, обзор, аудит, контроль изменений, оценка результатов выполнения плана контроля качества, прототипирование.

3. Цели и задачи верификации ПО (ОК 01-09, ПК 1.3-1.5)

Ответ: Цели тестирования: продемонстрировать разработчикам и заказчикам, что программа соответствует требованиям; выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации.

4. Статические и динамические методы верификации ПО (ОК 01-09, ПК 1.3-1.5)

Ответ: Статический анализ программы - это анализ который выполняется без фактического выполнения программы (анализ, проведенный при выполнении ПО известен как динамический анализ). В большинстве случаев анализируется некоторая версия исходного кода. В отличие от динамического анализа, статический анализ позволяет проанализировать все возможные пути выполнения программы. Этот термин обычно используется в случае, когда анализ производится с помощью автоматизированных инструментов.

Динамические методы верификации ПО - это методы, в рамках которых анализ программного обеспечения осуществляются при помощи реального выполнения программы. В случае имитационного моделирования выполняется не сама программа, а программа ее моделирующая. На вход программы поступают последовательности данных, которые могут вызвать недетерминированное поведение, тем самым позволяя обнаружить уязвимости и ошибки. Динамический анализ можно разделить на несколько видов: тестирование, мониторинг, имитационное тестирование, профилирование. Факторы и атрибуты внешнего и внутреннего качества ПО.

5. Виды и методы тестирования. Смоук-тестирование, регрессионное тестирование, тестирование белого и черного ящиков(ОК 01-09, ПК 1.3-1.5)

Ответ: Дымовой тест (англ. Smoke testing или smoke test, дымовое тестирование) — в тестировании программного обеспечения означает минимальный набор тестов на явные ошибки. Дымовой тест обычно выполняется программистом; не прошедшую этот тест программу не имеет смысла отдавать на более глубокое тестирование.

Регрессионное тестирование – это повторное выполнение тестов для проверки того, что изменения, внесенные в программу в результате разработки новой или изменения существующей функциональности, устранения ошибок не повлияли на функциональность, которая не изменилась.

Метод белого ящика. Для тестирования программного кода без его непосредственного запуска применяется метод белого ящика. Тестируемый имеет доступ к исходному коду программного средства и

может писать код, который связан с библиотеками тестируемого программного средства. Это типично для компонентного тестирования, при котором тестируются только отдельные части системы. Такие тесты основаны на знании кода приложения и его внутренних механизмов. Метод белого ящика часто используется на стадии, когда приложение еще не собрано воедино, но необходимо проверить каждый из его компонентов, модули процедур и подпрограмм. Компонентным тестированием чаще всего занимается программист, хорошо понимающий код или тестировщик, имеющий прекрасные знания в области программирования.

Метод черного ящика. При использовании метода черного ящика тестировщик имеет доступ к программному обеспечению только через те же интерфейсы, что и заказчик или пользователь. Либо через внешние интерфейсы, позволяющие другому компьютеру или процессу подключиться к системе для тестирования. Например, тестирующий модуль может виртуально нажимать клавиши или кнопки мыши в тестируемой программе с помощью механизмов взаимодействия процессов. Эти события вызывают тот же отклик, что и реальные нажатия клавиш и кнопок мыши. Как правило тестирование черного ящика ведется с использованием спецификации или иных документов, описывающих требования к системе. Тестировщик тестирует программу так, как с ней будет работать конечный пользователь и он ничего не знает о внутренних алгоритмах и механизмах по которым работает программа. Как при этом обрабатываются входные данные он не знает. Цель данного метода – проверить работу всех функций приложения на соответствие функциональным требованиям.

6. Тестирование нефункциональных требований (ОК 01-09, ПК 1.3-1.5)

Ответ: Нефункциональное тестирование проводится для проверки нефункциональных требований приложения, таких как производительность, безопасность, совместимость, надежность, удобство использования и т. д. В большинстве случаев это выполняется методом black box testing. Оно проверяет, соответствует ли поведение системы требованиям по всем аспектам, не охваченным функциональным тестированием. В нашем повседневном тестировании много внимания уделяется функциональному тестированию и функциональным требованиям и клиенты также заинтересованы в выполнении функциональных требований, которые напрямую связаны с функциональностью приложения, но когда ПО выходит на рынок и используется реальными конечными пользователями, у них есть шансы столкнуться с проблемами. Эти проблемы не связаны с функциональностью системы, но могут негативно повлиять на пользовательский опыт.

7. Взаимосвязь разработки и тестирования. V-модель разработки ПО.(ОК 01-09, ПК 1.3-1.5)

Ответ: V-модель — это тип модели SDLC, в которой процесс выполняется последовательно в V-образной форме. Модель основана на объединении фазы тестирования с каждой соответствующей стадией разработки. Разработка каждого шага напрямую связана с этапом тестирования. Следующая фаза начинается только после завершения предыдущей. Каждый этап разработки, напрямую связан с тестированием этого этапа.

Этапы тестирования V-модели:

Модульное тестирование — модульного тестирования разрабатываются на этапе проектирования модуля. Эти планы модульного тестирования выполняются для устранения ошибок на уровне кода или модуля.

Интеграционное тестирование — после завершения модульного тестирования выполняется интеграционное тестирование. При интеграционном тестировании модули интегрируются, и система тестируется. Интеграционное тестирование выполняется на этапе проектирования архитектуры. Этот тест проверяет связь модулей между собой.

Системное тестирование — тестирование тестирует все приложение с его функциональностью, взаимозависимостью и связью. Оно проверяет функциональные и нефункциональные требования разработанного приложения.

Пользовательское приемочное тестирование (UAT): UAT выполняется в пользовательской среде, напоминающей производственную среду. UAT проверяет, что поставленная система соответствует требованиям пользователя и готова к использованию в реальном мире.

8. Уровни тестирования. Модульное (unit), интеграционное (integration), системное (system), приемочное (acceptance) тестирование. (ОК 01-09, ПК 1.3-1.5)

Ответ: Еще одна классификация видов тестирования основана на том уровне детализации работ проекта, на который оно нацелено. Эти же разновидности тестирования можно связать с фазой жизненного цикла, на которой они выполняются.

Модульное тестирование(Unit-testing) — уровень тестирования, на котором тестируется минимально возможный для тестирования компонент, например, отдельный класс или функция. На этом уровне применяются методы «белого ящика». В современных проектах модульное тестирование («юнит-тестинг») осуществляется разработчиками.

Интеграционное тестирование(Integration testing) – уровень тестирования, на котором отдельные программные модули объединяются и тестируются в группе. Обычно интеграционное тестирование проводится после модульного тестирования (юнит-тесты для модулей должны быть выполнены и найденные ошибки исправлены).

Системное тестирование(System testing)- это тестирование программного обеспечения, выполняемое на полной, интегрированной системе, с целью проверки соответствия системы исходным требованиям. Системное тестирование относится к методам тестирования «чёрного ящика», и, тем самым, не требует знаний о внутреннем устройстве системы.

Приемочное тестирование(Acceptance testing)- это тестирование готового продукта конечными пользователями на реальном окружении, в котором будет функционировать тестируемое приложение. Приемочные тесты разрабатываются пользователями, обычно, в виде сценариев. Для того, чтобы найти больше ошибок рекомендуется планировать не только системное тестирование и приемочное, но и модульное и интеграционное.

9. Техники тест дизайна. Разбиение на классы эквивалентности и тестирование граничных значений.(ОК 01-09, ПК 1.3-1.5)

Ответ: Тест-дизайна - это стратегии, которые помогают лучше писать тест-кейсы. Их использование позволяет создавать меньше тестов, обеспечивая при этом широкий охват требований. Классы эквивалентности – это разбиение множества вводимых данных на отдельные подмножества, с которыми функционал ПО должен работать одинаково. Граничные условия – это то значение, по которому идет разбиение на классы эквивалентности.

10. Понятие дефекта. Основные определения и классификация дефектов.(ОК 01-09, ПК 1.3-1.5)

Ответ: Дефекты – это ошибки, проблемы или несоответствия, которые возникают в процессе разработки и эксплуатации программного обеспечения. Они могут привести к неправильной работе системы, нарушению ее функциональности или даже критическим сбоям.

Дефекты можно классифицировать по разным признакам, например, по их происхождению: архитектурные, функциональные, процессные и т. д. Дефекты архитектуры возникают из-за некорректного проектирования программной системы, которое приводит к непредсказуемым результатам при ее использовании. Дефекты функциональности относятся к ошибкам в реализации конкретной функциональности системы. Дефекты процесса связаны с ошибками на этапе разработки и тестирования, такими как неверная конфигурация системы, неправильное тестирование и т. п.

11. Описание дефектов (ОК 01-09, ПК 1.3-1.5)

Ответ: атрибуты отчета о дефекте: идентификатор, дата, краткое описание (что, где и при каких условиях), подробное описание (ожидаемый результат, фактический результат, ссылка на требование), шаги воспроизведения (1. Перейти по ссылке www.omg.org 2. Ввести в поле «логин» значение «admin» 3. Ввести в поле «пароль» значение «abcd» 4. Кликнуть левой кнопкой мыши по кнопке войти. Баг: В левом верхнем углу вместо логотипа пустое место.), воспроизводимость (указывается частота воспроизведения дефекта (всегда или иногда)) (пройти шаги воспроизведения минимум три раза), важность (насколько серьезна ошибка), срочность (как быстро необходимо исправить ошибку), симптом.

12. Атрибуты дефектов. Приоритет(priority) и серьезность(severity) дефектов (ОК 01-09, ПК 1.3-1.5)

Ответ: Серьезность (Severity) — это атрибут, характеризующий влияние дефекта на работоспособность приложения. Проставляется специалистом по тестированию.

Приоритет (Priority) – это атрибут, указывающий на очередность выполнения задачи или устранения дефекта. Проставляется руководителем или менеджером проекта.

P1 – Высокий (High) – требуется исправить в первую очередь.

P2 – Средний (Medium) – требуется исправить во вторую очередь, когда нет дефектов с высоким приоритетом.

P3 – Низкий (Low) – исправляется в последнюю очередь, когда все дефекты с более высоким приоритетом уже исправлены.

13. Определение серьезности дефекта по его описанию (ОК 01-09, ПК 1.3-1.5)

Ответ: Серьезность имеет несколько параметров в зависимости от типа дефекта. Ее степень зависит от того, как она влияет на бизнес-логику (реализацию правил программы).

S1 – Блокирующая (Blocker). Блокирующая ошибка, приводящая приложение в нерабочее состояние, в результате которого дальнейшая работа с тестируемой системой или ее функциями становится невозможна.

S2 – Критическая (Critical). Критическая ошибка, неправильно работающая бизнес-логика, проблема, приводящая в нерабочее состояние некоторую часть системы, но есть возможность для работы с тестируемой функцией, используя другие входные точки.

S3 – Значительная (Major). Значительная ошибка, часть бизнес-логики работает некорректно. Ошибка не критична или есть возможность для работы с тестируемой функцией, используя другие входные точки.

S4 – Незначительная (Minor). Незначительная ошибка, не нарушающая бизнес-логику тестируемой части приложения, очевидная проблема пользовательского интерфейса.

S5 – Тривиальная (Trivial). Тривиальная ошибка, не касающаяся бизнес-логики приложения, плохо воспроизводимая проблема, малозаметная по средствам пользовательского интерфейса, проблема сторонних библиотек или сервисов, проблема, не оказывающая никакого влияния на общее качество продукта.

14. Версионирование ПО на разных стадиях разработки. (ОК 01-09, ПК 1.3-1.5)

Ответ: Версионирование - это способ группировки и маркировки изменений, вызванных эволюцией системы.

Применение версионирования решает следующие основные задачи разработки и сопровождения ПО: Контроль изменений - с помощью версий, пользователи данной системы могут выбрать наиболее полезное для себя состояние (к примеру, с некоторой функциональностью или совместимую с данной ОС)

Группировка - версии позволяют автору ПО сгруппировать изменения и предоставить их пользователям в виде патча или обновления

Совместимость - при использовании “Семантического версионирования” (о котором мы поговорим немного позже), разработчики, использующие данный пакет, могут не бояться сломать свою систему после обновления или патча, так как правильный выбор версии гарантирует совместимость на уровне кода.

15. Инструментальные средства поддержки тестирования. (ОК 01-09, ПК 1.3-1.5)

Ответ: 1. Организатор тестов. Управляет выполнением тестов. Он отслеживает тестовые данные, ожидаемые результаты и тестируемые функции программы.

2. Генератор тестовых данных. Генерирует тестовые данные для тестируемой программы. Он может выбирать тестовые данные из базы данных или использовать специальные шаблоны для генерации случайных данных необходимого вида.

3. Оракул. Генерирует ожидаемые результаты тестов. В качестве оракулов могут выступать предыдущие версии программы или исследуемого объекта. При тестировании параллельно запускаются оракул и тестируемая программа и сравниваются результаты их выполнения.

4. Компаратор файлов. Сравнивает результаты тестирования с результатами предыдущего тестирования и составляет отчет об обнаруженных различиях. Компараторы особенно важны при сравнении различных версий программы. Различия в результатах указывают на возможные проблемы, существующие в новой версии системы.

5. Генератор отчетов. Формирует отчеты по результатам проведения тестов.
6. Динамический анализатор. Добавляет в программу код, который подсчитывает, сколько раз выполняется каждый оператор. После запуска теста создает исполняемый профиль, в котором показано, сколько раз в программе выполняется каждый оператор.
7. Имитатор. Существует несколько типов имитаторов. Целевые имитаторы моделируют машину, на которой будет выполняться программа. Имитатор пользовательского интерфейса – это программа, управляемая сценариями, которая моделирует взаимодействия с интерфейсом пользователя. Имитатор ввода-вывода генерирует последовательности повторяющихся транзакций.

16. Системы отслеживания ошибок (Bug Tracking Systems) (ОК 01-09, ПК 1.3-1.5)

Ответ: Система отслеживания ошибок (англ. bug tracking system) — прикладная программа, разработанная с целью помочь разработчикам программного обеспечения (программистам, тестировщикам и др.) учитывать и контролировать ошибки и неполадки, найденные в программах, пожелания пользователей, а также следить за процессом устранения этих ошибок и выполнения или невыполнения пожеланий.

Главный компонент такой системы — база данных, содержащая сведения об обнаруженных дефектах. Эти сведения могут включать в себя:

- номер (идентификатор) дефекта;
- короткое описание дефекта;
- кто сообщил о дефекте;
- дата и время, когда был обнаружен дефект;
- версия продукта, в которой обнаружен дефект;
- серьёзность (критичность) дефекта и приоритет решения;
- описание шагов для выявления дефекта (воспроизведения неправильного поведения программы);
- ожидаемый результат и фактический результат;
- кто ответственен за устранение дефекта;
- обсуждение возможных решений и их последствий;
- текущее состояние (статус) дефекта;
- версия продукта, в которой дефект исправлен.

17. Артефакты разработки ПО, относящиеся к тестированию. Тест-кейсы (test cases) (ОК 01-09, ПК 1.3-1.5)

Ответ: Test case (тест-кейс, тестовый пример/случай) - это артефакт, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или ее части. Более строго - формализованное описание одной показательной проверки на соответствие требованиям прямым или косвенным.

18. Артефакты разработки ПО, относящиеся к тестированию. План тестирования (test plan). (ОК 01-09, ПК 1.3-1.5)

Ответ: Тест план (Test Plan) - это артефакт, описывающий весь объем работ по тестированию, начиная с описания объекта, стратегии, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения.

ПРАКТИЧЕСКИЕ ЗАДАНИЯ К ЭКЗАМЕНУ

1. Требуется разработать приложение с графическим пользовательским интерфейсом, поддерживающее создание/редактирование/удаление/поиск заметок. Два варианта хранения заметок: А) в базе SQLite. Б) С использованием файловой системы. (ОК 01-09, ПК 1.3-1.5)

2. Создать приложение с графическим пользовательским интерфейсом с функциями: Определение местоположения пользователя на карте Google Map; Определение скорости и направления движения пользователя; Масштабирование карты. Программа должна быть конфигурируемой.

Настройки: Режим определения местоположения (через GPS либо по сотам); Включение/отключение режима поиска. (ОК 01-09, ПК 1.3-1.5)

3. Разработать приложение-таймер с использованием датчика ориентации в виде песочных часов. Каждый раз для того чтобы активировать таймер, необходимо перевернуть экран мобильного устройства вверх ногами. Используйте анимацию для показа «перетекающего песка» и переворота песочных часов. Для задания времени перетекания песка требуется разработать push-notification сервер. Через форму ввода на сервере можно отправлять на клиент (приложение-таймер) указанное время (числовой ввод). (ОК 01-09, ПК 1.3-1.5)

4. Разработка программы для обмена мгновенными сообщениями. Требуется разработать приложение для обмена мгновенными сообщениями через Wi-Fi/Bluetooth. Поддерживаемые режимы: 1. Активный режим. Приложение занимает весь экран, содержит поля для отправки сообщений и список принятых сообщений. 2. Режим уведомлений. Приложение через уведомления показывает принятые сообщения. (ОК 01-09, ПК 1.3-1.5)

5. Разработка мобильного сайта с адаптивным дизайном. Требуется разработать сайт, пригодный для просмотра на смартфонах и планшетах, с использованием принципов адаптивного дизайна: 1) Гибкая сетка. 2) Пропорциональные шрифты. 3) Масштабируемые изображения. 4) Медиа-запросы (ОК 01-09, ПК 1.3-1.5)

Критерии оценки при проведении экзамена

«отлично»	«хорошо»	«удовлетворительно»	«неудовлетворительно»
студент раскрывает теоретический вопрос билета, практическое задание выполняет без ошибок, уверенно отвечает на дополнительные вопросы	студент раскрывает теоретический вопрос, практическое задание выполняет без ошибок, на дополнительные вопросы отвечает неуверенно, допускает не точности в определениях.	студент раскрывает теоретический вопрос не в полной мере, допускает неточности в формулировках (1-2 ошибки), практическое задание выполнено частично, с допущением ошибок в расчётах	Теоретический вопрос не раскрыт, практическое задание не выполнено.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Елабужский институт (филиал) федерального государственного автономного
образовательного учреждения высшего образования
«Казанский (Приволжский) федеральный университет»

УТВЕРЖДАЮ

Заместитель директора по
образовательной деятельности
Елабужского института ЕИ КФУ
И.П. Михайлова
«01» марта 2024 г.



ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

МДК.01.03 Разработка мобильных приложений
(наименование дисциплины)

09.02.07 Информационные системы и программирование
(код и наименование специальности)

Программист
(квалификация выпускника)

**Паспорт
фонда оценочных средств по
МДК.01.03 Разработка мобильных приложений**
(наименование междисциплинарного курса)

Результаты (освоенные профессиональные компетенции)	Основные показатели оценки результата	Формы и методы кон- троля и оценки
ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;</p> <p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;</p>	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-20), практические задания (1-4)
ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;</p> <p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;</p>	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-20), практические задания (1-4)
ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;</p> <p>Знать: основные этапы разработки программного обеспечения; основ-</p>	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-20), практические задания (1-4)

	ные принципы технологии структурного и объектно-ориентированного программирования;	
ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-20), практические задания (1-4)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-20), практические задания (1-4)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-20), практические задания (1-4)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ОК 07. Содействовать сохранению окружающей среды, ресурсосбере-	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алго-	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-20), практи-

жению, эффективно действовать в чрезвычайных ситуациях	ритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	ческие задания (1-4)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-20), практические задания (1-4)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ОК. 09. Использовать информационные технологии в профессиональной деятельности	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-20), практические задания (1-4)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ОК.10. Пользоваться профессиональной документацией на государственном и иностранном языках	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-20), практические задания (1-4)

		Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием		Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-20), практические задания (1-4)
		Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	
		Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	
ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ		Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-20), практические задания (1-4)
	Уметь: осуществлять разработку кода		

	<p>программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p>	
	<p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	

**Отчет по лабораторным работам
 по МДК.01.03 Разработка мобильных приложений
 (ОК0 1, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.6)**

**Тема 3.1 Основные платформы и языки разработки мобильных приложений
 (ОК 01-10, ПК 1.2, 1.6)** (ОК-10, ПК

1. Установка инструментария и настройка среды для разработки мобильных приложений. (ОК-10, ПК 1.2, ПК 1.6)
2. Установка среды разработки мобильных приложений с применением виртуальной машины. (ОК-10, ПК 1.2, ПК 1.6)

**Тема 3.2 Создание и тестирование модулей для мобильных приложений
 (ОК 01-10, ПК 1.2, 1.6)**

1. Создание эмуляторов и подключение устройств» (ОК-10, ПК 1.2, ПК 1.6)
2. Настройка режима терминала» (ОК-10, ПК 1.2, ПК 1.6)
3. Создание нового проекта» (ОК-10, ПК 1.2, ПК 1.6)
4. Изучение и комментирование кода» (ОК-10, ПК 1.2, ПК 1.6)
5. Лабораторная работа «Изменение элементов дизайна» (ОК-10, ПК 1.2, ПК 1.6)
6. Обработка событий: подсказки» (ОК-10, ПК 1.2, ПК 1.6)
7. Обработка событий: цветовая индикация» (ОК-10, ПК 1.2, ПК 1.6)
8. Подготовка стандартных модулей» (ОК-10, ПК 1.2, ПК 1.6)
9. Обработка событий: переключение между экранами» (ОК-10, ПК 1.2, ПК 1.6)
10. Передача данных между модулями» (ОК-10, ПК 1.2, ПК 1.6)
11. Тестирование и оптимизация мобильного приложения». (ОК-10, ПК 1.2, ПК 1.6)

Критерии оценки:

Индекс и расшифровка компетенции	Результаты обучения	Критерии оценивания результатов обучения (баллы)			
		2	3	4	5
ОК-1	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений

	программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства				
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
OK-2	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений
	Знать: основные	Не знает	Демонстриру-	Знает доста-	Демонстри-

	этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Допускает грубые ошибки	ет частичные знания без грубых ошибок	точно в базовом объеме	рует высокий уровень знаний
ОК-3	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний

	способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов				
ОК-4	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
ОК-5	Уметь: осуществлять разработку	Не умеет Демонстри-	Демонстрирует частичные	Умеет применять зна-	Демонстрирует высо-

	<p>кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p>	<p>рует частичные умения, допуская грубые ошибки</p>	<p>умения без грубых ошибок</p>	<p>ния на практике в базовом объеме</p>	<p>кий уровень умений</p>
	<p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	<p>Не знает Допускает грубые ошибки</p>	<p>Демонстрирует частичные знания без грубых ошибок</p>	<p>Знает достаточно в базовом объеме</p>	<p>Демонстрирует высокий уровень знаний</p>
<p>ОК- 6</p>	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и</p>	<p>Не умеет Демонстрирует частичные умения, допуская грубые ошибки</p>	<p>Демонстрирует частичные умения без грубых ошибок</p>	<p>Умеет применять знания на практике в базовом объеме</p>	<p>Демонстрирует высокий уровень умений</p>

	тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства				
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
ОК- 7	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений

	оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства				
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
ОК- 8	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений
	Знать: основные этапы разработки программного	Не знает Допускает грубые	Демонстрирует частичные знания без	Знает достаточно в базовом объё-	Демонстрирует высокий уровень

	обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	ошибки	грубых ошибок	ме	знаний
ОК- 9	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы ре-	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний

	факторинга; основные принципы отладки и тестирования программных продуктов				
ОК-10	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
ПК- 1.2	Иметь практический опыт: разработка кода программного продук-	Не владеет Демонстрирует низкий уровень вла-	Демонстрирует частичные владения без грубых оши-	Владеет базовыми приемами	Демонстрирует владения на высоком уровне

	<p>та на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений</p>	<p>дения, допуская грубые ошибки</p>	<p>бок</p>		
	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p>	<p>Не умеет Демонстрирует частичные умения, допуская грубые ошибки</p>	<p>Демонстрирует частичные умения без грубых ошибок</p>	<p>Умеет применять знания на практике в базовом объеме</p>	<p>Демонстрирует высокий уровень умений</p>
	<p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии струк-</p>	<p>Не знает Допускает грубые ошибки</p>	<p>Демонстрирует частичные знания без грубых ошибок</p>	<p>Знает достаточно в базовом объеме</p>	<p>Демонстрирует высокий уровень знаний</p>

	турного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов				
ПК 1.6	Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений	Не владеет Демонстрирует низкий уровень владения, допуская грубые ошибки	Демонстрирует частичные владения без грубых ошибок	Владеет базовыми приемами	Демонстрирует владения на высоком уровне
	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на совре-	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений

	<p>менных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p>				
	<p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	<p>Не знает Допускает грубые ошибки</p>	<p>Демонстрирует частичные знания без грубых ошибок</p>	<p>Знает достаточно в базовом объеме</p>	<p>Демонстрирует высокий уровень знаний</p>

**Вопросы к дифференцированному зачету
по МДК.01.03 Разработка мобильных приложений**

(ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.6)

- 1. Программный стек android. виртуальная машина dalvik. (ОК 01-10, ПК 1.2, 1.6)**
Ответ: Система Android — это программный стек для мобильных устройств, который включает операционную систему, программное обеспечение промежуточного слоя (middleware), а также основные пользовательские приложения (e-mail-клиент, календарь, карты, браузер, контакты и др.). Dalvik - снятая с производства технологическая виртуальная машина (VM) в операционной системе Android, которая выполняет приложения, написанные для Android.[1] (Формат байт-кода Dalvik по-прежнему используется в качестве формата распространения, но больше не используется во время выполнения в более новых версиях Android.)
- 2. Архитектура Android- приложений. (ОК 01-10, ПК 1.2, 1.6)**
Ответ: Как понять архитектуру приложений в Android? Я приведу очень простой пример. Представьте себе автомобильный завод с пятью зонами: Первая зона создает шасси. Вторая зона соединяет механические части. Третья зона собирает электронную схему. Четвертая область — покрасочная. И последняя область добавляет эстетические детали. Это означает, что у каждой зоны есть своя ответственность, и они работают в цепочке с первой зоны по пятую для достижения результата.
- 3. Четыре пункта философии разработки приложений под android. (ОК 01-10, ПК 1.2, 1.6)**
Ответ: 1. Масштабируемость и гибкость: Философия разработки приложений под Android подразумевает создание приложений, способных адаптироваться к различным экранам и устройствам, а также легко масштабироваться для добавления новых функций и обновлений. 2. Использование стандартных компонентов и шаблонов: Это включает создание приложений с использованием стандартных компонентов Android. 3. Оптимизация производительности и эффективное управление ресурсами: Разработчики Android-приложений должны уделять особое внимание оптимизации производительности и эффективному управлению ресурсами. 4. Соблюдение рекомендаций платформы и стандартов безопасности: Философия разработки Android-приложений включает соблюдение руководящих принципов и рекомендаций Android-платформы, а также высокие стандарты в области безопасности, чтобы обеспечить защиту пользовательских данных и безопасность приложения в целом.
- 4. Приемы для улучшения производительности и уменьшения потребления памяти для приложений android. (ОК 01-10, ПК 1.2, 1.6)**
Ответ: приемы уменьшения памяти- Используйте оптимизированные структуры данных. Во многих случаях приложения используют слишком много памяти по той простой причине, что для них используются не самые подходящие структуры данных. Предотвращайте перемешивание памяти. Основная проблема на территории Android в данном случае такова, что мы не контролируем, когда будет происходить сборка мусора.
- 5. Основные составляющие манифеста приложения. (ОК 01-10, ПК 1.2, 1.6)**
Ответ: Имя пакета для приложения. Является идентификатором приложения. Описание возможностей компонентов приложения: Activity, Service, Broadcast Receiver. Процессы, в которых будут запускаться компоненты приложения. Раз-

решения, которые необходимо запросить у пользователя, чтобы у приложения был доступ к защищенным частям API, а также чтобы другие приложения могли обращаться к вашему приложению.

6. Жизненный цикл мобильного приложения. (ОК 01-10, ПК 1.2, 1.6)

Ответ: Жизненный цикл мобильного приложения можно разделить на несколько этапов, включая анализ требований, проектирование, разработку, тестирование, развертывание, обслуживание и вывод из эксплуатации. Такой системный подход гарантирует, что потенциальные проблемы будут решены на ранней стадии и на протяжении всего жизненного цикла, что приводит к максимальной вероятности создания успешного приложения.

7. Разработка интерфейсов, не зависящих от разрешения и плотности пикселей.

Ответ: Задавать размеры в пикселях не рекомендуется, поскольку из-за различной плотности пикселей на экранах разных устройств фактический размер макета будет неодинаков. Всегда задавайте размеры в единицах dp или sp. dp – это не зависящий от разрешения пиксель, равный физическому пикселю на экране с плотностью 160 точек/дюйм. sp является аналогичной единицей измерения, но масштабируется на основе выбранного пользователем размера текста, поэтому ее следует применять для указания величины шрифта, но не размера макета. (ОК 01-10, ПК 1.2, 1.6)

8. Intents и Activities. Принципы работы Intent-фильтров. (ОК 01-10, ПК 1.2, 1.6)

Ответ: Для взаимодействия между различными объектами activity ключевым классом является android.content.Intent. Он представляет собой задачу, которую надо выполнить приложению. по сути activity – это стандартные классы java, которые наследуются от класса Activity или его наследников. Поэтому вместо встроенных шаблонов в Android Studio можем добавлять обычные классы, и затем их наследовать от класса Activity. Однако в этом случае нужно будет вручную добавлять в файл манифеста данные об activity. Причем для MainActivity в элементе intent-filter определяется интент-фильтр. В нем элемент action значение "android.intent.action.MAIN" представляет главную точку входа в приложение. То есть MainActivity остается основной и запускается приложением по умолчанию.

9. Адаптеры и привязка данных. (ОК 01-10, ПК 1.2, 1.6)

Ответ: В Android часто используются адаптеры. Если говорить в общих чертах, то адаптеры упрощают связывание данных с элементом управления. Адаптеры используются при работе с виджетами, которые дополняют android.widget.AdapterView: ListView, ExpandableListView, GridView, Spinner, Gallery, а также в активности ListActivity и др. Сам AdapterView дополняет android.widget.ViewGroup. Итак, у нас есть набор объектов и есть компонент View. Назначение адаптера заключается в том, чтобы предоставлять дочерние виды для контейнера. Адаптер берет данные и метаданные определенного контейнера и строит каждый дочерний вид. Например, мы формируем пункты списка (массив строк) и передаём его списку ListView.

10. Работа с интернет-ресурсами. (ОК 01-10, ПК 1.2, 1.6)

Ответ: Глобальная сеть Интернет обладает огромным потенциалом, который специалисты компании Google оценивают в 1 трлн веб-страниц. Структура Интернета такова, что пользователь может воспользоваться веб-ресурсами, адрес которых ему известен, или поисковыми системами для поиска неизвестных ему веб-сайтов. По сути, единственным средством доступа к деловым ресурсам Интернета являются поисковые системы. Задача эффективной работы с ресурсами Интернета во многом определяется эффективностью поиска информации.

11. Диалоговые окна: создание и использование. (ОК 01-10, ПК 1.2, 1.6)

Ответ: Каждое диалоговое окно обычно содержит несколько элементов управления, которые являются дочерними окнами по отношению к диалогу, а диалог, являясь временным окном по отношению к основному окну, позволяет проводить

обмен данными между пользователем и приложением. Существует несколько функций Win32 API, которые помогают в конструировании, отображении и управлении содержимым диалоговых окон. Разработчикам приложений обычно не нужно заботиться о рисовании элементов управления или об обработке событий пользовательского интерфейса, достаточно сосредоточиться на самом обмене данными между элементами управления диалогового окна и приложением.

12. Курсоры, Content Values. Получение данных из SQLite. (ОК 01-10, ПК 1.2, 1.6)

Ответ: SQLite. Это база данных с таблицами и запросами - все как в обычных БД. Стандартный оператор SELECT возвращает набор данных, релевантный сформированным условиям запроса. В интерактивном SQL этот полученный набор данных просто выводится на консоль пользователя, и он может просмотреть полученные результаты. Встроенный оператор SELECT должен создавать структуры данных, которые согласуются с базовыми языками программирования.

Операции в реляционной базе данных выполняются над множеством строк. Набор строк, возвращаемый инструкцией SELECT, содержит все строки, которые удовлетворяют условиям, указанным в предложении WHERE инструкции. Такой полный набор строк, возвращаемых инструкцией, называется результирующим набором.

13. Фоновые службы, toast-уведомления и сигнализация. (ОК 01-10, ПК 1.2, 1.6)

Ответ: всплывающее уведомление (Toast Notification) является сообщением, которое появляется на поверхности окна приложения, заполняя необходимое ему количество пространства, требуемого для сообщения. При этом текущая деятельность приложения остаётся работоспособной для пользователя. В течение нескольких секунд сообщение плавно закрывается. Всплывающее уведомление также может быть создано службой, работающей в фоновом режиме. Как правило, всплывающее уведомление используется для показа коротких текстовых сообщений.

14. Геолокационные и картографические сервисы: конфигурирование и использование. (ОК 01-10, ПК 1.2, 1.6)

Ответ: Сервисы геолокации — приложения, функционирование которых основано на отслеживании местоположения пользователя. Такие данные можно использовать для оптимизации и популяризации бизнеса. С практической точки зрения приложения с геолокацией являются важным звеном успеха в мобильной индустрии. Значительная часть прибыли, извлекаемой из приложений, приходится на сервисы именно такого типа. И их число на рынке продолжает расти. Геолокационные сервисы применяют для создания карт и всевозможных навигационных систем. Таким образом, пользователь получает инструмент для определения своего местоположения и построения маршрутов.

15. Сенсорные датчики. sensor manager. (ОК 01-10, ПК 1.2, 1.6)

Ответ: Для работы с аппаратными датчиками, доступными в устройствах под управлением Android, применяется класс `SensorManager`, ссылку на который можно получить с помощью стандартного метода `getSystemService`. Чтобы начать работать с датчиком, нужно определить его тип. Удобнее всего это сделать с помощью класса `Sensor`, так как в нем уже определены все типы сенсоров в виде констант. При разработке приложения, эксплуатирующего показания сенсоров, вовсе не обязательно бегать по улице или прыгать в воду с высокой скалы, так как эмулятор, входящий в поставку Android SDK, умеет передавать приложению любые отладочные значения.

16. Анимация и спецэффекты. (ОК 01-10, ПК 1.2, 1.6)

Ответ: Анимации и эффекты играют важную роль в создании привлекательных и удовлетворительных пользовательских интерфейсов в Android-приложениях. В этой статье мы рассмотрим, как добавить динамичность и визуальное воздействие

в ваши приложения. От создания простых анимации до сложных эффектов переходов – давайте разберемся, как сделать ваше приложение незабываемым.

17. Акселерометр, датчик ориентации и компас: регулировка и программные функции. (ОК 01-10, ПК 1.2, 1.6)

Ответ: Этот термин изначально происходит от латинского слова «accelero», которое в дословном переводе означает «ускоряю». Прибор, именуемый акселерометр, используется для измерения кажущегося ускорения или разницы между гравитационным и истинным ускорением физического тела или объекта. Другое название прибора – G-сенсор. Реализация функции в смартфоне предполагает определение угла наклона аппарата относительно поверхности земли путем сопоставления трёх координат: X, Y и Z, или ширины, длины и высоты соответственно. Магнитометр действует в компании с акселерометром и GPS, облегчая навигацию. Именно благодаря ему вы не только можете ориентироваться в лесу, но и видите не просто точку на карте, а стрелку, показывающую в каком направлении вы смотрите.

18. Межпроцессное взаимодействие. Язык AIDL. (ОК 01-10, ПК 1.2, 1.6)

Ответ: В основе взаимодействия лежит bound-сервис, предоставляющий сторонним приложениям доступ к определенному набору методов. Другими словами, одно приложение выступает в роли "клиента" а другое в роли "сервера", причем общаются они по строго заданному интерфейсу, который идентичен для обоих приложений.

(ОК 01-10, ПК 1.2,

19. Основные права и полномочия для запуска приложений на устройстве. 1.6)

Ответ: Все разрешения Андроид затрагивают персональные данные пользователя, а потому их нужно выдавать только тем приложениям, которым вы доверяете. Всегда пытайтесь построить взаимосвязь и спросить себя, к примеру, зачем программе доступ к контактам, и для чего ей может понадобиться совершение звонков.

20. Работа с настройками сотовой сети, подключение голосовых услуг, получение и отправка коротких сообщений. (ОК 01-10, ПК 1.2, 1.6)

Тестовые задания по по МДК.01.03 Разработка мобильных приложений

1. Набор средств программирования, который содержит инструменты, необходимые для создания, компиляции и сборки мобильного приложения называется:

- 1) Android SDK
- 2) JDK
- 3) плагин ADT
- 4) Android NDK

Ответ: 1 (ОК 01-10, ПК 1.2, 1.6)

2. С какой целью был создан Open Handset Alliance?

- 1) Писать историю развития ОС Android
- 2) Продавать смартфоны под управлением Android
- 3) разрабатывать открытые стандарты для мобильных устройств
- 4) рекламировать смартфоны под управлением Android

Ответ: 3 (ОК 01-10, ПК 1.2, 1.6)

3. С какой целью инструмент Intel* Graphics Performance Analyzers (Intel* GPA) System Analyzer используется в среде разработки Intel* Beacon Mountain?

- 1) позволить разработчикам оптимизировать загрузку системы при использовании процедур OpenGL
- 2) для ускорения работы эмулятора в среде разработки
- 3) для оптимизированной обработки данных и изображений
- 4) позволить разработчикам эффективно распараллелить C++ мобильные приложения

Ответ: 1 (ОК 01-10, ПК 1.2, 1.6)

4. Библиотеки, реализованные на базе PacketVideo OpenCORE:

- 1) SQLite
- 2) FreeType
- 3) Media Framework
- 4) 3D библиотеки

Ответ: 3 (ОК 01-10, ПК 1.2, 1.6)

5. Какой движок баз данных используется в ОС Android?

- 1) InnoDB
- 2) DBM
- 3) MyISAM
- 4) SQLite

Ответ: 4 (ОК 01-10, ПК 1.2, 1.6)

6. С какой целью инструмент Intel* Integrated Performance Primitives (Intel* IPP) используется в среде разработки Intel* Veason Mountain?

- 1) для оптимизированной обработки данных и изображений
- 2) позволить разработчикам оптимизировать загрузенность системы при использовании процедур OpenGL
- 3) для ускорения работы эмулятора в среде разработки
- 4) позволить разработчикам эффективно распараллелить C++ мобильные приложения

Ответ: 1 (ОК 01-10, ПК 1.2, 1.6)

7. Intel XDK поддерживает разработку под:

- 1) JavaFX Mobile
- 2) Apple iOS, BlackBerry OS
- 3) MtkOS, Symbian OS, Microsoft Windows 8
- 4) Android, Apple iOS, Microsoft Windows 8, Tizen

Ответ: 4 (ОК 01-10, ПК 1.2, 1.6)

8. Каждый приемник широковещательных сообщений является наследником класса ...

- 1) ViewReceiver
- 2) IntentReceiver
- 3) ContentProvider
- 4) BroadcastReceiver

Ответ: 4 (ОК 01-10, ПК 1.2, 1.6)

9. Какой класс является основным строительным блоком для компонентов пользовательского интерфейса (UI), определяет прямоугольную область экрана и отвечает за прорисовку и обработку событий?

- 1) GUI
- 2) View
- 3) UIComponent
- 4) Widget

Ответ: 2 (ОК 01-10, ПК 1.2, 1.6)

10. Какой слушатель используется для отслеживания события касания экрана устройства?

- 1) OnPressListener
- 2) OnTouchListener
- 3) OnClickListener
- 4) OnInputListener

Ответ: 2 (ОК 01-10, ПК 1.2, 1.6)

11. В какой папке необходимо разместить XML файлы, которые определяют все меню приложения?

- 1) res/value
- 2) res/items

3) res/layout

4) res/menu

Ответ: 4 (ОК 01-10, ПК 1.2, 1.6)

12. Фоновые приложения ...

1) после настройки не предполагают взаимодействия с пользователем, большую часть времени находятся и работают в скрытом состоянии

2) выполняют свои функции и когда видимы на экране, и когда скрыты другими приложениями

3) небольшие приложения, отображаемые в виде графического объекта на рабочем столе

4) большую часть времени работают в фоновом режиме, однако допускают взаимодействие с пользователем и после настройки

Ответ: 1 (ОК 01-10, ПК 1.2, 1.6)

13. Полный иерархический список обязательных файлов и папок проекта можно увидеть на вкладке ...

1) Package Explorer

2) Internet Explorer

3) Navigator

4) Project Explorer

Ответ: 1,4 (ОК 01-10, ПК 1.2, 1.6)

14. Какой компонент управляет распределенным множеством данных приложения?

1) сервис (Service)

2) активность (Activity)

3) приемник широковещательных сообщений (Broadcast Receiver)

4) контент-провайдер (Content Provider)

Ответ: 4 (ОК 01-10, ПК 1.2, 1.6)

15. Какой язык разметки используется для описания иерархии компонентов графического пользовательского интерфейса Android-приложения?

1) html

2) xml

3) gml

4) xhtml

Ответ: 2 (ОК 01-10, ПК 1.2, 1.6)

16. Выберите верную последовательность действий, необходимых для создания в приложении контент-провайдера.

1) Создание класса наследника от класса ContentProvider; Определение способа организации данных; Заполнение контент-провайдера данными

2) Проектирование способа хранения данных; Определение способа организации данных;

3) Создание класса наследника от класса ContentProvider; Заполнение контент-провайдера данными; Определение способа работы с данными

4) Проектирование способа хранения данных; Создание класса-наследника от класса ContentProvider; Определение строки авторизации провайдера, URI для его строк и имен столбцов

Ответ: 4 (ОК 01-10, ПК 1.2, 1.6)

17. Выберите верные утверждения относительно объекта-намерения (Intent).

1) представляет собой структуру данных, содержащую описание операции, которая должна быть выполнена, и обычно используется для запуска активности или сервиса

2) используется для передачи сообщений пользователю

3) используется для получения инструкций от пользователя

- 4) используются для передачи сообщений между основными компонентами приложений
- Ответ: 14 (ОК 01-10, ПК 1.2, 1.6)**
18. Расположение элементов мобильного приложения:
- 1) полезно для передачи иерархии
 - 2) влияет на удобство использования
 - 3) полезно для создания пространственных отношений между объектами на экране и объектами реального мира
 - 4) все варианты ответа верны
- Ответ: 4 (ОК 01-10, ПК 1.2, 1.6)**
19. Какие элементы управления применяются для действий по настройке?
- 1) командные элементы управления
 - 2) элементы выбора
 - 3) элементы ввода
 - 4) элементы отображения
- Ответ: 2 (ОК 01-10, ПК 1.2, 1.6)**
20. Примерами комбо-элементов не являются:
- 1) комбо-список
 - 2) все вышеперечисленное
 - 3) комбо-кнопка
 - 4) комбо-поле
- Ответ: 4 (ОК 01-10, ПК 1.2, 1.6)**
21. Дизайн или проектирование интерфейса для графических дизайнеров:
- 1) все варианты ответа верны
 - 2) прозрачность и понятность информации
 - 3) тон, стиль, композиция, которые являются атрибутами бренда
 - 4) передача информации о поведении посредством ожидаемого назначения
- Ответ: 1 (ОК 01-10, ПК 1.2, 1.6)**
22. Более крупные элементы:
- 1) привлекают больше внимания
 - 2) все варианты ответа верны
 - 3) размер не влияет на уровень внимания
 - 4) привлекают меньше внимания
- Ответ: 1 (ОК 01-10, ПК 1.2, 1.6)**
23. К традиционным типографическим инструментам не относят
- 1) масштаб
 - 2) цвет
 - 3) разреженность
 - 4) выравнивание по сетке
- Ответ: 2 (ОК 01-10, ПК 1.2, 1.6)**
24. К элементам ввода относят:
- 1) ограничивающие элементы ввода
 - 2) ползунки
 - 3) счетчики
 - 4) все вышеперечисленное
- Ответ: 4 (ОК 01-10, ПК 1.2, 1.6)**
25. Выделяют следующие категории плотности экрана для Android-устройств:
- 1) HDPI, XHDPI, XXHDPI, и XXXHDPI
 - 2) правильный вариант ответа отсутствует
 - 3) LDPI, MDPI, HDPI, XHDPI, XXHDPI, и XXXHDPI
 - 4) LDPI, MDPI, HDPI
- Ответ: 3 (ОК 01-10, ПК 1.2, 1.6)**

**Практические задания к дифференцированному зачету
по МДК.01.03 Разработка мобильных приложений
(ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.6)**

1. Требуется разработать приложение с графическим пользовательским интерфейсом, поддерживающее создание/редактирование/удаление/поиск заметок «Записная книжка». Используя два варианта хранения заметок: в базе SQLite или с использованием файловой системы. (ОК 01-10, ПК 1.2, 1.6)
2. Создайте приложение «Карманный навигатор» с графическим пользовательским интерфейсом с функциями: определение местоположения пользователя на карте Google Map; определение скорости и направления движения пользователя; масштабирование карты. Программа должна быть конфигурируемой. (ОК 01-10, ПК 1.2, 1.6)
3. Разработайте приложение-таймер «Песочные часы» с использованием датчика ориентации в виде песочных часов. Каждый раз для того чтобы активировать таймер, необходимо перевернуть экран мобильного устройства вверх ногами. Используйте анимацию для показа «перетекающего песка» и переворота песочных часов. Для задания времени перетекания песка требуется разработать push-notification сервер. Через форму ввода на сервере можно отправлять на клиент (приложение-таймер) указанное время (числовой ввод). (ОК 01-10, ПК 1.2, 1.6)
4. Требуется разработать приложение для обмена мгновенными сообщениями через Wi-Fi/ Bluetooth. Поддерживаемые режимы: активный режим (приложение занимает весь экран, содержит поля для отправки сообщений и список принятых сообщений) и режим уведомлений (приложение через уведомления показывает принятые сообщения). (ОК 01-10, ПК 1.2, 1.6)

Критерии оценки при проведении дифференцированного зачета

«отлично»	«хорошо»	«удовлетворительно»	«неудовлетворительно»
студент раскрывает теоретический вопрос билета, практическое задание выполняет без ошибок, уверенно отвечает на дополнительные вопросы	студент раскрывает теоретический вопрос, практическое задание выполняет без ошибок, на дополнительные вопросы отвечает неуверенно, допускает не точности в определениях.	студент раскрывает теоретический вопрос не в полной мере, допускает неточности в формулировках (1-2 ошибки), практическое задание выполнено частично, с допущением ошибок в расчётах	Теоретический вопрос не раскрыт, практическое задание не выполнено.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Елабужский институт (филиал) федерального государственного автономного
образовательного учреждения высшего образования
«Казанский (Приволжский) федеральный университет»

УТВЕРЖДАЮ

Заместитель директора по
образовательной деятельности
Елабужского института ЕИ КФУ

 И. Н. Михайлова
«01» марта 2024 г.

МП

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

МДК.01.04 Системное программирование

(наименование дисциплины)

09.02.07 Информационные системы и программирование

(код и наименование специальности)

Программист

(квалификация выпускника)

г.Елабуга, 2024

**Паспорт
фонда оценочных средств по
МДК.01.04 Системное программирование**
(наименование междисциплинарного курса)

Результаты (освоенные профессиональные компетенции)	Основные показатели оценки результата	Формы и методы контроля и оценки
ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;</p> <p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;</p>	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-25), практические задания (1-12)
ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;</p> <p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;</p>	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-25), практические задания (1-12)
ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;</p> <p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;</p>	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-25), практические задания (1-12)
ОК 04. Работать в коллективе и команде, эффективно	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по	Отчет по лабораторным работам. Дифференцированный за-

взаимодействовать с коллегами, руководством, клиентами	разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	чет (вопросы 1-25), практические задания (1-12)
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-25), практические задания (1-12)
ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-25), практические задания (1-12)
ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-25), практические задания (1-12)
ОК 08. Использовать средства физической культуры	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по	Отчет по лабораторным работам. Дифференцированный за-

для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	чет (вопросы 1-25), практические задания (1-12)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ОК. 09. Использовать информационные технологии в профессиональной деятельности	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-25), практические задания (1-12)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ОК.10. Пользоваться профессиональной документацией на государственном и иностранном языках	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-25), практические задания (1-12)
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования;	
ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием	Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений	Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-25), практические задания (1-12)
	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный	

	<p>модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p> <p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	
<p>ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств</p>	<p>Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений</p> <p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p> <p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	<p>Отчет по лабораторным работам. Дифференцированный зачет (вопросы 1-25), практические задания (1-12)</p>

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
 Елабужский институт (филиал) федерального государственного автономного
 образовательного учреждения высшего образования
 «Казанский (Приволжский) федеральный университет»

**Отчет по лабораторным работам
 по МДК.01.04 Системное программирование
 (ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3)**

**Тема 4.1 Основные платформы и языки разработки мобильных приложений
 (ОК 01-10, ПК 1.2-1.3)**

1. Использование потоков. (ОК 01-10, ПК 1.2-1.3)
2. Обмен данными. (ОК 01-10, ПК 1.2-1.3)
3. Сетевое программирование сокетов. (ОК 01-10, ПК 1.2-1.3)
4. Работы с буфером экрана. (ОК 01-10, ПК 1.2-1.3)

Критерии оценки:

Индекс и расшифровка компетенции	Результаты обучения	Критерии оценивания результатов обучения (баллы)			
		2	3	4	5
ОК-1	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений

	<p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	<p>Не знает Допускает грубые ошибки</p>	<p>Демонстрирует частичные знания без грубых ошибок</p>	<p>Знает достаточно в базовом объеме</p>	<p>Демонстрирует высокий уровень знаний</p>
ОК-2	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p>	<p>Не умеет Демонстрирует частичные умения, допуская грубые ошибки</p>	<p>Демонстрирует частичные умения без грубых ошибок</p>	<p>Умеет применять знания на практике в базовом объеме</p>	<p>Демонстрирует высокий уровень умений</p>
	<p>Знать: основные этапы разработки программного обеспечения; основные принципы</p>	<p>Не знает Допускает грубые ошибки</p>	<p>Демонстрирует частичные знания без грубых ошибок</p>	<p>Знает достаточно в базовом объеме</p>	<p>Демонстрирует высокий уровень знаний</p>

	технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов				
ОК- 3	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний

	<p>ния; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>				
ОК-4	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p>	<p>Не умеет Демонстрирует частичные умения, допуская грубые ошибки</p>	<p>Демонстрирует частичные умения без грубых ошибок</p>	<p>Умеет применять знания на практике в базовом объеме</p>	<p>Демонстрирует высокий уровень умений</p>
	<p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отлад-</p>	<p>Не знает Допускает грубые ошибки</p>	<p>Демонстрирует частичные знания без грубых ошибок</p>	<p>Знает достаточно в базовом объеме</p>	<p>Демонстрирует высокий уровень знаний</p>

	ки и тестирования программных продуктов				
ОК-5	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
ОК-6	Уметь: осуществлять разработку	Не умеет Демонстри-	Демонстрирует частичные	Умеет применять зна-	Демонстрирует высо-

	<p>кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p>	<p>рует частичные умения, допуская грубые ошибки</p>	<p>умения без грубых ошибок</p>	<p>ния на практике в базовом объеме</p>	<p>кий уровень умений</p>
	<p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	<p>Не знает Допускает грубые ошибки</p>	<p>Демонстрирует частичные знания без грубых ошибок</p>	<p>Знает достаточно в базовом объеме</p>	<p>Демонстрирует высокий уровень знаний</p>
<p>ОК- 7</p>	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать про-</p>	<p>Не умеет Демонстрирует частичные умения, допуская грубые ошибки</p>	<p>Демонстрирует частичные умения без грубых ошибок</p>	<p>Умеет применять знания на практике в базовом объеме</p>	<p>Демонстрирует высокий уровень умений</p>

	грамму по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства				
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
ОК- 8	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений

	тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства				
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
ОК- 9	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода про-	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений

	граммного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства				
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
ОК-10	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять опти-	Не умеет Демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений

	мизацию и рефакторинг программного кода; оформлять документацию на программные средства				
	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Не знает Допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
ПК- 1.2	Иметь практический опыт: разработка кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений	Не владеет Демонстрирует низкий уровень владения, допуская грубые ошибки	Демонстрирует частичные владения без грубых ошибок	Владеет базовыми приемами	Демонстрирует владения на высоком уровне
	Уметь: осуществлять	Не умеет	Демонстриру-	Умеет при-	Демонстри-

	<p>лять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p>	<p>Демонстрирует частичные умения, допуская грубые ошибки</p>	<p>ет частичные умения без грубых ошибок</p>	<p>менять знания на практике в базовом объеме</p>	<p>рует высокий уровень умений</p>
	<p>Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	<p>Не знает Допускает грубые ошибки</p>	<p>Демонстрирует частичные знания без грубых ошибок</p>	<p>Знает достаточно в базовом объеме</p>	<p>Демонстрирует высокий уровень знаний</p>
<p>ПК 1.3</p>	<p>Иметь практический опыт: разработка кода программного продукта на основе готовой специфика-</p>	<p>Не владеет Демонстрирует низкий уровень владения, допуская гру-</p>	<p>Демонстрирует частичные владения без грубых ошибок</p>	<p>Владеет базовыми приемами</p>	<p>Демонстрирует владения на высоком уровне</p>

	<p>кации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений</p>	<p>бые ошибки</p>			
	<p>Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p>	<p>Не умеет Демонстрирует частичные умения, допуская грубые ошибки</p>	<p>Демонстрирует частичные умения без грубых ошибок</p>	<p>Умеет применять знания на практике в базовом объеме</p>	<p>Демонстрирует высокий уровень умений</p>
	<p>Знать: основные этапы разработки программного</p>	<p>Не знает Допускает грубые</p>	<p>Демонстрирует частичные знания без</p>	<p>Знает достаточно в базовом объё-</p>	<p>Демонстрирует высокий уровень</p>

	<p>обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов</p>	ошибки	грубых ошибок	ме	знаний
--	---	--------	---------------	----	--------

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Елабужский институт (филиал) федерального государственного автономного
образовательного учреждения высшего образования
«Казанский (Приволжский) федеральный университет»

**Вопросы к дифференцированному зачету
по МДК.01.04 Системное программирование
(ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3)**

1. Что такое программное обеспечение (ОК 01-10, ПК 1.2-1.3)

Ответ:совокупность компьютерных программ и связанных с ними данных, которая содержит инструкции по указанию компьютеру, что и как делать.

2.Что такое алгоритм Банкира

Ответ:Алгоритм Банкера - это алгоритм распределения ресурсов и предотвращения взаимоблокировки, разработанный Эдсгером Дейкстрой, который проверяет безопасность путем моделирования распределения заранее определенных максимально возможных объемов всех ресурсов, а затем выполняет проверку "s-состояния" для проверки возможных условий взаимоблокировки для всех других ожидающих выполнения действий, прежде чем решить, следует ли разрешить продолжение распределения.

2. Что такое алгоритм Медника (ОК 01-10, ПК 1.2-1.3)

Ответ:один из алгоритмов обнаружения тупиков

3. Что такое алгоритм замещения страниц (ОК 01-10, ПК 1.2-1.3)

Ответ:Алгоритм замещения страниц накапливает статистику обращения к памяти (исследуются бит обращения и/или бит модификации каждой страницы) и при необходимости выбирает “самую ненужную страницу”, перемещает её содержимое во внешний файл (своп) и отдаёт освободившуюся физическую страницу для нового отображения.

5.Основные компоненты системы ввода/вывода(ОК 01-10, ПК 1.2-1.3)

Ответ:Основными компонентами системы ввода — вывода являются драйверы, управляющие внешними устройствами, и файловая система. К подсистеме ввода - вывода можно также с некоторой долей условности отнести и диспетчер прерываний, рассмотренный выше.

6.Директивы определения данных (ОК 01-10, ПК 1.2-1.3)

Ответ:Директивы определения данных являются указаниями транслятору на выделение определённого объёма памяти

7.Какие существуют директивы распределения памяти (ОК 01-10, ПК 1.2-1.3)

Ответ:Директива STRUC позволяет определить структуру данных аналогично структурам в языках высокого уровня. Последовательность директив

8. Что такое регистры (ОК 01-10, ПК 1.2-1.3)

Ответ:это обычные переменные, находящиеся внутри центрального процессора, которым присвоены стандартные имена. Обычно регистры используются в качестве одного из операндов при выполнении команд процессором.

9.Архитектура микропроцессоров это (ОК 01-10, ПК 1.2-1.3)

Ответ:то совокупность сведений: о составе его компонентов, об организации обработки в нем информации и обмена информацией с внешними устройствами ЭВМ, о функциональных возможностях микропроцессора, выполняющего команды программы.

10.Предложения языка Ассемблер состоят из (ОК 01-10, ПК 1.2-1.3)

Ответ:В общем случае предложения языка Ассемблера состоят из следующих компонент:
метка или имя;
мнемоника; (имя инструкции)
операнды;
комментарии.

11. Что такое Операнды команд (ОК 01-10, ПК 1.2-1.3)

Ответ:это объекты, над которыми или при помощи которых выполняются действия, задаваемые инструкциями или директивами. Машинные команды могут либо совсем не иметь операндов, либо иметь один или два операнда

12. Язык Ассемблера это (ОК 01-10, ПК 1.2-1.3)

Ответ:система записи программы с детализацией до отдельной машинной команды, позволяющая использовать мнемоническое обозначение команд и символическое задание адресов.

13. Директивы определения данных (ОК 01-10, ПК 1.2-1.3)

Ответ:По директиве DB (define byte, определить байт) определяются данные размером в байт.

14. Что делают команды пересылки общего назначения (ОК 01-10, ПК 1.2-1.3)

Ответ:Команды пересылки данных осуществляют обмен данными или адресами между регистрами МП и ячейками памяти или портами ввода/вывода.

15. Команды загрузки адресных значений и обращения к стеку (ОК 01-10, ПК 1.2-1.3)

Ответ:К этой подгруппе относятся команды LEA, LDS и LES.

Команда загрузки исполнительного адреса LEA имеет формат:

LEA регистр16, память

и загружает в указанный 16-битный регистр адрес (смещение offset) указанной ячейки памяти, адресуемой любым способом.

16. Команды ввода/вывода (ОК 01-10, ПК 1.2-1.3)

Ответ:Команды ввода-вывода IN и OUT пересылают данные между портами ввода-вывода и регистрами EAX (32-разрядный ввод-вывод), AX (16-разрядный ввод-вывод) и AL (8-разрядный ввод-вывод). Команды адресуют порты ввода-вывода по адресу, задаваемому в команде, либо косвенно, при помощи адреса в регистре DX

17. Арифметические операции над двоичными числами. Сложение (вычитание) двоичных чисел без знака (ОК 01-10, ПК 1.2-1.3)

Ответ:Микропроцессор выполняет сложение операндов по правилам сложения двоичных чисел. Проблем не возникает до тех пор, пока значение результата не превышает размерности поля операнда. Например, при сложении операндов размером в байт результат не должен превышать число 255. Если это происходит, то результат оказывается неверен. Рассмотрим, почему так происходит. К примеру, выполним сложение: $254 + 5 = 259$ в двоичном виде. $11111110 + 0000101 = 1\ 00000011$. Результат вышел за пределы восьми бит, и правильное его значение укладывается в 9 битов, а в 8-битном поле операнда осталось значение 3, что, конечно, неверно. Для фиксирования ситуации выхода за разрядную сетку результата, как в данном случае, предназначен флаг переноса cf. Он располагается в бите 0 регистра флагов eflags/flags. Именно установкой этого флага фиксируется факт переноса единицы из старшего разряда операнда. Анализ этого флага можно провести различными способами. Самый простой и доступный — использовать команду условного перехода jc (переход в случае, если в результате работы предыдущей команды флаг cf установился в 1).

В системе команд микропроцессора имеются три команды двоичного сложения:

inc операнд — операция инкремента, то есть увеличения значения операнда на 1;

add операнд_1, операнд_2 — команда сложения с принципом действия:

операнд_1 = операнд_1 + операнд_2;

adc операнд_1, операнд_2 — команда сложения с учетом флага переноса cf. Принцип действия команды:

операнд_1 = операнд_1 + операнд_2 + значение флага переноса cf.

18. Арифметические операции над двоичными числами. Умножение, деление двоичных чисел
Ответ:Умножение двоичных чисел сводится к умножению множимого на каждый разряд множителя с последующим сдвигом и суммированию полученных произведений аналогично умножению в десятичной системе счисления. (ОК 01-10, ПК 1.2-1.3)

19. Арифметические операции над десятичными числами: сложение десятичных чисел

Ответ:Правила выполнения этих операций в десятичной системе хорошо известны — это сложение, вычитание, умножение столбиком и деление углом. (ОК 01-10, ПК 1.2-1.3)

20. Арифметические операции над десятичными числами. Умножение, деление десятичных чисел (ОК 01-10, ПК 1.2-1.3)

Ответ:Для операций с десятичными числами каждая цифра от 0 до 9 представляется двоичной тетрадой и далее операции производятся над тетрадами как над двоичными числами. Такое представление десятичных цифр получило название двоично-десятичного кода, который относится к классу взвешенных кодов.

22. Перечислите команды логических операций. (ОК 01-10, ПК 1.2-1.3)

Ответ:Команды логических операций во многом схожи с командами арифметических операций. Они так же оперируют с двумя восьмиразрядными двоичными числами, одно из которых должно находиться в аккумуляторе, а второе – регистре R, в ячейке памяти M или во втором байте команды.

К группе логических команд условилось относить так же операции сравнения двух чисел и сдвига числа, находящегося в аккумуляторе.

Логические действия выполняются над числами поразрядно, то есть нулевой разряд одного числа логически сравнивается с нулевым разрядом другого числа, и результат сравнения располагается в нулевом разряде аккумулятора. Затем такое же сравнение производится над первыми разрядами чисел, потом над вторыми и т.д. до последних седьмых разрядов.

23. Команды сравнения. (ОК 01-10, ПК 1.2-1.3)

Ответ:Команда сравнения CMP сравнивает два числа, вычитая второе из первого, также как и команда SUB. Отличие команд CMP и SUB состоит в том, что инструкция CMP не сохраняет результат, а лишь устанавливает в соответствии с результатом флаги состояния. Основное назначение команды CMP – это организация ветвлений (условных переходов) в ассемблерных программах

24. Команды сдвигов.(ОК 01-10, ПК 1.2-1.3)

Ответ:Для сдвига влево применяется инструкция shl

25. Способы передачи параметров в подпрограммы (ОК 01-10, ПК 1.2-1.3)

Ответ:Существует два способа передачи параметров в подпрограммы – передача по значению и передача по наименованию. Способ передачи указывается при объявлении параметра в списке формальных параметров.

Тестовые задания по МДК.01.04 Системное программирование

1. В ассемблере команда вычитания

- 1) add
- 2) sub
- 3) mul
- 4) div
- 5) xor

Ответ: 2 (ОК 01-10, ПК 1.2-1.3)

2. В микропроцессоре 80386 нет регистра

- 1) bp
- 2) al
- 3) bl
- 4) el
- 5) sp

Ответ: 4 (ОК 01-10, ПК 1.2-1.3)

3. Разрядность eax

- 1) 4
- 2) 8
- 3) 16
- 4) 24

5) 32

Ответ: 5 (ОК 01-10, ПК 1.2-1.3)

4. Ассемблер – язык

- 1) формальный
- 2) низкого уровня
- 3) высокого уровня
- 4) визуального программирования
- 5) автокодов

Ответ: 2 (ОК 01-10, ПК 1.2-1.3)

5. Результат работы фрагмента программы: `mov dl,8; xor dl,9`

- 1) 1
- 2) 17
- 3) 8
- 4) 9
- 5) 254

Ответ: 1(ОК 01-10, ПК 1.2-1.3)

6. В ассемблере команда сложения

- 1) add
- 2) sub
- 3) mul
- 4) div
- 5) xor

Ответ: 1(ОК 01-10, ПК 1.2-1.3)

7. В микропроцессоре 80386 указатель стека

- 1) ax
- 2) ip
- 3) bx
- 4) sp
- 5) ah

Ответ: 4 (ОК 01-10, ПК 1.2-1.3)

8. Комментарии в ассемблере отделяются символами:

- 1) ;
- 2) *)
- 3) /*
- 4) {
- 5) \

Ответ: 1 (ОК 01-10, ПК 1.2-1.3)

9. Что означает строка на ассемблере «`sl db 5 dup(?)`»:

- 1) Присвоение регистру `sl` содержимого ячейки памяти, адресуемой сегментом, определяемым `es` со смещением, равным сумме содержимого `bx` и числа 5.
- 2) `sl` – адрес переменной, хранящей 5 байт неопределенного значения.
- 3) Присвоение регистру `sl` содержимого ячейки памяти, адресуемой сегментом, определяемым `es` со смещением 5.
- 4) Арифметический сдвиг влево регистра `sl` на 5 разрядов, причем справа разряды заполняются произвольными значениями
- 5) `sl` – переменная, хранящая число 5

Ответ: 2 (ОК 01-10, ПК 1.2-1.3)

10. В ассемблере команда умножения

- 1) add
- 2) sub
- 3) mul
- 4) div

5) xor

Ответ: 3 (ОК 01-10, ПК 1.2-1.3)

11. Нет флага:

- 1) переноса
- 2) нулевого результата
- 3) разрешения прерывания
- 4) регистра адреса
- 5) переполнения

Ответ: 4 (ОК 01-10, ПК 1.2-1.3)

12. Результат работы фрагмента программы: mov dx,401 shr dx

- 1) 400
- 2) 402
- 3) 200
- 4) 802
- 5) 15983

Ответ: 3 (ОК 01-10, ПК 1.2-1.3)

13. Объявление сегмента кода на ассемблере

- 1) CODESG SEGMENT PARA 'Code'
- 2) CS SEGMENT
- 3) program SEGMENT
- 4) program code

Ответ: 1 (ОК 01-10, ПК 1.2-1.3)

14. jne в ассемблере

- 1) нет такой команды
- 2) переход, если не равно
- 3) переход к подпрограмме
- 4) конец программы
- 5) начало блока описания параметров подпрограммы

Ответ: 2 (ОК 01-10, ПК 1.2-1.3)

15. В ассемблере команда деления

- 1) add
- 2) sub
- 3) mul
- 4) div
- 5) imul

Ответ: 4 (ОК 01-10, ПК 1.2-1.3)

16. Системная программа:

- 1) Microsoft Word
- 2) Paint
- 3) Дефрагментация
- 4) Corel Draw
- 5) Пасьянс

Ответ: 3 (ОК 01-10, ПК 1.2-1.3)

17. Сколько байт определяет команда DQ:

- 1) 1
- 2) 2
- 3) 4
- 4) 8
- 5) 10

Ответ: 4 (ОК 01-10, ПК 1.2-1.3)

18. На языке ассемблера выполняет действия, противоположные push

- 1) and

- 2) xor
- 3) xlat
- 4) pop
- 5) mov

Ответ: 4 (ОК 01-10, ПК 1.2-1.3)

19. МП 80386 является:

- 1) 8-разрядным;
- 2) 16-разрядным;
- 3) 24-разрядным;
- 4) 32-разрядным

Ответ: 4 (ОК 01-10, ПК 1.2-1.3)

20. Внутренний КЭШ процессора Pentium разделен на:

- 1) КЭШ команд и КЭШ разрядов;
- 2) КЭШ разрядов и КЭШ данных
- 3) КЭШ команд и КЭШ адресов;
- 4) КЭШ команд и КЭШ данных.

Ответ: 4 (ОК 01-10, ПК 1.2-1.3)

21. Для программиста доступна:

- 1) вся рабочая память процессора;
- 2) внутренняя память процессора недоступна;
- 3) внутренняя память доступна через регистры;
- 4) внутренняя память доступна через информационную магистраль.

Ответ: 3 (ОК 01-10, ПК 1.2-1.3)

22. Аккумулятор используется:

- 1) для указания на стек;
- 2) для битового сложения;
- 3) как место для проведения операций и сохранения их результатов;
- 4) как регистр приемника.

Ответ: 3 (ОК 01-10, ПК 1.2-1.3)

23. Говоря о 16-разрядной ЭВМ, имеют в виду:

- 1) разрядность шины данных 16 бит;
- 2) разрядность шины адреса 16 бит;
- 3) размер слова 16 бит;
- 4) размер внутренних регистров памяти 16 бит.

Ответ: 4 (ОК 01-10, ПК 1.2-1.3)

24. В методе косвенной адресации адрес памяти содержится:

- 1) в одном из регистров;
- 2) в команде;
- 3) в стеке;
- 4) в ссылке на команду.

Ответ: 1 (ОК 01-10, ПК 1.2-1.3)

**Практические задания к дифференцированному зачету
по МДК.01.04 Системное программирование
(ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3)**

1. Написать программу вычисления суммы элементов массива. Количество элементов массива равно 10, беззнаковые, размерностью байт. Результат должен поместиться в переменной размерностью слово.
2. Написать программу вычисления произведения элементов массива. Количество элементов массива равно 10, знаковые, размерностью в слово. Результат должен поместиться в переменной размерностью в двойное слово.
3. Задать массив размерностью 20 с произвольными числами. Вычислить максимальный элемент массива.
4. Задать массив размерностью 20 с произвольными числами. Вычислить минимальный элемент массива.
5. Написать программу, считающую количество символов введенной строки.
6. Написать программу для подсчета количества вхождений заданного символа в строку текста.
7. Написать программу для замены заданного символа в тексте новым.
8. Объявите структуру с двумя массивами (mas1 и mas2) одинаковой размерности. Вычислите, сумма элементов какого массива имеет большее значение.
9. Написать программу, переписывающую в обратном порядке любые введенные с клавиатуры символы.
10. Написать программу, которая делит на 4 все элементы одномерного байтового массива.
11. Написать программу, которая выполняет операцию взятия модуля от байтового числа, т.е. из числа -112 она сделает 112, а положительное число 112 оставит без изменений.
12. Написать программу, инвертирующее байтовое число, т.е. число 25 превратит в -25. Число -127 превратит в 127.

), ПК 1.2-1.3)

Критерии оценки при проведении дифференцированного зачета

«отлично»	«хорошо»	«удовлетворительно»	«неудовлетворительно»
студент раскрывает теоретический вопрос билета, практическое задание выполняет без ошибок, уверенно отвечает на дополнительные вопросы	студент раскрывает теоретический вопрос, практическое задание выполняет без ошибок, на дополнительные вопросы отвечает неуверенно, допускает не точности в определениях.	студент раскрывает теоретический вопрос не в полной мере, допускает неточности в формулировках (1-2 ошибки), практическое задание выполнено частично, с допущением ошибок в расчётах	Теоретический вопрос не раскрыт, практическое задание не выполнено.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Елабужский институт (филиал) федерального государственного автономного
образовательного учреждения высшего образования
«Казанский (Приволжский) федеральный университет»



**РАБОЧАЯ ПРОГРАММА ЭКЗАМЕНА ПО МОДУЛЮ
ПМ.01 Разработка модулей программного обеспечения для компьютерных систем**

г.Елабуга, 2024

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Экзамен по модулю является частью оценки качества освоения программы подготовки специалистов среднего звена по специальности 09.02.07 «Информационные системы и программирование» и является обязательной процедурой для всех студентов, обучающихся по программам подготовки специалистов среднего звена в Елабужском институте (филиале) федерального государственного автономного образовательного учреждения высшего образования «Казанский (Приволжский) федеральный университет».

Программа экзамена по модулю при реализации программы подготовки специалистов среднего звена по специальности 09.02.07 «Информационные системы и программирование» (далее – Программа) представляет собой совокупность требований к подготовке и проведению экзамена по модулю по ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем».

Программа разработана на основе нормативных актов Российской Федерации и соответствующих положений Министерства образования и науки Российской Федерации.

2. ПАСПОРТ ПРОГРАММЫ ЭКЗАМЕНА ПО МОДУЛЮ

Специальность среднего профессионального образования

09.02.07 «Информационные системы и программирование»

Наименование квалификации: программист

Уровень образования, необходимый для приема на обучение:

основное общее образование

Срок получения среднего профессионального образования по программе подготовки специалистов среднего звена: 3 года 10 месяцев

Исходные требования к подготовке и проведению экзамена по модулю по ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем»»

Форма проведения экзамена по модулю	Выполнение комплексного практического задания
Объем времени на подготовку и проведение экзамена по модулю	Подготовка <u>1</u> неделя Проведение <u>1</u> день
Сроки проведения экзамена по модулю	7 семестр

Образовательные результаты по итогам освоения ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем»»

<i>Вид профессиональной деятельности:</i> Разработка модулей программного обеспечения для компьютерных систем	
Индекс компетенции	Расшифровка приобретаемой компетенции
ПК 1.1	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием
ПК 1.3	Выполнять отладку программных модулей с использованием специализированных программных средств
ПК 1.4	Выполнять тестирование программных модулей
ПК 1.5	Осуществлять рефакторинг и оптимизацию программного кода
ПК 1.6	Разрабатывать модули программного обеспечения для мобильных платформ

3.

УСЛОВИЯ ПОДГОТОВКИ И ПРОВЕДЕНИЯ ЭКЗАМЕНА ПО МОДУЛЮ

3.1. Документационное обеспечение подготовки и проведения экзамена по модулю

№ п/п	Наименование документа
1	Федеральный государственный образовательный стандарт среднего профессионального образования по специальности 09.02.07 «Информационные системы и программирование».
2	Документы, подтверждающие освоение обучающимися компетенций при изучении теоретического материала и прохождения практики (журналы теоретического обучения и практик за весь период обучения модуля, сводная ведомость успеваемости обучающихся, зачетные книжки обучающихся, аттестационные листы).
3	Протокол(ы) заседаний комиссии

3.2. Техническое обеспечение подготовки и проведения экзамена по модулю

№ п/п	Наименование	Требование
1	Оборудование	персональный компьютер медиапроектор;
2	Рабочие места	стандартный кабинет с посадочными местами для подготовки и защиты
3	Материалы	Распечатанные практические задания
5	Аудитория	учебный кабинет

4. ФОРМА ПРОВЕДЕНИЯ ЭКЗАМЕНА ПО МОДУЛЮ

Выполнение комплексного практического задания, соответствующего видам деятельности ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем»

4.1.1. Требования к практическим заданиям

Выполнение профессионального комплексного практического задания, разработанного на базе фонда оценочных средств Всероссийской олимпиады профессионального мастерства обучающихся по специальностям СПО

4.1.2. Требования к процедуре проведения экзамена по модулю

№ п/п	Этапы экзамена	Содержание
1.	Решение практических задач (30 минут)	Студенты решают предложенные практические задачи
2.	Представление результатов и ответы студентов на вопросы комиссии	Представление результатов работы и ответы студентов на вопросы членов комиссии, непосредственно связанные с рассматриваемыми вопросами практических задач.
3	Принятие решения комиссией по результатам проведения экзамена	Решение комиссии об оценке принимаются на закрытом заседании открытым голосованием простым большинством голосов членов комиссии, участвовавших в заседании. При равном числе голосов голос председателя является решающим.
4	Документальное оформление результатов экзамена	Фиксирование решений комиссии в протоколах.

5. КРИТЕРИИ ОЦЕНКИ

5.1. Критерии оценки результатов экзамена по модулю

«Отлично»

Студент обнаруживает глубокие знания при решении практических задач. Выводы и рекомендации аргументированы. Материал изложен в определенной логической последовательности, осознанно, литературным языком, с использованием современных научных терминов.

Основные задания раскрыты полностью и без ошибок. Развернуто аргументируют выдвигаемые положения, приводят убедительные примеры. Обнаруживает аналитический подход при подготовке решения практических задач. Делает содержательные выводы.

При ответе на вопросы комиссии студент отвечает четко и грамотно демонстрирует свободное владение материалом.

«Хорошо»

Студент обнаруживает хорошие знания при решении практических задач. Выводы и рекомендации достаточно аргументированы. Материал изложен в логической последовательности, осознанно, литературным языком, с использованием современных научных терминов.

Основные задания в целом раскрыты, имеются незначительные ошибки. Студент аргументирует выдвигаемые положения, приводит примеры. Обнаруживает аналитический подход при подготовке решения практических задач. Делают достаточно содержательные выводы.

При ответе на вопросы членов комиссии студент отвечает по существу и демонстрирует хорошее владение материалом работы.

«Удовлетворительно»

Студент обнаруживают удовлетворительные знания при решении практических задач. Выводы и рекомендации аргументированы недостаточно. Материал изложен последовательно.

Основные задания раскрыты не полностью, имеются значительные ошибки. Студент не аргументирует выдвигаемые положения, примеры носят общий характер.

Ответы на вопросы членов комиссии вызывают у студента затруднения, студент демонстрирует удовлетворительное владение материалом работы.

«Неудовлетворительно»

Студент обнаруживает неудовлетворительные знания при решении практических задач. Выводы и рекомендации аргументированы неверно. Материал изложен бессвязно и непоследовательно.

Основные задания не решены, либо решены неверно, имеются критические ошибки. Студент не аргументирует выдвигаемые положения, примеры отсутствуют.

На вопросы членов комиссии студент не отвечает, либо отвечает неверно, демонстрирует неудовлетворительное владение материалом работы.

Комплексные задания, проверяющие освоение группы компетенций:

1. Создать приложение, которое состоит из двух форм. Первая форма (парольная) располагается в центре экрана. Из первой формы вызывается вторая (главная) форма. Расположить на ней два однострочных редактора и один многострочный. В однострочные редакторы вводятся Фамилия и Телефон клиента (заполнение полей контролируется либо программно, либо по маске). Оба значения объединяются в одну строку и добавляются очередной записью в многострочный редактор. При попытке закрытия второй формы выдать запрос-подтверждение о необходимости закрытия формы. (ПК 1.1-1.6)
2. Создать приложение, состоящее из двух форм. Первая форма должна выполнять роль заставки и всегда находиться поверх других окон. Разместить на первой форме какое-либо изображение и кнопку, при нажатии на которую отображается вторая форма. Установить на второй форме компонент Вкладки и расположить на них 3 таблицы - одну с исходными данными о работнике: ФИО, должность, дата рождения, зарплата. Во вторую выбирать работников, родившихся завтра, в третью выбрать тех работников, ФИО которых начинается на букву «М» с зарплатой больше 7000 из менеджеров. (ПК 1.1-1.6)

3. Создать приложение, которое состоит из двух форм. Первая форма располагается в центре экрана, содержит текстовое поле с приветствием и кнопку открытия второй формы. На второй форме расположить компонент StringGrid, размер которого задается с помощью двух текстовых полей. Таблицу заполнить случайными целыми числами от -100 до 100. Найти в таблице максимальный по модулю элемент и выделить контрастным цветом все столбцы и все строки, в которых встречается этот элемент. (ПК 1.1-1.6)
4. Создать приложение, состоящее из двух форм. На первой форме располагаются компоненты для вывода приветствия, авторизации пользователя и настройки графического интерфейса (задание цвета, шрифта). Из первой формы вызывается вторая (главная) форма. На второй форме разместить текстовое поле и три переключателя, при выборе каждого из которых активизируется соответствующая ему процедура: найти каких цифр «3» или «5» в тексте больше; заменить все «,» на «*»; вывести новый текст на форму. (ПК 1.1-1.6)
5. Дана программа (без исходного кода). Описан её функционал. Необходимо спроектировать тесты, описать их, провести тестирование, зафиксировать его результаты и сделать выводы о наличии ошибок в программе. (ПК 1.1-1.6)
6. Дан программный код, описано его назначение. В коде заведомо есть ошибка, которая проявляется не при каждом тестовом наборе данных. Необходимо составить два тестовых набора: тот, при котором ошибка проявит себя, и тот, при котором не проявит. (ПК 1.1-1.6)
7. Требуется разработать приложение с графическим пользовательским интерфейсом, поддерживающее создание/редактирование/удаление/поиск записок «Записная книжка». Используя два варианта хранения записок: в базе SQLite или с использованием файловой системы. (ПК 1.1-1.6)
8. Создайте приложение «Карманный навигатор» с графическим пользовательским интерфейсом с функциями: определение местоположения пользователя на карте Google Map; определение скорости и направления движения пользователя; масштабирование карты. Программа должна быть конфигурируемой. (ПК 1.1-1.6)
9. Разработайте приложение-таймер «Песочные часы» с использованием датчика ориентации в виде песочных часов. Каждый раз для того чтобы активировать таймер, необходимо перевернуть экран мобильного устройства вверх ногами. Используйте анимацию для показа «перетекающего песка» и переворота песочных часов. Для задания времени перетекания песка требуется разработать push-notification сервер. Через форму ввода на сервере можно отправлять на клиент (приложение-таймер) указанное время (числовой ввод). (ПК 1.1-1.6)
10. Требуется разработать приложение для обмена мгновенными сообщениями через Wi-Fi/Bluetooth. Поддерживаемые режимы: активный режим (приложение занимает весь экран, содержит поля для отправки сообщений и список принятых сообщений) и режим уведомлений (приложение через уведомления показывает принятые сообщения). (ПК 1.1-1.6)
11. Написать программу вычисления суммы элементов массива. Количество элементов массива равно 10, беззнаковые, размерностью байт. Результат должен поместиться в переменной размерностью слово. (ПК 1.1-1.6)
12. Написать программу вычисления произведения элементов массива. Количество элементов массива равно 10, знаковые, размерностью в слово. Результат должен поместиться в переменной размерностью в двойное слово. (ПК 1.1-1.6)
13. Задать массив размерностью 20 с произвольными числами. Вычислить максимальный элемент массива. (ПК 1.1-1.6)
14. Задать массив размерностью 20 с произвольными числами. Вычислить минимальный элемент массива. (ПК 1.1-1.6)
15. Написать программу, считающую количество символов введенной строки. (ПК 1.1-1.6)
16. Написать программу для подсчета количества вхождений заданного символа в строку текста. (ПК 1.1-1.6)

17. Написать программу для замены заданного символа в тексте новым.
18. Объявите структуру с двумя массивами (mas1 и mas2) одинаковой размерности. Вычислите, сумма элементов какого массива имеет большее значение.
19. Написать программу, переписывающую в обратном порядке любые введенные с клавиатуры символы.
20. Написать программу, которая делит на 4 все элементы одномерного байтового массива. (1-1.6)
21. Написать программу, которая выполняет операцию взятия модуля от байтового числа, т.е. из числа -112 она сделает 112, а положительное число 112 оставит без изменений.
22. Написать программу, инвертирующее байтовое число, т.е. число 25 превратит в -25. Число -127 превратит в 127.
23. Написать программу с использованием процедур, которая запрашивает строку (ввод с клавиатуры), и затем переводит все символы по следующему алгоритму: Если символ в нижнем регистре, перевести его в верхний регистр; если в верхнем – в нижний.
24. Написать программу с использованием процедур, которая запрашивает строку (ввод с клавиатуры), и затем переводит все символы по следующему алгоритму: Вывести строку, в закодированном виде, от каждого кода символа строки отнимается число 10.
25. Написать программу с использованием процедур, которая запрашивает строку (ввод с клавиатуры), и затем переводит все символы по следующему алгоритму: Найти позицию символа (вводится с клавиатуры) в строке и вывести позицию (и) в шестнадцатеричном виде.