Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Умаров Марат Файзуллаевич
Должность: Директор
Дата подписания: 17.02.2026 09:50:16
Уникальный программный ключ:
48505f11ec15aca386f3219d3113d727fefda78

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
«1» марта 2024 г.

ПРОГРАММА ДИСЦИПЛИНЫ

ОП.04 «Основы алгоритмизации и программирования»

Специальность: 09.02.07 Информационные системы и программирование
На базе: основного общего образования
Квалификация: программист
Форма обучения: очная
Язык обучения: русский
Год начала обучения: 2024

Елабуга, 2024

1. Цели освоения дисциплины

формирование знаний по:

- понятию алгоритмизации, свойств алгоритмов, общих принципов построения алгоритмов, основным алгоритмическим конструкциям;
- эволюции языков программирования, их классификации, понятию системы программирования;
- основным элементам языка, структуре программы, операторам и операциям, управляющим структуры, структуре данных, файлам, классам памяти;
- подпрограммам, составлению библиотек подпрограмм
- объектно-ориентированной модели программирования, основным принципам объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

формирование умений по:

- разработке алгоритмов для конкретных задач;
- использованию программ для графического отображения алгоритмов;
- определению сложностей работы алгоритмов;
- работе в среде программирования;
- оформлению кода программы в соответствии со стандартом кодирования;
- выполнению проверки, отладки кода программы.

2. Место дисциплины в структуре ППССЗ

Учебная дисциплина ОП.04 «Основы алгоритмизации и программирования» является общепрофессиональной дисциплиной профессионального цикла в соответствии с ФГОС по специальности СПО 09.02.07 «Информационные системы и программирование».
Осваивается на втором и третьем курсах курсе (3, 4 и 5 семестры).

3. Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины формируются компетенции:

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 04	Эффективно взаимодействовать и работать в коллективе и команде;
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста
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 2.4	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения
ПК 2.5	Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования

В результате освоения дисциплины обучающийся должен **знать:**

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.
- В результате освоения дисциплины обучающийся должен **уметь:**
- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

4. Структура и содержание дисциплины

4.1. Общая трудоемкость дисциплины в часах

Общая трудоемкость дисциплины составляет 188 часов.

Форма промежуточной аттестации по дисциплине: дифференцированный зачет в 4 семестре, экзамен в 5 семестре.

Разделы и темы дисциплины		Семестр	Виды и часы аудиторной работы, их трудоемкость (в часах)			Самостоятельная работа	Консультации	Текущие формы контроля
			Лекции	Практические занятия	Лабораторные работы			
Тема 1	**Теоретические основы алгоритмизации и программирования**	3	16	0	16	2	1	Контрольная работа 1*

Тема 2	**Основы программирования. Программирование структур данных**	4	46	0	46	2	1	Контрольная работа 2*
Тема 3	**Объектно-ориентированное программирование**	5	24	0	24	2	1	Защита программного приложения
Всего: 181			**86**	**0**	**86**	**6**	**3**	
Дифференцированный зачет			**0**					
Экзамен			**7**					
Всего по дисциплине			**188**					

* письменная контрольная точка (выполняется за компьютером)

4.2. Содержание дисциплины

Наименование разделов и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся	Объем часов (лек/ лаб.р./самост)	Уровень освоения*
1	**2**	**3**	**4**
Тема 1 Теоретические основы алгоритмизации и программирования	**Содержание учебного материала**	**34 (16/16/2)**	
	1. Развитие языков программирования.	16	*1*
	2. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы.		
	3. Жизненный цикл программы. Программа. Программный продукт и его характеристики.		
	4. Основные этапы решения задач на компьютере.		
	Лабораторные работы: 1.Структурная организация данных. Модели объектов и процессов. 2.Эволюция языков программирования. Правила оформления текстов программ. 3.Функция сложности алгоритма. Методы сортировки и поиска. 4.Итеративные и рекурсивные алгоритмы. 5.Основные определения теории графов. Алгоритмы построения остовного (покрывающего) дерева сети. Алгоритмы нахождения кратчайших путей на графах. 6.Эвристические алгоритмы. Метод ветвей и границ. Задача коммивояжера. 7.Моделирование с использованием генераторов случайных чисел. 8.Машина Тьюринга. Элементы математической логики.	16	2
	Самостоятельная работа обучающегося Подготовка к контрольной работе по темам: Функция сложности алгоритма. Методы сортировки. Методы поиска. Итеративные и рекурсивные алгоритмы. Колдаев, В. Д. Основы алгоритмизации и программирования: учебное пособие / В.Д. Колдаев; под ред. проф. Л.Г. Гагариной. — Москва: ФОРУМ: ИНФРА-М, 2022. — 414 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0733-7. - Текст: электронный. - URL: https://znanium.com/read?id=378685 (дата обращения: 28.08.2021). – Режим доступа: по подписке. Стр. 316-321	2	*3*
Тема 2. Типы данных	**Содержание учебного материала**	**94 (46/46/2)**	

	1. Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных.	46	*1*
	2. Этапы решения задач на компьютере.		
	3. Основные понятия языка программирования Python		
	4. Введение в Python		
	5. Линейный алгоритм		
	6. Разветвляющийся алгоритм		
	7. Циклический алгоритм		
	8. Работа с кортежами и списками		
	9. Работа со строками		
	10. Обработка вложенных последовательностей		
	11. Работа с функциями. Создание модулей		
	12. Работа с файлами		
	Лабораторные работы 1. Алгоритм. Свойства алгоритма. Способы описания алгоритма. 2. Этапы решения задач на компьютере. 3. Основные понятия языка программирования Python 4. Введение в Python 5. Линейный алгоритм 6. Разветвляющийся алгоритм 7. Циклический алгоритм 8. Работа с кортежами и списками 9. Работа со строками 10. Обработка вложенных последовательностей 11. Работа с функциями. Создание модулей 12. Работа с файлами	46	2
	Самостоятельная работа обучающегося Подготовка к контрольной работе по темам: Разветвляющийся алгоритм; Циклический алгоритм; Работа с кортежами и списками; Работа со строками; Обработка вложенных последовательностей; Работа с функциями; Создание модулей; Работа с файлами. Гуриков, С. Р. Основы алгоритмизации и программирования на Python: учебное пособие / С.Р. Гуриков. — Москва: ИНФРА-М, 2022. — 343 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-017142-5. - Текст: электронный. - URL: https://znanium.com/read?id=379975 (дата обращения: 28.08.2021). – Режим доступа: по подписке. Стр. 41-241	2	*3*

Тема 3. Объектно-ориентированное программирование	**Содержание учебного материала**	**50 (24/24/2)**	
	1. Понятия объектно-ориентированного программирования	24	*1*
	2. Создание классов		
	3. Создание конструкторов		
	4. Инкапсуляция		
	5. Создание свойств		
	6. Наследование		
	7. Примеры решения задач		
	8. Событийно-ориентированное программирование		
	9. Создание формы и виджетов Кнопка, Текстовое поле, Надпись		
	10. Создание виджета Флажок		
	11. Создание виджета Переключатель		
	12. Примеры решения задач		
	Лабораторные работы 1. Понятия объектно-ориентированного программирования 2. Создание классов 3. Создание конструкторов 4. Инкапсуляция 5. Создание свойств 6. Наследование 7. Примеры решения задач 8. Событийно-ориентированное программирование 9. Создание формы и виджетов Кнопка, Текстовое поле, Надпись 10. Создание виджета Флажок 11. Создание виджета Переключатель 12. Примеры решения задач	24	2
	Самостоятельная работа обучающегося Разработка программного приложения по темам: Объектно-ориентированное программирование, событийное программирование. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва: ИНФРА-М, 2022. — 343 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-017142-5. - Текст: электронный. - URL: https://znanium.com/read?id=379975 (дата обращения: 28.08.2021). – Режим доступа: по подписке. Стр. 243 – 281	2	*3*

Примерная тематика курсовой работы (проекта)	не предусмотрено	
Всего	**178 (86/86/6)**	
Консультация	**3**	
Экзамен	**7**	
Всего по дисциплине	**188**	

*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)

4.3. Структура и содержание самостоятельной работы студентов

Темы дисциплины		Виды самостоятельной работы	Трудоемкость (в часах)	Формы контроля самостоятельной работы
1	2	3	4	5
Тема 1	Теоретические основы алгоритмизации и программирования	**Самостоятельная работа обучающегося** Подготовка к контрольной работе по темам: Функция сложности алгоритма. Методы сортировки. Методы поиска. Итеративные и рекурсивные алгоритмы. Колдаев, В. Д. Основы алгоритмизации и программирования : учебное пособие / В.Д. Колдаев ; под ред. проф. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2022. — 414 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0733-7. - Текст : электронный. - URL: https://znanium.com/read?id=378685 (дата обращения: 28.08.2021). – Режим доступа: по подписке. Стр. 316-321	2	Контрольная работа 1
Тема 2	Основы программирования. Программирование структур данных	**Самостоятельная работа обучающегося** Подготовка к контрольной работе по темам: Разветвляющийся алгоритм; Циклический алгоритм; Работа с кортежами и списками; Работа со строками; Обработка вложенных последовательностей; Работа с функциями; Создание модулей; Работа с файлами. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2022. — 343 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-017142-5. - Текст : электронный. - URL: https://znanium.com/read?id=379975 (дата обращения: 28.08.2021). – Режим доступа: по подписке. Стр. 41-241	2	Контрольная работа 2*
Тема 3	Объектно-ориентированное программирование	**Самостоятельная работа обучающегося** Разработка программного приложения по темам: Объектно-ориентированное программирование, событийное программирование. Гуриков, С. Р. Основы алгоритмизации и программирования на Python: учебное пособие / С.Р. Гуриков. — Москва :	2	Разработка программного приложения

		ИНФРА-М, 2022. — 343 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-017142-5. - Текст: электронный. - URL: https://znanium.com/read?id=379975 (дата обращения: 28.08.2021). – Режим доступа: по подписке. Стр. 243 – 281		
Всего по дисциплине			**6**	

5. Образовательные технологии

Освоение дисциплины «Основы алгоритмизации и программирования» предполагает использование как традиционных (лекции, практические занятия с использованием методических материалов), так использованием в учебном процессе активных и интерактивных форм проведения занятий:
работа в малых группах, решение кейсов (анализ реальных проблемных ситуаций, имевших место в соответствующей области профессиональной деятельности, и поиск вариантов лучших решений), проблемное обучение (стимулирование студентов к самостоятельному приобретению знаний, необходимых для решения конкретной проблемы). Выполнение заданий требует использования не только учебников и пособий, но и информации, содержащейся в Интернете.
На лекциях и практических занятиях используются:
- информационная и презентационная лекция;
- тематические опросы;
- решение задач;
- коллективное выполнение заданий в подгруппах для обобщения тематического теоретического материала в схемах, таблицах;
- выполнение практических заданий с использованием среды программирования разработки PyCharm Community Edition.

Занятия, проводимые в активной и интерактивной формах

Номер темы	Наименование темы	Форма проведения занятия	Объем в часах
Тема 1	Теоретические основы алгоритмизации и программирования	Информационно-проблемная лекция	16
Тема 2	Основы программирования. Программирование структур данных	Информационно-проблемная лекция	46
Тема 3	Объектно-ориентированное программирование	Информационно-проблемная лекция	24
Всего по дисциплине			86

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

6.1 Оценочные средства для текущего контроля успеваемости

Контрольная работа 1 (ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ОК 10, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5)

Пример заданий контрольной работы 1:

1 Чем характеризуется сложность алгоритма?
2. Чем различаются простые и интегрированные структуры данных?
3 Как получается верхняя оценка сложности алгоритма?
4. Перечислите основные типы связных списков.
5 Какие функции используются для представления верхней оценки сложности алгоритма?

6. Чем различаются стек, очередь и дек?
7 Как оценивается сложность экспериментальным методом?
8. Приведите примеры графовых структур.
9 Как влияет размер массива на временную сложность алгоритма?
10. Перечислите свойства алгоритма.
11 Как определяется емкостная сложность алгоритма?

Контрольная работа 2 (ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ОК 10, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5)

Пример заданий контрольной работы 2:

1. В кортеже целых чисел вычислите произведение отрицательных элементов, имеющих нечетные индексы.
2. Разработайте программу, которая проверяет, является ли введенная с клавиатуры последовательность символов целым числом, записанным в двоичной системе счисления.
3. Список A содержит N чисел. Найдите количество элементов списка, значения которых превышают заданное значение znach. Назначение функции: подсчет количества элементов, превышающих заданное значение. Оформите созданную функцию в виде программного модуля и подключите его к основной программе.
4. Из строки, расположенной в текстовом файле, исключите все символы, входящие в нее более одного раза. Полученную строку сохраните в другом файле.

Защита программного приложения (ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ОК 10, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5)

Примеры заданий:

1. Разработайте тестовую GUI-программу на тему: Российские кинофильмы.
2. Разработайте тестовую GUI-программу на тему: Спорт.
3. Разработайте тестовую GUI-программу на тему: Автомобили.
4. Разработайте тестовую GUI-программу на тему: Российская эстрада.
5. Разработайте тестовую GUI-программу на тему: Программирование.
6. Разработайте тестовую GUI-программу на тему: Животные.
7. Разработайте тестовую GUI-программу на тему: Города России.

6.2 Оценочные средства для промежуточной аттестации по итогам освоения дисциплины

Пример билета к дифференцированному зачёту (ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ОК 10, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5)

1. Расскажите о назначении IDLE. С какими способами создания программ в IDLE вы познакомились?
2. Перечислите основные функции для работы с символами. Приведите примеры.

Пример билета к экзамену (ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ОК 10, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5)

1. Назовите и поясните два основных аспекта объектно-ориентированного программирования.
2. В чем заключается такой принцип ООП, как инкапсуляция?
3. 1. Создайте класс ФИГУРА с методами вычисления площади и периметра, а также методом, выводящим информацию о фигуре на экран. Создайте дочерние классы ПРЯМОУГОЛЬНИК, КРУГ, ТРЕУГОЛЬНИК со своими методами вычисления площади и периметра. Создайте список n фигур и выведите полную информацию о фигурах на экран.

7 Таблица соответствия компетенций, критериев оценки их освоения и оценочных средств

Шифр компетенции	Планируемые результаты обучения	Оценочные средства	Критерии оценивания результатов обучения (баллы)			
			2	3	4	5
ОК 01	Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь разрабатывать алгоритмы для конкретных задач; определять сложность работы алгоритмов	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений
ОК 02	Знать эволюцию языков программирования, их классификацию, понятие системы программирования	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний

	Уметь оформлять код программы в соответствии со стандартом кодирования	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений
ОК 04	Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь реализовывать построенные алгоритмы в виде программ на конкретном языке программирования; определять сложность работы алгоритмов	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений

ОК 05	Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь реализовывать построенные алгоритмы в виде программ на конкретном языке программирования	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений
ОК 09	Знать подпрограммы, составление библиотек подпрограмм	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь работать в среде программирования	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений

ОК 10	Знать объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь использовать программы для графического отображения алгоритмов	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений

ПК 1.1	Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь оформлять код программы в соответствии со стандартом кодирования	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений
ПК 1.2	Знать подпрограммы, составление библиотек подпрограмм	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь реализовывать построенные алгоритмы в виде программ на конкретном языке программирования	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений

ПК 1.3	Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь выполнять проверку, отладку кода программы	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений
ПК 1.4	Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь выполнять проверку, отладку кода программы	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений

ПК 1.5	Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь выполнять проверку, отладку кода программы	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений
ПК 2.4	Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь работать в среде программирования	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений

ПК 2.5	Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь работать в среде программирования	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений

8. Методические указания для обучающихся при освоении дисциплины

Работа на лабораторных занятиях предполагает активное участие в дискуссиях и решении задач. Для подготовки к занятиям рекомендуется выделять в материале проблемные вопросы, затрагиваемые преподавателем в лекции, и группировать информацию вокруг них.

При работе с терминами необходимо обращаться к словарям, в том числе доступным в Интернете, например на сайте http://dic.academic.ru.

При подготовке обучающихся по темам используются конспекты лекций и источники основной и дополнительной литературы. Подготовка докладов осуществляется с использованием нормативно-правовых документов и учебников.

Устный опрос по этой теме проводится в форме беседы. Работа на практических занятиях предполагает активное участие в дискуссиях и решении задач. Для подготовки к занятиям рекомендуется выделять в материале проблемные вопросы, затрагиваемые преподавателем в лекции, и группировать информацию вокруг них.

Решение задач проводится в группе с обсуждением хода решения, применяемых, схем, способов, проверкой результатов и проведением работы над ошибками.

Тестирование проводится после ознакомления с материалом темы. Обучающийся выполняет тестирование, рассчитанное по времени на 40-50 минут, на бумажном носителе. Тест включает в себя задания разного типа: на выбор одного или нескольких правильных ответов, на соответствие, краткий и числовой ответ. Для прохождение теста дается одна попытка. Далее сверяются и обсуждаются результаты с определением правильных ответов.

Промежуточная аттестация по этой дисциплине проводится в форме экзамена. При подготовке к экзамену необходимо опираться, прежде всего, на источники, которые разбирались на лекционных занятиях и на материалы практических занятий. В каждом билете экзамена содержится два вопроса.

9. Учебно-методическое и информационное обеспечение дисциплины

9.1. Основная литература:

1. Голицына, О. Л. Основы алгоритмизации и программирования : учебное пособие / О.Л. Голицына, И.И. Попов. — 4-е изд., испр. и доп. — Москва : ФОРУМ : ИНФРА-М, 2021. — 431 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-570-7. - Текст : электронный. - URL: https://znanium.com/catalog/product/1150328 (дата обращения: 28.08.2023). – Режим доступа: по подписке.
2. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2022. — 343 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-017142-5. - Текст : электронный. - URL: https://znanium.com/catalog/product/1356003 (дата обращения: 28.08.2023). – Режим доступа: по подписке.
3. Канцедал, С. А. Алгоритмизация и программирование: учебное пособие / С. А. Канцедал. — Москва: ФОРУМ: ИНФРА-М, 2021. — 352 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0727-6. - Текст: электронный. - URL: https://znanium.com/catalog/product/1189320 (дата обращения: 14.09.2023). – Режим доступа: по подписке
4. Колдаев, В. Д. Основы алгоритмизации и программирования : учебное пособие / В.Д. Колдаев ; под ред. проф. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2022. — 414 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0733-7. - Текст : электронный. - URL: https://znanium.com/catalog/product/1735805 (дата обращения: 28.08.2023). – Режим доступа: по подписке.

9.2. Дополнительная литература:

1. Ахмедханлы, Д. М. Основы алгоритмизации и программирования : учебно-методическое пособие / Д. М. Ахмедханлы, Н. В. Ушмаева. — Тольятти : ТГУ, 2016. — 123 с. — ISBN 978-5-8259-1022-2. — Текст : электронный // Лань : электронно-библиотечная система. —

URL: https://e.lanbook.com/book/139958 (дата обращения: 28.08.2021). — Режим доступа: для авториз. пользователей.
2. Бедердинова, О. И. Основы алгоритмизации и структурного программирования : учебное пособие / О. И. Бедердинова. — Архангельск : САФУ, 2017. — 88 с. — ISBN 978-5-261-01227-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/161718 (дата обращения: 28.08.2021). — Режим доступа: для авториз. пользователей.
3. Быкадорова, Е. А. Программирование. Практикум: учебное пособие / Е. А. Быкадорова. — Санкт-Петербург: Лань, 2020. — 60 с. — ISBN 978-5-8114-4612-4. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/139323 (дата обращения: 14.09.2021). — Режим доступа: для авториз. пользователей.
4. Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке С++: учебное пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; под ред. Л.Г. Гагариной. — Москва: ФОРУМ: ИНФРА-М, 2021. — 512 с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование). - ISBN 978-5-8199-0699-6. - Текст: электронный. - URL: https://znanium.com/catalog/product/1172261 (дата обращения: 14.09.2021). – Режим доступа: по подписке.
5. Ночка, Е. И. Основы алгоритмизации и программирования. Ответы на контрольные вопросы.: Учебник / Ночка Е.И. - Москва :КУРС, НИЦ ИНФРА-М, 2017. - 59 с.: ISBN 978-5-906818-82-9. - Текст : электронный. - URL: https://znanium.com/catalog/product/772548 (дата обращения: 28.08.2021). – Режим доступа: по подписке.
6. Основы алгоритмизации и программирования : учебное пособие / составитель Е. И. Николаев. — Ставрополь : СКФУ, 2015. — 211 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/155245 (дата обращения: 28.08.2021). — Режим доступа: для авториз. пользователей.
7. Ульянова, Н. Д. Основные принципы алгоритмизации : учебно-методическое пособие / Н. Д. Ульянова. — Брянск : Брянский ГАУ, 2020. — 56 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/172114 (дата обращения: 28.08.2021). — Режим доступа: для авториз. пользователей.

Руководитель библиотеки Л.В. Беляева

10. Материально-техническое и программное обеспечение дисциплины

Освоение дисциплины ОП.04 Основы алгоритмизации и программирования предполагает использование следующего материально-технического обеспечения:

Наименование дисциплины	Наименование кабинета, перечень оборудования
ОП.04 Основы алгоритмизации и программирования	Лаборатория «Программирования и баз данных» 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, в том числе включающее в себя следующее ПО: Office Professional Plus 2010, GIMP, Inkscape, Notepad ++, Python, Lazarus. Сервер в лаборатории (8-ми ядерный процессор Intel Core i7 , тактовая частота 3.8ГГц, оперативная память 16гб, жёсткий диск 1ТБ, программное обеспечение: Virtual box); Программное обеспечение: EclipseIDEforJavaEEDevelopers, NETFrameworkJDK 8, MicrosoftSQLServerExpressEdition, MySQLInstallerforWindows, NetBeans, SQLServerManagementStudio, MicrosoftSQLServerJavaConnector, AndroidStudio, MicrosoftVisioProfessional, MicrosoftVisualStudio, IntelliJIDEA

Учебно-методическая литература для данной дисциплины имеется в наличии в электронно-библиотечной системе "ZNANIUM.COM", доступ к которой предоставлен обучающимся. ЭБС "ZNANIUM.COM" содержит 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 соответствует всем требованиям федеральных государственных образовательных стандартов среднего профессионального образования нового поколения.
Учебно-методическая литература для данной дисциплины 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 по естественным, техническим и гуманитарным наукам. ЭБС Издательства "Лань" обеспечивает доступ к научной, учебной литературе и научным периодическим изданиям.

11. Методы обучения для обучающихся инвалидов и лиц с ограниченными возможностями здоровья

В образовательном процессе используются социально-активные и рефлексивные методы обучения, технологии социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими обучающимися, создании комфортного психологического климата в студенческой группе.
Условия обучения инвалидов и лиц с ограниченными возможностями здоровья:
- учебные аудитории, в которых проводятся занятия со студентами с нарушениями слуха, оборудованы мультимедийной системой (ПК и проектор), компьютерные тифлотехнологии базируются на комплексе аппаратных и программных средств, обеспечивающих преобразование компьютерной информации доступные для слабовидящих формы (укрупненный текст);
- в образовательном процессе используются социально-активные и рефлексивные методы обучения: кейс-метод, метод проектов, исследовательский метод**,** дискуссии в форме круглого стола**,** конференции, метод мозгового штурма.

Программа составлена в соответствии с требованиями ФГОС СПО по специальности 09.02.07 «Информационные системы и программирование».

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
«1» марта 2024 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

ОП.04 Основы алгоритмизации и программирования

09.02.07 Информационные системы и программирование

(код и наименование специальности)

Программист

(квалификация выпускника)

Елабуга, 2024

Паспорт
фонда оценочных средств по дисциплине
ОП.04 Основы алгоритмизации и программирования

Индекс компетенции	**Расшифровка компетенции**	**Показатель формирования компетенции для данной дисциплины**	**Оценочные средства**
1	2	3	4
ОК 01	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.	Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции Уметь разрабатывать алгоритмы для конкретных задач; определять сложность работы алгоритмов	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15
ОК 02	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности	Знать эволюцию языков программирования, их классификацию, понятие системы программирования Уметь оформлять код программы в соответствии со стандартом кодирования	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Практические задания к дифференцированному зачету 1-15 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15
ОК 04	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами	Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти Уметь реализовывать построенные алгоритмы в виде программ на конкретном языке программирования; определять сложность работы алгоритмов	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Практические задания к дифференцированному зачету 1-15 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15
ОК 05	Осуществлять устную и письменную коммуникацию на государственном	Знать понятие алгоритмизации, свойства алгоритмов, общие принципы	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73

	языке с учетом особенностей социального и культурного контекста	построения алгоритмов, основные алгоритмические конструкции Уметь реализовывать построенные алгоритмы в виде программ на конкретном языке программирования	Практические задания к дифференцированному зачету 1-15 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15
ОК 09	Использовать информационные технологии в профессиональной деятельности	Знать подпрограммы, составление библиотек подпрограмм Уметь работать в среде программирования	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Практические задания к дифференцированному зачету 1-15 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15
ОК 10	Пользоваться профессиональной документацией на государственном и иностранном языках	Знать объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения Уметь использовать программы для графического отображения алгоритмов	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Практические задания к дифференцированному зачету 1-15 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15
ПК 1.1	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции Уметь оформлять код программы в соответствии со стандартом кодирования	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Практические задания к дифференцированному зачету 1-15 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15

ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием	Знать подпрограммы, составление библиотек подпрограмм Уметь реализовывать построенные алгоритмы в виде программ на конкретном языке программирования	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Практические задания к дифференцированному зачету 1-15 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15
ПК 1.3	Выполнять отладку программных модулей с использованием специализированных программных средств	Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти Уметь выполнять проверку, отладку кода программы	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Практические задания к дифференцированному зачету 1-15 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15
ПК 1.4	Выполнять тестирование программных модулей	Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти Уметь выполнять проверку, отладку кода программы	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Практические задания к дифференцированному зачету 1-15 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15
ПК 1.5	Осуществлять рефакторинг и оптимизацию программного кода	Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти Уметь выполнять проверку, отладку кода программы	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Практические задания к дифференцированному зачету 1-15 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15
ПК 2.4	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения	Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти Уметь работать в среде программирования	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Практические задания к дифференцированному зачету 1-15 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15

ПК 2.5	Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования	Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти Уметь работать в среде программирования	Контрольные работы 1-2 Вопросы к дифференцированному зачету 1-73 Практические задания к дифференцированному зачету 1-15 Вопросы к экзамену 1-21 Практические задания к экзамену 1-15

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

Тема 1 Теоретические основы алгоритмизации и программирования

Комплект заданий для контрольной работы 1
ОП.4 «Основы алгоритмизации и программирования»
(ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ОК 10, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5)

Вариант 1

1 Чем характеризуется сложность алгоритма?
2. Чем различаются простые и интегрированные структуры данных?
3 Как получается верхняя оценка сложности алгоритма?
4. Перечислите основные типы связных списков.
5 Какие функции используются для представления верхней оценки сложности алгоритма?
6. Чем различаются стек, очередь и дек?
7 Как оценивается сложность экспериментальным методом?
8. Приведите примеры графовых структур.
9 Как влияет размер массива на временную сложность алгоритма?
10. Перечислите свойства алгоритма.
11 Как определяется емкостная сложность алгоритма?

Вариант 2

1. Что называется структурой данных?
2 Как оценивается асимптотическая сложность алгоритма?
3. Назовите основные особенности статических, полустатических и динамических структур.
4 Отличаются ли и на сколько асимптотическая и верхняя оценка сложности алгоритма?
5. Дайте определение линейных и нелинейных структур.
6 У каких известных вам алгоритмов сложность является константной, а у каких - линейной?
7. В чем заключается основная особенность древовидных структур.
8 Совпадают ли результаты экспериментальной и верхней оценок и, если нет, то на сколько они отличаются?
9. Что такое алгоритм?
11. Назовите основные типы моделей.
10 Как влияет количество циклов повторения исследуемого алгоритма на погрешность определения времени его выполнения?

Тема 2. Основы программирования. Программирование структур данных

Комплект заданий для контрольной работы 2
ОП.4 «Основы алгоритмизации и программирования»
(ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ОК 10, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5)

Вариант 1

1. В кортеже целых чисел вычислить произведение отрицательных элементов, имеющих нечетные индексы.
2. Разработайте программу, которая проверяет, является ли введенная с клавиатуры последовательность символов целым числом, записанным в двоичной системе счисления.

3. Список A содержит N чисел. Найдите количество элементов списка, значения которых превышают заданное значение znach. Назначение функции: подсчет количества элементов, превышающих заданное значение. Оформите созданную функцию в виде программного модуля и подключите его к основной программе.
4. Из строки, расположенной в текстовом файле, исключите все символы, входящие в нее более одного раза. Полученную строку сохраните в другом файле.

Вариант 2

1. В кортеже целых чисел вычислите среднее арифметическое значение квадратов положительных элементов
2. Разработайте программу, которая вычисляет среднюю длину слов во введенной с клавиатуры строке.
3. Одномерный массив A содержит N чисел. Найдите сумму значений элементов массива, меньших заданного значения Znach. Назначение функции: суммирование элементов, значения которых меньше заданного.
4. Проверить, правильно ли расставлены в тексте, расположенном в текстовом файле, круглые скобки.

Тема 3. Объектно-ориентированное программирование

Комплект заданий для контрольной работы 3
ОП.4 «Основы алгоритмизации и программирования»
Защита программного приложения (ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ОК 10, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5)

1. Разработайте тестовую GUI-программу на тему: Российские кинофильмы.
2. Разработайте тестовую GUI-программу на тему: Спорт.
3. Разработайте тестовую GUI-программу на тему: Автомобили.
4. Разработайте тестовую GUI-программу на тему: Российская эстрада.
5. Разработайте тестовую GUI-программу на тему: Программирование.
6. Разработайте тестовую GUI-программу на тему: Животные.
7. Разработайте тестовую GUI-программу на тему: Города России.
8. Разработайте тестовую GUI-программу на тему: Достопримечательности моего города.
9. Разработайте тестовую GUI-программу на тему: Кулинария.
10. Разработайте тестовую GUI-программу на тему: Мотоциклы.
11. Разработайте тестовую GUI-программу на тему: Живопись.
12. Разработайте тестовую GUI-программу на тему: Книги.
13. Разработайте тестовую GUI-программу на тему: Мультфильмы.
14. Разработайте тестовую GUI-программу на тему: Зарубежное кино.
15. Разработайте тестовую GUI-программу на тему: Зарубежная эстрада.
16. Разработайте тестовую GUI-программу на тему: Военная техника.
17. Разработайте тестовую GUI-программу на тему: Классическая музыка.
18. Разработайте тестовую GUI-программу на тему: Домашние животные.
19. Разработайте тестовую GUI-программу на тему: Информатика.
20. Разработайте тестовую GUI-программу на тему: Моря и океаны.
21. Разработайте тестовую GUI-программу на тему: Великие пустыни мира.
22. Разработайте тестовую GUI-программу на тему: Герои олимпиад.
23. Разработайте тестовую GUI-программу на тему: История России.
24. Разработайте тестовую GUI-программу на тему: Русские поэты.
25. Разработайте тестовую GUI-программу на тему: Дикие животные.
26. Разработайте тестовую GUI-программу на тему: Российские киноактеры.
27. Разработайте тестовую GUI-программу на тему: Интернет.

28. Разработайте тестовую GUI-программу на тему: Русские народные сказки.
29. Разработайте тестовую GUI-программу на тему: Легенды российского балета.
30. Разработайте тестовую GUI-программу на тему: Оргтехника.

Критерии оценки:

Компетенции	Планируемые результаты обучения	Критерии оценивания результатов обучения (баллы)			
		«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
		[критерии выставления оценки «неудовлетворительно»]	[критерии выставления оценки «удовлетворительно»]	[критерии выставления оценки «хорошо»]	[критерии выставления оценки «отлично»]
ОК 01	Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь разрабатывать алгоритмы для конкретных задач; определять сложность работы алгоритмов	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений
ОК 02	Знать эволюцию языков программирования, их классификацию, понятие системы программирования	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь оформлять код программы в соответствии со стандартом кодирования	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений
ОК 04	Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний

	Уметь реализовывать построенные алгоритмы в виде программ на конкретном языке программирования; определять сложность работы алгоритмов	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений
ОК 05	Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь реализовывать построенные алгоритмы в виде программ на конкретном языке программирования	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений
ОК 09	Знать подпрограммы, составление библиотек подпрограмм	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь работать в среде программирования	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений

ОК 10	Знать объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь использовать программы для графического отображения алгоритмов	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений
ПК 1.1	Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь оформлять код программы в соответствии со стандартом кодирования	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений
ПК 1.2	Знать подпрограммы, составление библиотек подпрограмм	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний

	Уметь реализовывать построенные алгоритмы в виде программ на конкретном языке программирования	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений
ПК 1.3	Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь выполнять проверку, отладку кода программы	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений
ПК 1.4	Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь выполнять проверку, отладку кода программы	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений
ПК 1.5	Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний

	Уметь выполнять проверку, отладку кода программы	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений
ПК 2.4	Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь работать в среде программирования	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений
ПК 2.5	Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь работать в среде программирования	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений

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

Вопросы к дифференцированному зачёту ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ОК 10, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5)

1. Расскажите о назначении IDLE. С какими способами создания программ в IDLE вы познакомились?
2. Какие операторы ввода и вывода данных используются для приложений, разрабатываемых на языке Python? Напишите синтаксис используемых операторов.
3. В каких случаях при разработке концепции глобальной обработки ошибок применяется конструкция try…except…finally? Поясните работу обработчиков исключений на примерах.
4. Назовите основные типы исключений и укажите причины их возникновения.
5. Какова роль отступов в программах, написанных на языке Python? 6. Какой алгоритм называется линейным?
7. Нарисуйте общий вид линейного алгоритма.
8. С какой целью используется оператор exit(0) в программах, написанных на языке Python?
1. Что называется разветвляющимся алгоритмом?
2. Как записывается простой условный оператор в блок-схемах?
3. Как записывается простой условный оператор в программах?
4. Как работает простой условный оператор?
5. Как записывается сокращенный условный оператор в блок-схемах?
6. Как записывается сокращенный условный оператор в программах?
7. Как работает сокращенный условный оператор?
8. Как записывается составной условный оператор в блок-схемах?
9. Как записывается составной условный оператор в программах?
10. Как работает составной условный оператор?
11. Как записываются многозначные ветвления в блок-схемах?
12. Как записываются многозначные ветвления в программах?
13. Как работает условный оператор if при проверке нескольких условий?
14. Поясните назначение метода format и приведите примеры его применения.
15. Расскажите, в каких случаях применяются циклы с неизвестным числом повторений.
16. Какая циклическая структура может считаться итеративной?
17. Нарисуйте общий вид алгоритма оператора цикла while.
18. Напишите синтаксис оператора цикла while.
19. Расскажите о работе оператора цикла while. Приведите примеры.
20. В каких случаях применяются рекуррентные соотношения? Расскажите об алгоритме вывода рекуррентной формулы.
21. В чем состоит главная особенность кортежей?
22. Каковы преимущества кортежей с точки зрения их использования в программах?
23. Напишите синтаксис объявления кортежей. 24. Каким образом осуществляется доступ к каждому элементу кортежа при его обработке?
25. Перечислите классические способы обработки кортежей.
26. Каким образом можно реализовать в программе срез кортежа?
27. Поясните, каким образом осуществляется обмен значений элементов кортежа.
28. Поясните, в чем состоит отличие списков, созданных на языке Python, от кортежей.
29. Напишите синтаксис объявления списков.
30. Какие возможности языка Python используются для генерации списков?
31. Перечислите и поясните основные методы работы со списками.
32. Дайте определение такой структуры данных языка Python, как словарь.
33. Напишите синтаксис создания словаря.

34. Какие правила следует использовать при создании словаря?
35. Как называется кодировка, поддерживающая кодирование буквенно-цифровых символов? Расскажите о ее структуре.
36. Перечислите основные функции для работы с символами. Приведите примеры.
37. Перечислите методы работы со строками, позволяющие преобразовывать символы строки к различным регистрам клавиатуры.
38. Какой метод позволяет разбить строку на подстроки? Напишите его синтаксис.
39. Какой метод отвечает за преобразование строки в список? Напишите его синтаксис.
40. Приведите примеры базовых алгоритмов строк.
41. Каким образом можно осуществить срез строки?
43. Нарисуйте общий вид квадратной матрицы. Какая диагональ называется главной, какая – побочной?
44. Каким образом можно обратиться к элементу вложенного списка?
45. Перечислите базовые алгоритмы обработки вложенных последовательностей.
46. Опишите словесный алгоритм нахождения количества элементов вложенной последовательности при некотором условии.
47. Опишите словесный алгоритм нахождения суммы элементов вложенной последовательности при некотором условии.
48. Опишите словесный алгоритм нахождения экстремальных значений вложенной последовательности при некотором условии.
49. Поясните, каким образом осуществляется обмен столбцов во вложенной последовательности.
50. Перечислите преимущества, которые получает программист в результате создания собственных функций.
51. Напишите синтаксис, в соответствии с которым создаются пользовательские функции.
52. Каким образом происходит вызов функции?
53. Как называются переменные, которые указываются в заголовке функции?
Объясните механизм действия функции на примере.
54. Поясните особенности работы с аргументами функции.
55. Каким образом можно передать в функцию произвольное количество параметров? Приведите примеры.
56. Приведите примеры того, как происходит вызов ранее написанной функции другой функцией.
57. Раскройте особенности модульного построения программ. Обоснуйте достоинства такого способа программирования.
58. Опишите словесный алгоритм создания модулей в языке Python.
59. Каким образом можно подключить созданный пользовательский модуль к программе, написанной на языке Python?
60. Из каких этапов, с точки зрения программирования, состоит работа с любым файлом?
61. Напишите синтаксис функции open(), предназначенной для открытия файла. Поясните назначение параметров функции.
62. Какие возможные значения режима Mode функции open() вы знаете?
63. Напишите синтаксис функции write(), предназначенной для записи информации в файл.
64. Напишите синтаксис функции close(), предназначенной для закрытия файла. 65. Каким образом осуществляется чтение информации из файла? Приведите пример.
66. Объясните, каким образом происходит обработка ошибок, возникающих при работе с файлами. Приведите пример.
67. Раскройте особенности методов read() и readline().
68. Дайте характеристику бинарных файлов.
69. Какова цель консервации данных, используемой в языке Python?
70. Прокомментируйте назначение модулей pickle и shelving.

71. Какая инструкция используется для записи информации в бинарный файл? Напишите ее синтаксис.
72. Какие возможные значения режима Mode функции open(), используемой при работе с бинарными файлами, вы знаете?
73. Каким образом осуществляется запись информации в бинарный файл?
Приведите пример программной конструкции.

Вопросы к экзамену (ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ОК 10, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5)

1. Назовите и поясните два основных аспекта объектно-ориентированного программирования.
2. Поля, методы, атрибуты – дайте характеристику.
3. Напишите синтаксис создания класса в языке Python.
4. Какой синтаксис используется при обращении к атрибуту класса?
5. Чем методы класса отличаются от обычных функций?
6. Поясните роль параметра self.
7. Какой синтаксис используется при обращении к методу класса?
8. С какой целью создается метод __init__? Напишите его синтаксис.
9. Объясните роль статических методов языка Python. Какие методы объявления статических методов вы знаете?
10. В чем заключается такой принцип ООП, как инкапсуляция?
11. Расскажите о методах создания закрытых атрибутов и способах доступа к ним.
12. С какой целью создаются свойства, и как происходит обращение к ним из клиентского кода?
13. Раскройте особенности одного из основных принципов ООП – наследования. Приведите синтаксис создания производного класса.
14. Что понимают под интерфейсом?
15. Какой синтаксис используется для доступа к свойствам и методам?
16. Опишите словесный алгоритм создания в Python базового окна при использовании библиотеки tkinter.
17. Какой метод необходимо использовать для получения данных из текстового окна?
18. Какой метод отвечает за вставку текста в текстовое окно, расположенное на форме?
19. Объясните основные принципы размещения виджетов на форме.
20. Какие классы используются в Python для обработки дробных или целых чисел, логических значений?
21. Какой параметр элемента управления позволяет выровнять его относительно сетки? Какие значения он может принимать?

Практические задания к экзамену (ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ОК 10, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5)

Пример заданий:

1. Создайте класс ФИГУРА с методами вычисления площади и периметра, а также методом, выводящим информацию о фигуре на экран. Создайте дочерние классы ПРЯМОУГОЛЬНИК, КРУГ, ТРЕУГОЛЬНИК со своими методами вычисления площади и периметра. Создайте список n фигур и выведите полную информацию о фигурах на экран.
2. Создайте класс ИЗДАНИЕ с методом, позволяющим вывести на экран информацию об издании, а также определить, является ли данное издание искомым. Создайте дочерние классы КНИГА (название, фамилия автора, год издания, издательство), СТАТЬЯ (название, фамилия автора, название журнала, его номер и год издания), ЭЛЕКТРОННЫЙ РЕСУРС (название, фамилия автора, ссылка, аннотация) со своими методами вывода информации на экран. Создайте список из n изданий, выведите полную информацию из списка, а также организуйте поиск изданий по фамилии автора.

3. Создайте класс ТРЕУГОЛЬНИК, заданный длинами двух сторон и угла между ними, с методами вычисления площади и периметра треугольника, а также методом, выводящим информацию о фигуре на экран. Создайте дочерние классы ПРЯМОУГОЛЬНЫЙ, РАВНОБЕДРЕННЫЙ, РАВНОСТОРОННИЙ со своими методами вычисления площади и периметра. Создайте список n треугольников и выведите полную информацию о треугольниках на экран.
4. Создайте класс ТРАНСПОРТ с методами, позволяющими вывести на экран информацию о транспортном средстве, а также определить грузоподъемность транспортного средства. Создайте дочерние классы АВТОМОБИЛЬ (марка, номер, скорость, грузоподъемность), МОТОЦИКЛ (марка, номер, скорость, грузоподъемность, наличие коляски, при этом если коляска отсутствует, то грузоподъемность равна нулю), ГРУЗОВИК (марка, номер, скорость, грузоподъемность, наличие прицепа, при этом если есть прицеп, то грузоподъемность увеличивается в два раза) со своими методами вывода информации на экран и определения грузоподъемности. Создайте список из n машин, выведите полную информацию на экран, а также организуйте поиск машин, удовлетворяющих требованиям грузоподъемности.
5. Создайте класс ТОВАР с методами, позволяющими вывести на экран информацию о товаре, а также определить, может ли приобрести товар покупатель, имеющий заданную сумму денег. Создайте дочерние классы ПРОДУКТ (название, цена, дата производства, срок годности), ПАРТИЯ (название, цена за штуку, количество штук, дата производства, срок годности), ТЕЛЕФОН (название, цена) со своими методами вывода информации на экран и определения соответствия заданной цене. Создайте список из n товаров, выведите полную информацию из базы на экран, а также организуйте поиск товара, который может приобрести покупатель, имеющий заданную сумму денег.
6. Создайте класс ТОВАР с методами, позволяющими вывести на экран информацию о товаре, а также определить, предназначен ли он для заданного возраста потребителя. Создайте дочерние классы ИГРУШКА (название, цена, производитель, материал, возраст, на который рассчитана), КНИГА (название, автор, цена, издательство, возраст, на который рассчитана), СПОРТИНВЕНТАРЬ (название, цена, производитель, возраст, на который рассчитан) со своими методами вывода информации на экран и определения соответствия возрасту потребителя. Создайте список из n товаров, выведите полную информацию из базы на экран, а также организуйте поиск товаров для потребителя в заданном возрастном диапазоне.
7. Создайте класс ТЕЛЕФОННЫЙ_СПРАВОЧНИК с методами, позволяющими вывести на экран информацию о записях в телефонном справочнике, а также определить соответствие записи критерию поиска. Создайте дочерние классы ПЕРСОНА (фамилия, адрес, номер телефона), ОРГАНИЗАЦИЯ (название, адрес, телефон, факс, контактное лицо), ДРУГ (фамилия, адрес, номер телефона, дата рождения) со своими методами вывода информации на экран и определения соответствия заданной фамилии. Создайте список из n записей, выведите полную информацию из базы на экран, а также организуйте поиск в базе по фамилии. 8. Создайте класс КЛИЕНТ с методами, позволяющими вывести на экран информацию о клиентах банка, а также определить соответствие клиента критерию поиска. Создайте дочерние классы ВКЛАДЧИК (фамилия, дата открытия вклада, размер вклада, процент по вкладу), КРЕДИТОР (фамилия, дата выдачи кредита, размер кредита, процент по кредиту, остаток долга), ОРГАНИЗАЦИЯ (название, дата открытия счета, номер счета, сумма на счету) со своими методами вывода информации на экран и определения соответствия дате (открытия вклада, выдаче кредита, открытия счета). Создайте список из n клиентов, выведите полную информацию из базы на экран, а также организуйте поиск клиентов, начавших сотрудничать с банком в заданную дату.
9. Создайте класс ПРОГРАММНОЕ_ОБЕСПЕЧЕНИЕ с методами, позволяющими вывести на экран информацию о программном обеспечении, а также определить соответствие возможности использования (на текущую дату). Создайте дочерние классы СВОБОДНОЕ (название, производитель), УСЛОВНО_БЕСПЛАТНОЕ (название, производитель, дата установки, срок бесплатного использования), КОММЕРЧЕСКОЕ (название, производитель,

цена, дата установки, срок использования) со своими методами вывода информации на экран и определения возможности использования на текущую дату. Создайте список из n видов программного обеспечения, выведите полную информацию из базы на экран, а также организуйте поиск программного обеспечения, которое допустимо использовать на текущую дату.
10. Создайте класс ТРАНСПОРТ с методами, позволяющими вывести на экран информацию о транспортном средстве, а также определить, находится ли транспортное средство в пределах заданных координат. Создайте дочерние классы САМОЛЕТ (марка, максимальные скорость и высота, количество пассажиров, координаты), АВТОМОБИЛЬ (марка, номер, год выпуска, координаты), КОРАБЛЬ (название, координаты, скорость, количество пассажиров, порт приписки) со своими методами вывода информации на экран и определения присутствия транспортного средства в пределах заданных координат. Создайте список из n транспортных средств, выведите полную информацию из базы на экран, а также организуйте поиск транспортных средств, которые сейчас находятся в пределах заданных координат.
11. Создайте класс ИГРУШКА с методами, позволяющими вывести на экран информацию о товаре, а также определить соответствие игрушки критерию поиска. Создайте дочерние классы КУБИК (цвет, цена, материал, размер ребра), МЯЧ (цена, цвет, диаметр, материал), АШИНКА (название, цена, производитель, цвет) со своими методами вывода информации на экран и определения соответствия заданному цвету. Создайте список из n игрушек, выведите полную информацию из базы на экран, а также организуйте поиск игрушек заданного цвета.
12. Создайте класс ТЕЛО с методами вычисления площади поверхности и объема, а также методом, выводящим информацию о фигуре на экран. Создайте дочерние классы ПАРАЛЛЕЛЕПИПЕД, ШАР, ПИРАМИДА со своими методами вычисления площади и объема. Создайте список n фигур и выведите полную информацию о фигурах на экран.
13. Создайте класс УРАВНЕНИЕ с методами вычисления корня уравнения и вывода результата на экран. Создайте дочерние классы ЛИНЕЙНОЕ, КВАДРАТНОЕ со своими методами вычисления корней и вывода на экран. Создайте список n уравнений и выведите полную информацию об уравнениях на экран.
14. Создайте класс ВАЛЮТА с методами перевода денежной суммы в рубли и вывода на экран. Создайте дочерние классы ДОЛЛАР, ЕВРО со своими методами перевода и вывода на экран. Создайте список n валютных денежных сумм и выведите полную информацию о них на экран.
15. Создайте класс ПРОГРЕССИЯ с методами вычисления j-го элемента прогрессии, ее суммы и методом, выводящим сумму на экран. Создайте дочерние классы: АРИФМЕТИЧЕСКАЯ, ГЕОМЕТРИЧЕСКАЯ со своими методами вычисления. Создайте список n прогрессий и выведите сумму каждой из них экран.

Критерии оценки на экзамене

«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.