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:
48505f1...

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.14 Основы объектно-ориентированного 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:

— теории объектно-ориентированного программирования;
— проблемам создания и сопровождения современных программных продуктов;
— использованию систем, созданных с использованием объектно-ориентированного программирования;

формирование умений по:

— осуществлению разработки кода программного модуля на основе объектно-ориентированного программирования;
— отладке и модификации прикладных программ, написанных в объектно-ориентированном стиле

2. Место дисциплины в структуре ППССЗ

Учебная дисциплина ОП.14 «Основы объектно-ориентированного программирования» является общепрофессиональной дисциплиной профессионального цикла в соответствии с ФГОС по специальности СПО 09.02.07 «Информационные системы и программирование».

Осваивается третьем и четвёртом курсах (6-7 семестр).

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	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 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
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием
ПК 2.4	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения

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

- теорию объектно-ориентированного программирования;
- проблемы создания и сопровождения современных программных продуктов;
- использовать системы, созданные с использованием объектно-ориентированного программирования.

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

- осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования;
- отладку и модификацию прикладных программ, написанных в объектно-

ориентированном стиле.

4. Структура и содержание дисциплины

4.1. Общая трудоемкость дисциплины в часах

Общая трудоемкость дисциплины составляет 153 часа.

Форма промежуточной аттестации по дисциплине: экзамен в 7 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	**Введение в объектно-ориентированное программирование**	6	17	0	17	1	1	Проверочная работа 1*
Тема 2	**Поддержка объектно-ориентированного программирования в Python**	6	17	0	34	2	1	Проверочная работа 2*
Тема 3	**Реализация объектно-ориентированного программирования в Python**	7	13	0	39	1	1	Защита проекта
Итого: 146			**47**	**0**	**90**	**4**	**3**	
Экзамен		**7**						
Всего по дисциплине		**151**						

* письменная Проверочная 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 Введение в объектно-ориентированное программирование	**Содержание учебного материала**	**36 (17/17/1)**	
	1. Проблемы процедурного подхода	5	*1*
	2. Основные понятия и терминология	6	*1*
	3. Принципы объектно-ориентированного программирования	6	*1*
	Лабораторные работы: 1.Проблемы процедурного подхода. 2.Основные понятия и терминология. 3.Принципы объектно-ориентированного программирования.	 5 6 6	 2 2 2
	Самостоятельная работа обучающегося Подготовка к проверочной работе по темам: Проблемы процедурного подхода. Основные понятия и терминология. Принципы объектно-ориентированного программирования. 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: 30.09.2021). – Режим доступа: по подписке	2	*3*
Тема 2. Поддержка объектно-ориентированного программирования в Python	**Содержание учебного материала**	**36 (17/34/2)**	
	1. Объектно-ориентированная модель.	8	*1*
	2. Инкапсуляция. Полиморфизм.	12	*1*
	3. Характеристики объектно-ориентированного программирования.	12	*1*
	Лабораторные работы 1. Объектно-ориентированная модель. 2. Инкапсуляция. Полиморфизм. 3. Характеристики объектно-ориентированного программирования.	 4 6 6	 2 2 2
	Самостоятельная работа обучающегося	2	*3*

	Подготовка к проверочной работе по темам: Объектно-ориентированная модель. Инкапсуляция. Полиморфизм. Характеристики объектно-ориентированного программирования. 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: 30.09.2021). – Режим доступа: по подписке		
Тема 3. Реализация объектно-ориентированного программирования в Python	**Содержание учебного материала**	**44 (13/39/1)**	
	1. Классы и объекты классов. Конструкторы и деструкторы.	3	*1*
	2. Наследование и полиморфизм.	3	*1*
	3. Знакомство с PyQT5. Базовая функциональность PyQT5.	4	*1*
	4. События, сигналы и слоты.	3	*1*
	Лабораторные работы 1. Классы и объекты классов. 2. Конструкторы и деструкторы. 3. Наследование и полиморфизм. 4. Знакомство с PyQT5. Базовая функциональность PyQT5. 5. События, сигналы и слоты.	 5 6 6 6 6	 2 2 2 2 2
	Самостоятельная работа обучающегося Защита проекта: Классы и объекты классов. Конструкторы и деструкторы. Наследование и полиморфизм. Знакомство с PyQT5. Базовая функциональность PyQT5. События, сигналы и слоты. 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: 30.09.2021). – Режим доступа: по подписке	2	*3*
Примерная тематика курсовой работы (проекта)		не предусмотрено	
Всего		**143 (47/90/4)**	

Консультация	**3**	
Экзамен	**7**	
Всего по дисциплине	**151**	

*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	Введение в объектно-ориентированное программирование	**Самостоятельная работа обучающегося** Подготовка к проверочной работе по темам: Проблемы процедурного подхода. Основные понятия и терминология. Принципы объектно-ориентированного программирования. 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: 30.09.2021). – Режим доступа: по подписке	1	Проверочная работа 1
Тема 2	Поддержка объектно-ориентированного программирования в Python	**Самостоятельная работа обучающегося** Подготовка к проверочной работе по темам: Объектно-ориентированная модель. Инкапсуляция. Полиморфизм. Характеристики объектно-ориентированного программирования. 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: 30.09.2021). – Режим доступа: по подписке	2	Проверочная работа 2*
Тема 3	Реализация объектно-ориентированного программирования в Python	**Самостоятельная работа обучающегося** Защита проекта: Классы и объекты классов. Конструкторы и деструкторы. Наследование и полиморфизм. Знакомство с PyQT5. Базовая функциональность PyQT5. События, сигналы и слоты. Гуриков, С. Р. Основы алгоритмизации и программирования на Python: учебное пособие / С.Р. Гуриков. — Москва:	1	Защита проекта

		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: 30.09.2021). – Режим доступа: по подписке		
Всего по дисциплине			**4**	

5. Образовательные технологии

Освоение дисциплины «Основы объектно-ориентированного 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;
- выполнение и защита проекта.

Занятия, проводимые в активной и интерактивной формах

Номер темы	Наименование темы	Форма проведения занятия	Объем в часах
Тема 1	Введение в объектно-ориентированное программирование	Информационно-проблемная лекция	17
Тема 2	Поддержка объектно-ориентированного программирования в Python	Информационно-проблемная лекция	17
Тема 3	Реализация объектно-ориентированного программирования в Python	Информационно-проблемная лекция	13
Всего по дисциплине			47

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

6.1 Оценочные средства для текущего контроля успеваемости

Проверочная работа 1 (ОК 01, ОК 02, ОК 04, ОК 05, ОК 08, ОК 09, ПК 1.2, ПК 2.4)

Пример заданий проверочной работы 1:

1. Назовите и поясните два основных аспекта объектно-ориентированного программирования.
2. Напишите синтаксис создания класса в языке Python.
3. Чем методы класса отличаются от обычных функций?
4. Какой синтаксис используется при обращении к методу класса?
5. Объясните роль статических методов языка Python. Какие методы объявления статических методов вы знаете?
6. Расскажите о методах создания закрытых атрибутов и способах доступа к ним.

Проверочная работа 2 (ОК 01, ОК 02, ОК 04, ОК 05, ОК 08, ОК 09, ПК 1.2, ПК 2.4)

Пример заданий проверочной работы 2:

Иерархия классов

Выберите вариант под номером (ЦИФРА_СБ % 5) + 1 (Таблица 10.5.2).

Таблица Иерархия классов

№ п/п	Классы	Методы базового класса
1	Плеер, АудиоПлеер, ВидеоПлеер, DvdПлеер	запустить(), остановить()
2	ПищущаяПринадлежность, Карандаш, Ручка, ГелеваяРучка	писать()
3	ТранспортноеСредство, ВодноеТС, КолесноеТС, Автомобиль	ехать()
4	ДенежныйПеревод, ПочтовыйПеревод, БанковскийПеревод, ВалютныйПеревод	выполнить()
5	ПроезднойБилет, БезлимитныйБилет, БилетСОграничением, БилетСОграничениемПоездок	списать_поездку()

Далее:

- выстройте классы в иерархию, продумайте их общие и отличительные характеристики и действия;
- добавьте собственную реализацию методов базового класса в каждый из классов, предусмотрев:
  - необходимые параметры для базовых методов (например, в метод воспроизведения в Dvd-плеере можно передать абстрактный DVD-диск);
  - необходимые поля для функционирования базовых методов (например, при остановке Dvd-плеера имеет смысл сохранить текущую позицию воспроизведения); классы должны содержать как минимум по одному общедоступному, не общедоступному и закрытому полю/методу;
  - вывод на экран работы метода (например, вызов метода остановки в Dvd-плеере должен сообщать на экране, что плеер установлен на определенной позиции).
- по желанию добавьте собственные методы в классы иерархии.

Реализуйте все классы в отдельном модуле, а также создайте main.py, который бы тестировал все его возможности.

По согласованию иерархия может быть расширена или выбрана самостоятельная индивидуальная тема для данной задачи.

Примеры тем проекта (ОК 01, ОК 02, ОК 04, ОК 05, ОК 08, ОК 09, ПК 1.2, ПК 2.4)

Требуется разработать средствами PyQT модель программного обеспечения. Некоторые вариант задания включают в себя разработку схемы базы данных.

Вариант 1. Модель программного обеспечения, управляющего работой цифрового диктофона.

Вариант 2. Модель программного обеспечения встроенного процессора универсального торгового автомата.

Вариант 3. Модель программного обеспечения табло для информационной службы метрополитена.

Вариант 4. Модель программной системы автоматизации работы пункта проката самокатов.

Вариант 5. Модель программного обеспечения встроенного микропроцессора учрежденческой мини-АТС (автоматической телефонной станции).

Вариант 6. Модель программного обеспечения встроенного микропроцессора для аппарата учрежденческой телефонной сети. Примечание: аппарат подключен к линии связи, ведущей к мини-АТС.

Вариант 7. Модель программного обеспечения встроенного микропроцессора стиральной машины.

Вариант 8. Модель встроенной системы управления работой таксофона городской телефонной сети.

6.2 Оценочные средства для промежуточной аттестации по итогам освоения дисциплины

Пример билета к экзамену (ОК 01, ОК 02, ОК 04, ОК 05, ОК 08, ОК 09, ПК 1.2, ПК 2.4)

1. Что такое объектно-ориентированное программирование?
2. Что такое перегрузка метода
3. Николай – оригинальный человек. Он решил создать класс Nikola, принимающий при инициализации 2 параметра: имя и возраст. Но на этом он не успокоился. Не важно, какое имя передаст пользователь при создании экземпляра, оно всегда будет содержать Николая. В частности - если пользователя на самом деле зовут Николаем, то с именем ничего не произойдет, а если его зовут, например, Максим, то оно преобразуется в Я не Максим, а Николай.

Более того, никаких других атрибутов и методов у экземпляра не может быть добавлено, даже если кто-то и вздумает так поступить (т.е. если некий пользователь решит прибавить к экземпляру свойство отчество или метод приветствие, то ничего у такого хитреца не получится).

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 теорию объектно-ориентированного программиров	Проверочные работы 1-2 Темы для защиты проекта 1-25	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний

	ания; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования	Вопросы к экзамену 1-30 Практические задания к экзамену 1-5		ошибок		
	Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	Проверочные работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5	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 теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы,	Проверочные работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний

	созданные с использованием объектно-ориентированного программирования					
	Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	Проверочные работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5	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 теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования	Проверочные работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку	Проверочные работы 1-2 Темы для	Не умеет, демонстрирует	Демонстрирует частичные	Умеет применять знания на	Демонстрирует высокий

	кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5	частичные умения, допуская грубые ошибки	умения без грубых ошибок	практике в базовом объёме	уровень умений
ОК 05	Знать теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования	Проверочные работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и	Проверочные работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5	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

	модификацию прикладных программ, написанных в объектно-ориентированном стиле					
ОК 08	Знать теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования	Проверочные работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	Проверочные работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5	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 теорию объектно-ориентирован	Проверочные работы 1-2 Темы для	Не знает, допускает грубые	Демонстрирует частичные	Знает достаточно в базовом	Демонстрирует высокий

	ного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования	защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5	ошибки	знания без грубых ошибок	объёме	уровень знаний
	Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	Проверочные работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5	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 теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов;	Проверочные работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний

	использовать системы, созданные с использовани ем объектно-ориентирован ного программиров ания					
	Уметь осуществлять разработку кода программного модуля на основе объектно-ориентирован ного программиров ания; отладку и модификацию прикладных программ, написанных в объектно-ориентирован ном стиле	Проверочны е работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практически е задания к экзамену 1-5	Не умеет, демонстриру ет частичные умения, допуская грубые ошибки	Демонстр ирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстр ирует высокий уровень умений
ПК 2.4	Знать теорию объектно-ориентирован ного программиров ания; проблемы создания и сопровожден ия современных программных продуктов; использовать системы, созданные с использовани ем объектно-ориентирован ного программиров ания	Проверочны е работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практически е задания к экзамену 1-5	Не знает, допускает грубые ошибки	Демонстр ирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстр ирует высокий уровень знаний
	Уметь	Проверочны	Не умеет,	Демонстр	Умеет	Демонстр

	осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	е работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5	демонстрирует частичные умения, допуская грубые ошибки	ирует частичные умения без грубых ошибок	применять знания на практике в базовом объёме	ирует 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. Жуков, Р. А. Язык программирования Python. Практикум : учебное пособие / Р. А. Жуков. — Москва : ИНФРА-М, 2021. — 216 с. — (Среднее профессиональное образование). - ISBN 978-5-16-015638-5. - Текст : электронный. - URL: https://znanium.com/read?id=365208 (дата обращения: 04.09.2021). – Режим доступа: по подписке.
2. Голицына, О. Л. Программирование на языке высокого уровня : 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: 28.08.2021). – Режим доступа: по подписке.
3. Гуриков, С. Р. Программирование на языке высокого уровня 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: 28.08.2021). – Режим доступа: по подписке.

9.2. Дополнительная литература:

1. 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.
2. Ночка, Е. И. Программирование на языке высокого уровня. 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.
3. Ахмедханлы, Д. М. Программирование на языке высокого уровня : 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.
4. Программирование на языке высокого уровня : 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.
5. 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.

Руководитель библиотеки Л.В. Беляева

10. Материально-техническое и программное обеспечение дисциплины

Освоение дисциплины ОП.014 «Основы объектно-ориентированного программирования» предполагает использование следующего материально-технического обеспечения:

Наименование дисциплины	Наименование кабинета, перечень оборудования
Основы объектно-ориентированного 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.14 Основы объектно-ориентированного программирования

09.02.07 Информационные системы и программирование
(код и наименование специальности)

Программист
(квалификация выпускника)

Елабуга, 2024

Паспорт
фонда оценочных средств по дисциплине
ОП.14 Основы объектно-ориентированного 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 теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	Проверочные работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5
ОК 02	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности	Знать теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования;	Проверочные работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5

		отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	
ОК 04	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами	Знать теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	Проверочные работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста	Знать теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	Проверочные работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5
ОК 08	Использовать средства физической культуры	Знать теорию объектно-ориентированного	Проверочные работы 1-2 Темы для защиты проекта 1-25

	для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	Вопросы к экзамену 1-30 Практические задания к экзамену 1-5
ОК 09	Использовать информационные технологии в профессиональной деятельности	Знать теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	Проверочные работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием	Знать теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы,	Проверочные работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5

		созданные с использованием объектно-ориентированного программирования Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	
ПК 2.4	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения	Знать теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	Проверочные работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5

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

Тема 1 Введение в объектно-ориентированное программирование

Комплект заданий для проверочной работы 1
ОП.14 «Основы объектно-ориентированного программирования»
(ОК 01, ОК 02, ОК 04, ОК 05, ОК 08, ОК 09, ПК 1.2, ПК 2.4)

Вариант 1

1. Назовите и поясните два основных аспекта объектно-ориентированного программирования.
2. Напишите синтаксис создания класса в языке Python.
3. Чем методы класса отличаются от обычных функций?
4. Какой синтаксис используется при обращении к методу класса?
5. Объясните роль статических методов языка Python. Какие методы объявления статических методов вы знаете?
6. Расскажите о методах создания закрытых атрибутов и способах доступа к ним.

Вариант 2

1. Поля, методы, атрибуты – дайте характеристику.
2. Какой синтаксис используется при обращении к атрибуту класса?
3. Поясните роль параметра self.
4. С какой целью создается метод __init__? Напишите его синтаксис.
5. В чем заключается такой принцип ООП, как инкапсуляция?
6. С какой целью создаются свойства, и как происходит обращение к ним из клиентского кода?

Тема 2. Поддержка объектно-ориентированного программирования в Python

Комплект заданий для проверочной работы 2
ОП.14 «Основы объектно-ориентированного программирования»
(ОК 01, ОК 02, ОК 04, ОК 05, ОК 08, ОК 09, ПК 1.2, ПК 2.4)
Вариант 1

Класс-контейнер

Создайте класс-контейнер, который будет содержать набор объектов из предыдущей задачи. Например, класс `VectorCollection` будет содержать объекты класса `Vector`.

Для класса-контейнера предусмотрите:

- специальные методы:
  - `__init__(self, ...)` - инициализация с необходимыми параметрами;
  - `__str__(self)` - представление объекта в удобном для человека виде;
  - `__getitem__()` - индексация и срез для класса-контейнера.
- поля, методы, свойства:
  - поле `_data` - содержит набор данных;
  - метод `add(self, value)` - добавляет элемент `value` в контейнер;
  - метод `remove(self, index)` - удаляет элемент из контейнера по индексу `index`;

- метод `save(self, filename)` - сохраняет объект в JSON-файл `filename`;
- метод `load(self, filename)` - загружает объект из JSON-файла `filename`.

Вариант 2

Иерархия классов

Выбе `(ЦИФРА_СБ % 5) + 1` (Таблица 10.5.2).

Таблица Иерархия классов		
№ п/п	**Классы**	**Методы базового класса**
1	Плеер, АудиоПлеер, ВидеоПлеер, DvdПлеер	запустить(), остановить()
2	ПищущаяПринадлежность, Карандаш, Ручка, ГелеваяРучка	писать()
3	ТранспортноеСредство, ВодноеТС, КолесноеТС, Автомобиль	ехать()
4	ДенежныйПеревод, ПочтовыйПеревод, БанковскийПеревод, ВалютныйПеревод	выполнить()
5	ПроездноеБилет, БезлимитныйБилет, БилетСОграничением, БилетСОграничениемПоездок	списать_поездку()

Далее:

- выстройте классы в иерархию, продумайте их общие и отличительные характеристики и действия;
- добавьте собственную реализацию методов базового класса в каждый из классов, предусмотрев:
  - необходимые параметры для базовых методов (например, в метод воспроизведения в Dvd-плеере можно передать абстрактный DVD-диск);
  - необходимые поля для функционирования базовых методов (например, при остановке Dvd-плеера имеет смысл сохранить текущую позицию воспроизведения); классы должны содержать как минимум по одному общедоступному, не общедоступному и закрытому полю/методу;
  - вывод на экран работы метода (например, вызов метода остановки в Dvd-плеере должен сообщать на экране, что плеер установлен на определенной позиции).
- по желанию добавьте собственные методы в классы иерархии.

Реализуйте все классы в отдельном модуле, а также создайте `main.py`, который бы тестировал все его возможности.

По согласованию иерархия может быть расширена или выбрана самостоятельная индивидуальная тема для данной задачи.

Тема 3. Реализация объектно-ориентированного программирования в Python

Темы проектов

ОП.14 «Основы объектно-ориентированного программирования»

Темы для защиты проекта (ОК 01, ОК 02, ОК 04, ОК 05, ОК 08, ОК 09, ПК 1.2, ПК 2.4)

Требуется разработать средствами PyQT модель программного обеспечения. Некоторые вариант задания включают в себя разработку схемы базы данных.

Вариант 1. Модель программного обеспечения, управляющего работой цифрового диктофона.

Вариант 2. Модель программного обеспечения встроенного процессора универсального торгового автомата.

Вариант задания включает в себя разработку схемы базы данных о товарах.

Вариант 3. Модель программного обеспечения табло для информационной службы метрополитена.

Вариант 4. Модель программной системы автоматизации работы пункта проката самокатов.

Вариант 5. Модель программного обеспечения встроенного микропроцессора учрежденческой мини-АТС (автоматической телефонной станции).

Вариант 6. Модель программного обеспечения встроенного микропроцессора для аппарата учрежденческой телефонной сети. Примечание: аппарат подключен к линии связи, ведущей к мини-АТС.

Вариант 7. Модель программного обеспечения встроенного микропроцессора стиральной машины.

Вариант 8. Модель встроенной системы управления работой таксофона городской телефонной сети.

Вариант 9. Модель программного обеспечения банкомата.

Вариант 10. Модель программного обеспечения встроенного процессора холодильника.

Вариант 11. Модель программного обеспечения встроенного микропроцессора для кодового замка, регулирующего доступ в помещение.

Вариант 12. Модель программного обеспечения встроенного процессора турникета для метрополитена.

Вариант 13. Модель системы поддержки заказа и учета товаров в бакалейной лавке. В ходе выполнения этого варианта задания должна быть разработана схема базы данных, хранящей информацию о товарах, заказах и поставщиках.

Вариант 14. Модель системы автоматизирующей деятельность библиотеки.

Вариант 15. Модель программного обеспечения Интернет-магазина.

Вариант 16. Модель программного обеспечения WWW-конференции. WWW-конференция представляет собой хранилище сообщений в сети Интернет.

Вариант 17. Модель программного обеспечения каталога ресурсов сети Интернет.

Вариант 18. Модель программного обеспечения встроенного микропроцессора для будильника.

Вариант 19. Модель системы для поддержки генеалогических деревьев. Система хранит сведения о персонах (Ф.И.О., пол, дата рождения, дата смерти, биография) и о родственных связях между ними.

Вариант 20. Модель встроенной системы управления работой телевизора.

Вариант 21. Модель системы поддержки составления расписания занятий.

При работе над этим вариантом задания необходимо разработать схему базы данных для хранения расписания.

Вариант 22. Модель программного обеспечения встроенного микропроцессора домофона.

Вариант 23. Модель программного обеспечения мастерской по ремонту телефонных аппаратов.

Вариант 24. Модель программного обеспечения автоматизированного рабочего места проектировщика.

Вариант 25. Модель программного обеспечения автоматизированного рабочего места менеджера по продаже программного обеспечения.

Критерии оценки:

Компетенции	Планируемые результаты обучения	Критерии оценивания результатов обучения (баллы)			
		«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
		[критерии выставления	[критерии выставления	[критерии выставления	[критерии выставления

		оценки «неудовлетворительно»]	оценки «удовлетворительно»]	оценки «хорошо»]	оценки «отлично»]
ОК 01	Знать теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	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 теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного	Не умеет, демонстрирует частичные умения,	Демонстрирует частичные умения без грубых	Умеет применять знания на практике в базовом	Демонстрирует высокий уровень умений

	программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	допуская грубые ошибки	ошибок	объёме	
ОК 04	Знать теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	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 теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний

	Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	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 теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	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 теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний

	использованием объектно-ориентированного программирования				
	Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	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 теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	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 теорию объектно-ориентированного программирования; проблемы создания и сопровождения	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний

	современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования				
	Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	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 теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	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 теорию	Не знает,	Демонстрир	Знает	Демонстрирует

	объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования	допускает грубые ошибки	ует частичные знания без грубых ошибок	достаточно в базовом объёме	высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	Не умеет, демонстриру ет частичные умения, допуская грубые ошибки	Демонстрир ует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений
ПК 1.5	Знать теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования	Не знает, допускает грубые ошибки	Демонстрир ует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных	Не умеет, демонстриру ет частичные умения, допуская грубые ошибки	Демонстрир ует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объёме	Демонстрирует высокий уровень умений

	в объектно-ориентированном стиле				
ПК 2.4	Знать теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	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 теорию объектно-ориентированного программирования; проблемы создания и сопровождения современных программных продуктов; использовать системы, созданные с использованием объектно-ориентированного программирования	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объёме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на основе объектно-ориентированного	Не умеет, демонстрирует частичные умения,	Демонстрирует частичные умения без грубых	Умеет применять знания на практике в базовом	Демонстрирует высокий уровень умений

	программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	допуская грубые ошибки	ошибок	объёме	

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

Вопросы к экзамену (ОК 01, ОК 02, ОК 04, ОК 05, ОК 08, ОК 09, ПК 1.2, ПК 2.4)

1. Что такое объектно-ориентированное программирование?
2. Зачем использовать объектно-ориентированное программирование?
3. Назовите основные принципы объектно-ориентированного программирования.
4. Что такое объект?
5. Что такое класс?
6. В чем разница между классом и структурой?
7. Можете ли вы вызвать метод базового класса, не создавая экземпляр?
8. В чем разница между классом и объектом?
9. Что такое наследование?
10. Какие бывают виды наследования?
11. В чем разница между множественным и многоуровневым наследованием?
12. Назовите ограничения в наследовании.
13. Что такое суперкласс?
14. Что такое подкласс?
15. Что такое полиморфизм?
16. Что такое перегрузка метода?
17. Что такое переопределение метода?
18. Что такое инкапсуляция?
19. Что такое абстракция данных?
20. Как добиться абстракции данных?
21. Что такое абстрактный класс?
22. Можете ли вы создать экземпляр абстрактного подкласса?
23. Что такое конструктор?
24. Что такое сборка мусора (GC)?
25. Что такое исключение?
26. Что из себя представляет обработка исключений?
27. Что такое блок try/except?
28. Что такое блок finally?
29. В чем разница между ошибкой и исключением?
30. Назовите ограничения объектно-ориентированного программирования.

Практические задания к экзамену (ОК 01, ОК 02, ОК 04, ОК 05, ОК 08, ОК 09, ПК 1.2, ПК 2.4)

Вариант 1

Создайте класс Soda (для определения типа газированной воды), принимающий 1 аргумент при инициализации (отвечающий за добавку к выбираемому лимонаду). В этом классе реализуйте метод show_my_drink(), выводящий на печать Газировка и {ДОБАВКА} в случае наличия добавки, а иначе отобразится следующая фраза: Обычная газировка.

Вариант 2

Николаю требуется проверить, возможно ли из представленных отрезков условной длины сформировать треугольник. Для этого он решил создать класс TriangleChecker, принимающий только положительные числа. С помощью метода is_triangle() возвращаются следующие значения (в зависимости от ситуации):

– Ура, можно построить треугольник!;
– С отрицательными числами ничего не выйдет!;
– Нужно вводить только числа!;

– Жаль, но из этого треугольник не сделать.

Вариант 3

Евгения создала класс KgToPounds с параметром kg, куда передается определенное количество килограмм, а с помощью метода to_pounds() они переводятся в фунты. Чтобы закрыть доступ к переменной kg она реализовала методы set_kg() - для задания нового значения килограммов, get_kg() - для вывода текущего значения кг. Из-за этого возникло неудобство: нам нужно теперь использовать эти 2 метода для задания и вывода значений. Помогите ей переделать класс с использованием функции property() и свойств-декораторов. Код приведен ниже.

Пример – IDE

```
class KgToPounds:

____def __init__(self, kg):
________self.__kg = kg

____def to_pounds(self):
________return self. _ kg * 2.205

____def set_kg(self, new_kg):
________if isinstance(new_kg, (int, float)):
____________self.__kg = new_kg
________else:
____________raise ValueError('Килограммы задаются только числами')

____def get_kg(self):
________return self.__kg
```

Вариант 4

Николай – оригинальный человек. Он решил создать класс Nikola, принимающий при инициализации 2 параметра: имя и возраст. Но на этом он не успокоился. Не важно, какое имя передаст пользователь при создании экземпляра, оно всегда будет содержать Николая. В частности - если пользователя на самом деле зовут Николаем, то с именем ничего не произойдет, а если его зовут, например, Максим, то оно преобразуется в Я не Максим, а Николай.

Более того, никаких других атрибутов и методов у экземпляра не может быть добавлено, даже если кто-то и вздумает так поступить (т.е. если некий пользователь решит прибавить к экземпляру свойство отчество или метод приветствие, то ничего у такого хитреца не получится).

Вариант 5

Строки в Питоне сравниваются на основании значений символов. Т.е. если мы захотим выяснить, что больше: Apple или Яблоко, – то Яблоко окажется бОльшим. А все потому, что английская буква A имеет значение 65 (берется из таблицы кодировки), а русская буква Я – 1071 (с помощью функции ord() это можно выяснить). Такое положение дел не устроило Анну. Она считает, что строки нужно сравнивать по количеству входящих в них символов.

Для этого девушка создала класс RealString и реализовала озвученный инструментарий. Сравнивать между собой можно как объекты класса, так и обычные строки с экземплярами класса RealString. К слову, Анне понадобилось только 3 метода внутри класса (включая __init__()) для воплощения задуманного.

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.