

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего
образования
"Казанский (Приволжский) федеральный университет"
Елабужский институт (филиал) КФУ



УТВЕРЖДАЮ

Заместитель директора по
образовательной деятельности
Елабужского института ЕИ КФУ


И.П. Михайлова
«01» марта 2024 г.
МП

Программа дисциплины (модуля)

ОП.14 Основы объектно-ориентированного программирования

Направление подготовки/специальность: 09.02.07 «Информационные системы и
программирование»

Квалификация выпускника: Программист

Форма обучения: очная

Язык обучения: русский

Год начала обучения по образовательной программе: 2022

г.Елабуга, 2024

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 ПССЗ

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 09	Использовать информационные технологии в профессиональной деятельности
ПК 1.1	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием

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	2	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	2	1	Защита проекта
Итого: 146			47	0	90	6	3	
Экзамен		7						
Всего по дисциплине		153						

* 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 (17/17/2)	
	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: 30.09.2023). – 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

	<p>Подготовка к проверочной работе по темам: Объектно-ориентированная модель. Инкапсуляция. Полиморфизм. Характеристики объектно-ориентированного программирования.</p> <p>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: 30.09.2023). – Режим доступа: по подписке</p>		
Тема 3. Реализация объектно-ориентированного программирования в Python	Содержание учебного материала	44 (13/39/2)	
	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.	5	2
	2. Конструкторы и деструкторы.	6	2
3. Наследование и полиморфизм.	6	2	
4. Знакомство с PyQt5. Базовая функциональность PyQt5.	6	2	
5. События, сигналы и слоты.	6	2	
Самостоятельная работа обучающегося	2	3	
<p>Защита проекта: Классы и объекты классов. Конструкторы и деструкторы. Наследование и полиморфизм. Знакомство с PyQt5. Базовая функциональность PyQt5. События, сигналы и слоты.</p> <p>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: 30.09.2023). – Режим доступа: по подписке</p>			
Примерная тематика курсовой работы (проекта)	не предусмотрен	о	
Всего	143 (47/90/6)		

Консультация	3	
Экзамен	7	
Всего по дисциплине	153	

*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	<p>Самостоятельная работа обучающегося</p> <p>Подготовка к проверочной работе по темам: Проблемы процедурного подхода. Основные понятия и терминология. Принципы объектно-ориентированного программирования.</p> <p>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: 30.09.2023). – Режим доступа: по подписке</p>	2	Проверочная работа 1
Тема 2	Поддержка объектно-ориентированного программирования в Python	<p>Самостоятельная работа обучающегося</p> <p>Подготовка к проверочной работе по темам: Объектно-ориентированная модель. Инкапсуляция. Полиморфизм. Характеристики объектно-ориентированного программирования.</p> <p>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: 30.09.2023). – Режим доступа: по подписке</p>	2	Проверочная работа 2*
Тема 3	Реализация объектно-ориентированного программирования в Python	<p>Самостоятельная работа обучающегося</p> <p>Защита проекта: Классы и объекты классов. Конструкторы и деструкторы. Наследование и полиморфизм. Знакомство с PyQt5. Базовая функциональность PyQt5. События, сигналы и слоты.</p> <p>Гуриков, С. Р. Основы алгоритмизации и программирования на Python: учебное пособие / С.Р. Гуриков. — Москва:</p>	2	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: 30.09.2023). – Режим доступа: по подписке		
Всего по дисциплине			6	

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

Фонд оценочных средств по дисциплине (модулю) включает оценочные материалы, направленные на проверку освоения компетенций, в том числе знаний, умений

и навыков. Фонд оценочных средств включает оценочные средства текущего контроля и оценочные средства промежуточной аттестации.

Фонд оценочных средств по дисциплине находится в Приложении 1 к программе дисциплины (модулю).

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 09	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципами отладки и тестирования программных продуктов; язык программирования Python	Проверочные работы 1-3 Вопросы к	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программы на языке программирования Python по разработанному алгоритму; выполнять отладку и	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений

	тестирование программ на языке программирования Python					
ПК 1.1	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов	Проверочные работы 1-3 Вопросы к экзамену 1-80 Практические варианты к экзамену 1-2	Не знает, допускает грубые ошибки	Демонстрирует частичные знания без грубых ошибок	Знает достаточно в базовом объеме	Демонстрирует высокий уровень знаний
	Уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку	Проверочные работы 1-3 Вопросы к экзамену 1-2 варианты к экзамену 1-2	Не умеет, демонстрирует частичные умения, допуская грубые ошибки	Демонстрирует частичные умения без грубых ошибок	Умеет применять знания на практике в базовом объеме	Демонстрирует высокий уровень умений

	<p>кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p>					
ПК 1.2	<p>Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного программирования; основным принципами отладки и тестирования программных продуктов; язык программирования Python</p>	<p>Проверочные работы 1-3 Вопросы к</p>	<p>Не знает, допускает грубые ошибки</p>	<p>Демонстрирует частичные знания без грубых ошибок</p>	<p>Знает достаточно в базовом объеме</p>	<p>Демонстрирует высокий уровень знаний</p>
	<p>Уметь осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программы на языке программирования Python по разработанно</p>	<p>Проверочные работы 1-3 Вопросы к экзамену 1-80</p>	<p>Не умеет, демонстрирует частичные умения, допуская грубые ошибки</p>	<p>Демонстрирует частичные умения без грубых ошибок</p>	<p>Умеет применять знания на практике в базовом объеме</p>	<p>Демонстрирует высокий уровень умений</p>

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 прохождения 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

Основная литература:

Залогова, Л. А. Основы объектно-ориентированного программирования на базе языка C# / Л. А. Залогова. — 4-е изд., стер. — Санкт-Петербург : Лань, 2023. — 192 с. — ISBN 978-5-507-48276-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/345992> (дата обращения: 02.11.2023). — Режим доступа: для авториз. пользователей.

Жуков, Р. А. Язык программирования Python. Практикум : учебное пособие / Р.А. Жуков. — Москва : ИНФРА-М, 2023. — 216 с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование). - ISBN 978-5-16-015638-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1916202> (дата обращения: 02.11.2023). – Режим доступа: по подписке.

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: 02.11.2023). – Режим доступа: по подписке.

Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2023. — 343 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-017142-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1913856> (дата обращения: 02.11.2023). – Режим доступа: по подписке.

Дополнительная литература:

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: 02.11.2023). — Режим доступа: для авториз. пользователей.

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: 02.11.2023). – Режим доступа: по подписке.

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: 02.11.2023). – Режим доступа: по подписке.

Шакин, В. Н. Объектно-ориентированное программирование на Visual Basic в среде Visual Studio.NET : учебное пособие / В.Н. Шакин, А.В. Загвоздкина, Г.К. Сосновиков. — Москва : ФОРУМ : ИНФРА-М, 2024. — 398 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-551-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2078382> (дата обращения: 02.11.2023). – Режим доступа: по подписке.

10. Материально-техническое и программное обеспечение дисциплины

Наименование дисциплины	Наименование кабинета, перечень оборудования
Основы объектно-ориентированного программирования	<p>Лаборатория «Программирования и баз данных»</p> <p>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.</p> <p>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</p>

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.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

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

УТВЕРЖДАЮ

Заместитель директора по
образовательной деятельности
Елабужского института ЕИ КФУ

 И. П. Михайлова
«01» марта 2024 г.

МП

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

ОП.14 Основы объектно-ориентированного программирования

(наименование дисциплины)

09.02.07 Информационные системы и программирование

(код и наименование специальности)

Программист

(квалификация выпускника)

г.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 09	Использовать информационные технологии профессиональной деятельности в	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 1.1	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	Знать: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов. Уметь: осуществлять разработку кода программного модуля на языках низ-кого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять	Проверочные работы 1-3 Вопросы к экзамену 1-30 Практические варианты к экзамену 1-2

		<p>отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства</p>	
ПК 1.2	<p>Разрабатывать программные модули в соответствии с техническим заданием</p>	<p>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 отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле</p>	<p>Проверочные работы 1-2 Темы для защиты проекта 1-25 Вопросы к экзамену 1-30 Практические задания к экзамену 1-5</p>

Тема 1 Введение в объектно-ориентированное программирование

Комплект заданий для проверочной работы 1 ОП.14 «Основы объектно-ориентированного программирования» (ОК 09, ПК 1.1, ПК 1.2)

Вариант 1

1. Назовите и поясните два основных аспекта объектно-ориентированного программирования. (ОК 09, ПК 1.1, ПК 1.2)

Ответ: Два основных аспекта объектно-ориентированного программирования — классы и объекты. Класс создаёт новый тип, а объекты являются экземплярами класса. Аналогично, когда мы говорим о "переменных типа int", это означает, что переменные, которые хранят целочисленные значения, являются экземплярами (объектами) класса int.

2. Чем методы класса отличаются от обычных функций? (ОК 09, ПК 1.1, ПК 1.2)

Ответ: Методы класса имеют одно отличие от обычных функций: они должны иметь дополнительно имя, добавляемое к началу списка параметров. Однако, при вызове метода никакого значения этому параметру присваивать не нужно — его укажет Python.

3. Какой синтаксис используется при обращении к методу класса? (ОК 09, ПК 1.1, ПК 1.2)

Ответ: Для обращения к методам применяется нотация точки - после имени объекта ставится точка и после нее идет вызов метода.

Вариант 2

1. Поля, методы, атрибуты – дайте характеристику. (ОК 09, ПК 1.1, ПК 1.2)

Ответ: Поле класса или атрибут в объектно-ориентированном программировании — переменная, связанная с классом или объектом. Все данные объекта хранятся в его полях. Доступ к полям осуществляется по их имени. Обычно тип данных каждого поля задаётся в описании класса, членом которого является поле.

2. Поясните роль параметра self. (ОК 09, ПК 1.1, ПК 1.2)

Ответ: В Python, self является параметром, который обычно используется в методах класса. Этот параметр представляет экземпляр класса, и он позволяет доступ к атрибутам и методам этого экземпляра. Почему Python использует self. В некоторых других объектно-ориентированных языках программирования, таких как Java или C++, не нужно явно указывать ссылку на экземпляр внутри методов класса. Это делается автоматически за вас.

3. В чем заключается такой принцип ООП, как инкапсуляция? (ОК 09, ПК 1.1, ПК 1.2)

Ответ: Инкапсуляция ООП позволяет устанавливать уровни доступа к данным и методам класса. Таким образом, некоторые данные и функциональность могут быть скрыты от других классов или пакетов, что способствует защите и контролю кода. Применение процесса инкапсуляции в практике программирования помогает создавать более структурированный, безопасный и расширяемый код.

Тема 2. Поддержка объектно-ориентированного программирования в Python

Комплект заданий для проверочной работы 2 ОП.14 «Основы объектно-ориентированного программирования» (ОК 09, ПК 1.1, ПК 1.2)

Вариант 1 (ОК 09, ПК 1.1, ПК 1.2)

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 (ОК 09, ПК 1.1, ПК 1.2)

Иерархия классов

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

№ п/п	Классы	Методы базового класса
1	Плеер, АудиоПлеер, ВидеоПлеер, DvdПлеер	запустить(), остановить()
2	ПишущаяПринадлежность, 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.

Тема 3. Реализация объектно-ориентированного программирования в Python

Темы проектов

ОП.14 «Основы объектно-ориентированного программирования»

Темы для защиты проекта (ОК 09, ПК 1.1, ПК 1.2)

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

Вариант 1. Модель программного обеспечения, управляющего работой цифрового диктофона. (ОК 09, ПК 1.1, ПК 1.2)

Вариант 2. Модель программного обеспечения встроенного процессора универсального торгового автомата. (ОК 09, ПК 1.1, ПК 1.2)

Вариант задания включает в себя разработку схемы базы данных о товарах. (ОК 09, ПК 1.1, ПК 1.2)

Вариант 3. Модель программного обеспечения табло для информационной службы метрополитена. (ОК 09, ПК 1.1, ПК 1.2)

Вариант 4. Модель программной системы автоматизации работы пункта проката самокатов. (ОК 09, ПК 1.1, ПК 1.2)

Вариант 5. Модель программного обеспечения встроенного микропроцессора учрежденческой мини-АТС (автоматической телефонной станции). (ОК 09, ПК 1.1, ПК 1.2)

Вариант 6. Модель программного обеспечения встроенного микропроцессора для аппарата учрежденческой телефонной сети. Примечание: аппарат подключен к линии связи, ведущей к мини-АТС. (ОК 09, ПК 1.1, ПК 1.2)

Вариант 7. Модель программного обеспечения встроенного микропроцессора стиральной машины. (ОК 09, ПК 1.1, ПК 1.2)

Вариант 8. Модель встроенной системы управления работой таксофона городской телефонной сети. (ОК 09, ПК 1.1, ПК 1.2)

Вариант 9. Модель программного обеспечения банкомата. (ОК 09, ПК 1.1, ПК 1.2)

Вариант 10. Модель программного обеспечения встроенного процессора холодильника. (ОК 09, ПК 1.1, ПК 1.2)

Тестовые задания ОП.14 «Основы объектно-ориентированного программирования»

1. В основе концепции объектно-ориентированного программирования лежит понятие:

- А) Объекта
- Б) Класса
- В) Инкапсуляции

Ответ: А (ОК 09, ПК 1.1, ПК 1.2)

2. Инкапсуляция – это:

А) Свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью.

Б) Сущность в адресном пространстве вычислительной системы, появляющаяся при создании экземпляра класса или копирования прототипа (например, после запуска результатов компиляции и связывания исходного кода на выполнение)

В) Свойство системы, позволяющее объединить данные и методы, работающие с ними в классе, и скрыть детали реализации от пользователя.

Ответ: В (ОК 09, ПК 1.1, ПК 1.2)

3. Способ выделить набор значимых характеристик объекта, исключая из рассмотрения незначимые.

- А) Полиморфизм
- Б) Абстрагирование
- В) Прототип

Ответ: Б (ОК 09, ПК 1.1, ПК 1.2)

4. Термин "наследование" обозначает, что...

- А) В производных классах присутствует часть состояния родительского класса
- Б) Производные классы содержат поля и методы родительского
- В) Производные классы наследуют модификаторы доступа членов родительского

класса

Ответ: Б (ОК 09, ПК 1.1, ПК 1.2)

5. Соотнесите понятия:

- А) Состояние объекта
- Б) Поведение объекта
- В) Значение атрибута объекта

- 1. некоторый объект или множество объектов
- 2. набор методов (программный код), оперирующих над состоянием объекта
- 3. набор значений его атрибутов

Ответ: 321 (ОК 09, ПК 1.1, ПК 1.2)

6. Соотнести понятия спецификаторов:

- А) private
- Б) protected
- В) public

- 1. защищенный, разрешено обращаться как с текущего класса так и с классов наследников.
- 2. закрытый, то есть к нему можно обращаться только в текущем классе
- 3. общедоступный, разрешено обращаться из любого места программы

Ответ: 213

7. В каких отношениях может находиться один класс с другим:

- А) Отношение наследования
- Б) Отношение включения
- В) Отношение использования

Ответ: АБВ (ОК 09, ПК 1.1, ПК 1.2)

8. Способ защититься от использования объектов одного класса вместо другого, или по крайней мере управлять таким использованием – это:

- А) Типизация
- Б) Наследование
- В) Полиморфизм

Ответ: А (ОК 09, ПК 1.1, ПК 1.2)

9. В каких случаях вызывается деструктор:

- А) создание объекта
- Б) удаление объекта
- В) редактирование объекта

Ответ: Б (ОК 09, ПК 1.1, ПК 1.2)

10. Соотнести определения:

- | | |
|----------------|---|
| А) Caption | 1. Определяет общий вид окна и операции с ним, которые разрешается выполнять пользователю |
| Б) Icon | 2. Определяет заголовок окна приложения. |
| В) BorderStyle | 3. Определяет значок, который будет использоваться в строке заголовка формы при выполнении программы, |

Ответ: 231 (ОК 09, ПК 1.1, ПК 1.2)

11. Под скрытием информации понимается?

- А) скрытие спецификаций от пользователей скрытие реализации от пользователей
- Б) механизм, делающий определённые компоненты недоступными для клиентов
- В) недокументированные возможности ПО

Ответ: Б (ОК 09, ПК 1.1, ПК 1.2)

12. Статическая типизация означает?

- А) после того, как сущность связана с объектом, другие объекты не могут присоединяться к сущности
- Б) связывание объекта и сущности выполняется еще на этапе трансляции и эта связь не может изменяться динамически в процессе выполнения приложения
- В) тип объекта, связываемого с сущностью, должен совпадать с типом сущности
- Г) для каждой сущности при ее объявлении задается тип

Ответ: ВГ (ОК 09, ПК 1.1, ПК 1.2)

13. Правило прямого отображения требует, чтобы

- А) модульная структура ПО непосредственно отображала структуру модели предметной области
- Б) модульная структура ПО непосредственно отображала структуру предметной области
- В) модульная структура ПО непосредственно отображала структуру спецификаций

Ответ: А (ОК 09, ПК 1.1, ПК 1.2)

14. К критериям модульности относятся

- А) непрерывность
- Б) единственный выбор
- В) унифицированный доступ
- Г) слабая связность интерфейсов

Ответ: А (ОК 09, ПК 1.1, ПК 1.2)

15. Роль универсальности в повторном использовании в том, что?

- А) взаимозависимости операций
- Б) позволяет справиться с проблемой изменчивости реализаций
- В) позволяет справиться с проблемой изменчивости типов
- Г) позволяет справиться с проблемой независимости представлений

Ответ: В (ОК 09, ПК 1.1, ПК 1.2)

16. Спецификация АД включает разделы

- А) функций
- Б) типов
- В) аксиом
- Г) все ответы верны

Ответ: Г (ОК 09, ПК 1.1, ПК 1.2)

17. Согласно принципу унифицированного доступа, клиент не может отличить
- А) вызов атрибута от вызова метода
 - Б) вызов атрибута от вызова функции
 - В) вызов атрибута от вызова функции без аргументов
 - Г) вызов функции от вызова процедуры
- Ответ: В** (ОК 09, ПК 1.1, ПК 1.2)
18. К какому понятию относятся высказывания: «статическое понятие, применимое к тексту, идентификатор, имеет значение в период выполнения»?
- А) класса
 - Б) сущности
 - В) объекта
 - Г) ссылки
- Ответ: Б** (ОК 09, ПК 1.1, ПК 1.2)
19. В инструкции множественного выбора `inspect e when v1 ... else ... end`
- А) может возникнуть исключение, когда ветвь `else` отсутствует и не найдено соответствие между `e` и проверяемыми значениями `v`
 - Б) выражение `e` может быть строкой символов
 - В) для каждой ветви указывается ровно одна выполняемая инструкция
- Ответ: А** (ОК 09, ПК 1.1, ПК 1.2)
20. В вызове `p(x)`
- А) `x` – может быть, как входным, так и выходным аргументом
 - Б) может измениться значение `x`, если оно является ссылкой
 - В) значение `x` не может измениться, если оно является ссылкой, но поля объекта, с которым связана ссылка, могут меняться
- Ответ: В** (ОК 09, ПК 1.1, ПК 1.2)

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 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 объеме	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 объеме	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 объеме	Демонстрирует высокий уровень знаний
	Уметь осуществлять разработку кода	Не умеет, демонстрирует	Демонстрирует	Умеет применять	Демонстрирует высокий

	программного модуля на основе объектно-ориентированного программирования; отладку и модификацию прикладных программ, написанных в объектно-ориентированном стиле	ирует частичные умения, допуская грубые ошибки	частичные умения без грубых ошибок	знания на практике в базовом объеме	уровень умений
ПК 1.2	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 объеме	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 объеме	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 экзамену (ОК 09, ПК 1.1, ПК 1.2)

1. Что такое объектно-ориентированное программирование?

Ответ: это одна из парадигм разработки, подразумевающая организацию программного кода, ориентируясь на данные и объекты, а не на функции и логические структуры. (ОК 09, ПК 1.1, ПК 1.2)

2. Зачем использовать объектно-ориентированное программирование?

Ответ: Используя объектное мышление, вы легко сможете проектировать сложные оперируя объектами и взаимодействием между ними. Т.е. ООП без объектного мышления не позволит вам начать использовать всю силу и мощь ООП. (ОК 09, ПК 1.1, ПК 1.2)

3. Назовите основные принципы объектно-ориентированного программирования.

Ответ: Абстракция. Моделирование требуемых атрибутов и взаимодействий сущностей в виде классов для определения абстрактного представления системы. Инкапсуляция. Скрытие внутреннего состояния и функций объекта и предоставление доступа только через открытый набор функций Наследование. Возможность создания новых абстракций, на основе существующих Полиморфизм. Возможность реализации наследуемых свойств или методов отличающимися способами в рамках множества абстракций (ОК 09, ПК 1.1, ПК 1.2)

4. Что такое класс?

Ответ: это логическая группа данных и функций. Он дает возможность создавать структуры данных, которые содержат произвольный контент и, следовательно, легко доступны. (ОК 09, ПК 1.1, ПК 1.2)

5. В чем разница между классом и структурой?

Ответ: Разница между классом и структурой заключается в том, что классы являются ссылочными типами, размещенными в куче и сборке мусора. (ОК 09, ПК 1.1, ПК 1.2)

6. В чем разница между классом и объектом?

Ответ: Объект в Python – это экземпляр класса, созданный на основе его шаблона. Объект имеет те же свойства и методы, что и класс, но может иметь свои собственные значения свойств. Например, если мы создали класс Person с свойствами name и age, то каждый созданный на его основе объект будет иметь свое имя и возраст, а также методы, определенные в классе Person. (ОК 09, ПК 1.1, ПК 1.2)

7. Какие бывают виды наследования?

Ответ: Родительский класс
Дочерний класс (ОК 09, ПК 1.1, ПК 1.2)

8. В чем разница между множественным и многоуровневым наследованием?

Ответ: Основное различие между множественным и многоуровневым наследованием состоит в том, что при многоуровневом наследовании суперкласс также может наследовать другой суперкласс. Таким образом, между классами могут быть созданы разные уровни наследования. (ОК 09, ПК 1.1, ПК 1.2)

9. Назовите ограничения в наследовании.

Ответ: это связано со способом реализации поиска метода / атрибута в python ("порядок разрешения метода" или MRO). Python использует механизм линеаризации суперкласса C3; это делается для решения того, что называется "Проблемой алмаза". После того, как вы ввели множественное наследование в свою иерархию классов, у любого данного класса нет ни одного потенциального класса, от которого он

наследуется, у него есть только "следующий класс в MRO", даже для классов, которые ожидают, что они наследуются от какого-то конкретного класса. (ОК 09, ПК 1.1, ПК 1.2)

10. Что такое суперкласс?

Ответ: это встроенная функция языка Python. Она возвращает прокси-объект, который делегирует вызовы методов классу-родителю (или собрату) текущего класса (или класса на выбор, если он указан, как параметр) (ОК 09, ПК 1.1, ПК 1.2)

11. Что такое подкласс?

Ответ: Дочерние классы, или подклассы – это классы, которые наследуют параметры родительских классов. Каждый дочерний класс сможет использовать методы и переменные родительского класса. (ОК 09, ПК 1.1, ПК 1.2)

12. Что такое полиморфизм?

Ответ: Полиморфизм подразумевает возможность использовать один и тот же интерфейс для различных базовых элементов (таких как типы данных или классы). При этом функции могут использовать объекты разных классов. (ОК 09, ПК 1.1, ПК 1.2)

13. Что такое перегрузка метода?

Ответ: возможность использования одинаковых имен методов, но с разными параметрами внутри одного класса (ОК 09, ПК 1.1, ПК 1.2)

14. Что такое переопределение метода?

Ответ: Переопределение метода - это способность любого объектно-ориентированного языка программирования, которая позволяет подклассу или дочернему классу предоставлять конкретную реализацию метода, который уже предоставлен одним из его суперклассов или родительских классов. (ОК 09, ПК 1.1, ПК 1.2)

15. Как добиться абстракции данных?

Ответ: Чтобы создать абстрактный метод и абстрактные классы, мы должны импортировать классы "ABC" и "abstractmethod" из библиотеки abc (абстрактный базовый класс). (ОК 09, ПК 1.1, ПК 1.2)

16. Что такое абстрактный класс?

Ответ: Абстрактным называется объявленный, но не реализованный метод. Абстрактные классы не могут быть инстанцированы, от них нужно унаследовать, реализовать все их абстрактные методы и только тогда можно создать экземпляр такого класса. (ОК 09, ПК 1.1, ПК 1.2)

17. Можете ли вы создать экземпляр абстрактного подкласса?

Ответ: Создать экземпляр абстрактного класса нельзя, его надо наследовать и уже у дочернего класса можно создать экземпляр. При этом экземпляр дочернего класса можно создать только в том случае, если у дочернего класса есть реализация всех абстрактных методов. (ОК 09, ПК 1.1, ПК 1.2)

18. Что такое конструктор?

Ответ: это специальный метод, который инициализирует объект класса. Это метод, который вызывается при создании нового экземпляра класса. (ОК 09, ПК 1.1, ПК 1.2)

19. Что такое сборка мусора (GC)?

Ответ: Как только объекты больше не нужны, Python автоматически освобождает память из-под них. Несмотря на это, понимание как работает GC поможет писать более качественный код. (ОК 09, ПК 1.1, ПК 1.2)

20. Что такое исключение?

Ответ: Исключительные ситуации или исключения (exceptions) – это ошибки, обнаруженные при исполнении. Или если файл был случайно удален пока программа работала? Такие ситуации обрабатываются при помощи исключений.

(ОК 09, ПК 1.1, ПК 1.2)

21.

21. Что из себя представляет обработка исключений? (ОК 09, ПК 1.1, ПК 1.2)

Ответ: Обработка исключений — это процесс написания кода для перехвата и обработки ошибок или исключений, которые могут возникать при выполнении программы

22. Что такое блок try/except? (ОК 09, ПК 1.1, ПК 1.2)

Ответ: Конструкция try/except используется для перехвата исключений в коде и исполнения альтернативного блока кода в случае возникновения исключения.

23. Что такое блок finally? (ОК 09, ПК 1.1, ПК 1.2)

Ответ: Блок finally - это еще один опциональный блок в конструкции try. Он выполняется всегда, независимо от того, было ли исключение или нет.

24. В чем разница между ошибкой и исключением? (ОК 09, ПК 1.1, ПК 1.2)

Ответ:

25. Назовите ограничения объектно-ориентированного программирования. (ОК 09, ПК 1.1, ПК 1.2)

Ответ: Ограничения классов состоит в том, чтобы контролировать атрибуты и методы родительских классов и подклассов.

Ограничение атрибутов подкласса (slots)

Ограничение на методе суб –класса

26. Что такое объект? (ОК 09, ПК 1.1, ПК 1.2)

Ответ: это экземпляр класса, созданный на основе его шаблона. Объект имеет те же свойства и методы, что и класс, но может иметь свои собственные значения свойств

27. Что такое наследование? (ОК 09, ПК 1.1, ПК 1.2)

Ответ: механизм, который позволяет создавать классы (подклассы) на основе других классов — базовые или суперклассы.

Практические задания к экзамену (ОК 09, ПК 1.1, ПК 1.2)

Вариант 1 (ОК 09, ПК 1.1, ПК 1.2)

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 (ОК 09, ПК 1.1, ПК 1.2)

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

(ОК 09, ПК 1.1, ПК 1.2)

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
```

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.