

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

УТВЕРЖДАЮ

Заведующий кафедрой
цифровых технологий



/ Кургалин С.Д.

22.04.2024 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

**Б1.В.ДВ.04.01 МЕТОДЫ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО
ПРОЕКТИРОВАНИЯ НА ЯЗЫКЕ PYTHON**

1. Код и наименование направления подготовки/специальности:

02.04.01 Математика и компьютерные науки

2. Профиль подготовки/специализация:

компьютерные науки и информационные технологии для цифровой экономики

3. Квалификация выпускника: магистр

4. Форма обучения: очная

5. Кафедра, отвечающая за реализацию дисциплины: цифровых технологий

6. Составители программы:

Романов Александр Викторович, старший преподаватель

7. Рекомендована: НМС ФКН (протокол № 5 от 05.03.2024)

8. Учебный год: 2025-2026 Семестр: 4

9. Цели и задачи учебной дисциплины

Цель изучения дисциплины: дать представление об основных возможностях языка программирования Python в проектировании объектно-ориентированных программ.

Задачи дисциплины:

- рассмотреть основные типы и средств объектно-ориентированного проектирования на языке Python;
- рассмотреть вопрос документирования и сопровождения объектно-ориентированного программного кода, написанного на языке Python;
- изучить вопросы модульного построения программ;
- изучить объектно-ориентированную модель функционирования языка программирования Python.

10. Место учебной дисциплины в структуре ООП

Дисциплина относится к блоку Б1 учебного плана (часть, формируемая участниками образовательных отношений, дисциплины по выбору).

11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями) и индикаторами их достижения:

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ПК-2	Способен создавать и исследовать новые математические модели в естественных науках, промышленности и бизнесе, с учетом возможностей современных информационных технологий, программирования и компьютерной техники.	ПК-2.1	Знает основные методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов, их сопровождения, администрирования и развития (эволюции).	Знать: методы и средства языка программирования Python, основы объектной модели языка, документирования и сопровождения программного кода.
ПК-2	Способен создавать и исследовать новые математические модели в естественных науках, промышленности и бизнесе, с учетом возможностей современных информационных технологий, программирования и компьютерной техники.	ПК-2.2	Умеет использовать методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного продукта.	Уметь: использовать средства программирования языка Python с учетом модульного построения программ и объектной модели языка.
ПК-2	Способен создавать и исследовать новые математические модели в естественных науках, промышленности и бизнесе, с учетом возможностей современных информационных технологий,	ПК-2.3	Имеет практический опыт применения указанных выше методов и технологий.	Владеть: практическими навыками решения вычислительных задач с помощью языка программирования Python.

	программирования и компьютерной техники.			
ПК-3	Способен использовать современные методы разработки и реализации конкретных алгоритмов математических моделей на базе языков программирования и пакетов прикладных программ моделирования.	ПК-3.1	Владеет современными методами разработки и реализации алгоритмов математических моделей на базе языков и пакетов прикладных программ моделирования.	Знать: методами разработки и реализации алгоритмов математических моделей на базе объектно-ориентированной модели языка программирования Python.
ПК-3	Способен использовать современные методы разработки и реализации конкретных алгоритмов математических моделей на базе языков программирования и пакетов прикладных программ моделирования.	ПК-3.2	Умеет разрабатывать и реализовывать алгоритмы математических моделей на базе языков и пакетов прикладных программ моделирования.	Уметь: разрабатывать и реализовывать алгоритмы математических моделей на базе объектно-ориентированной модели языка программирования Python.
ПК-3	Способен использовать современные методы разработки и реализации конкретных алгоритмов математических моделей на базе языков программирования и пакетов прикладных программ моделирования.	ПК-3.3	Имеет практический опыт разработки и реализации алгоритмов на базе языков и пакетов прикладных программ моделирования.	Владеть: практическими навыками разработки и реализации алгоритмов на базе объектно-ориентированной модели языка программирования Python.

12. Объем дисциплины в зачетных единицах/час. — 4/144.

Форма промежуточной аттестации: 4 семестр – зачет с оценкой.

13. Трудоемкость по видам учебной работы

Вид учебной работы	Трудоемкость	
	Всего	По семестрам
		4 сем.
Аудиторные занятия	56	56
в том числе:	лекции	28
	практические	
	лабораторные	28
Самостоятельная работа	88	88
Зачёт с оценкой		
Итого:	144	144

13.1. Содержание дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК*
1. Лекции			
1.1	Объектно-ориентированное программирование: общая картина	Наследование и композиция. Множество экземпляров. Настройка через наследование. Перегрузка операций. Поиск в иерархии наследования. Классы и экземпляры. Вызовы методов. Создание деревьев классов. Полиморфизм и классы.	https://edu.vsu.ru/course/view.php?id=15567
1.2	Основы написания классов	Создание экземпляров. Написание кода конструкторов. Добавление методов, реализующих поведение. Написание кода методов. Реализация отображения. Настройка поведения за счет создания подклассов. Наследование, настройка и расширение. Настройка конструкторов. Использование инструментов интроспекции. Специальные атрибуты класса. Атрибуты экземпляра или атрибуты класса. Сохранение объектов в базе данных.	https://edu.vsu.ru/course/view.php?id=15567
1.3	Детали реализации классов	Оператор class. Вызов конструкторов суперклассов. Построение дерева атрибутов. Специализации унаследованных методов. Методики связывания классов. Абстрактные суперклассы. Имена атрибутов: пространства имен объектов. Вложенные классы. Словари пространств имен.	https://edu.vsu.ru/course/view.php?id=15567
1.4	Перегрузка операций	Конструкторы и выражения: <code>__init__</code> и <code>__sub__</code> . Распространенные методы перегрузки операций. Индексирование и нарезание: <code>__getitem__</code> и <code>__setitem__</code> . Перехват срезов. Итерируемые объекты: <code>__iter__</code> и <code>__next__</code> . Множество итераторов в одном объекте.	https://edu.vsu.ru/course/view.php?id=15567
1.5	Проектирование с использованием классов	Python и объектно-ориентированное программирование. Объектно-ориентированное программирование и делегирование: промежуточные объекты-оболочки. Псевдозакрытые атрибуты классов. Множественное наследование: "подмешиваемые" классы.	https://edu.vsu.ru/course/view.php?id=15567
1.6	Расширенные возможности классов	Расширение встроенных типов. Расширение типов путем внедрения. Расширение типов путем создания подклассов. Модель классов "нового стиля". Расширения в классах нового стиля. Метод <code>__getattr__</code> и дескрипторы: инструменты для работы с атрибутами. Статические методы и методы классов	https://edu.vsu.ru/course/view.php?id=15567
1.7	Основы исключений	Роли, исполняемые исключениями. Стандартный обработчик исключений. Перехват исключений. Генерация исключений. Исключения, определяемые пользователем.	https://edu.vsu.ru/course/view.php?id=15567
1.8	Детали обработки исключений	Оператор <code>try/except/finally</code> . Оператор <code>raise</code> . Оператор <code>assert</code> . Диспетчеры контекстов <code>with/as</code> .	https://edu.vsu.ru/course/view.php?id=15567
1.9	Объекты исключений	Исключения на основе классов. Встроенные классы исключений. Категории встроенных исключений. Специальные данные и поведение.	https://edu.vsu.ru/course/view.php?id=15567

1.10	Проектирование с использованием исключений	Вложение обработчиков исключений. Идиомы исключений. Отладка с помощью внешних операторов try. Выполнение внутривещественных тестов. Отображение сообщений об ошибках и трассировок.	https://edu.vsu.ru/course/view.php?id=15567
1.11	Декораторы	Использование и определение декораторов. Декораторы функций. Декораторы классов. Вложение декораторов. Аргументы декораторов. Добавление аргументов к декоратору.	https://edu.vsu.ru/course/view.php?id=15567
1.12	Метаклассы	Модель метаклассов. Протокол оператора class. Объявление метаклассов. Реализация метаклассов. Методы метаклассов. Перегрузка операций в методах метакласса.	https://edu.vsu.ru/course/view.php?id=15567
2. Лабораторные занятия			
2.1	Объектно-ориентированное программирование: общая картина	Наследование и композиция. Множество экземпляров. Настройка через наследование. Перегрузка операций. Поиск в иерархии наследования. Классы и экземпляры. Вызовы методов. Создание деревьев классов. Полиморфизм и классы.	https://edu.vsu.ru/course/view.php?id=15567
2.2	Основы написания классов	Создание экземпляров. Написание кода конструкторов. Добавление методов, реализующих поведение. Написание кода методов. Реализация отображения. Настройка поведения за счет создания подклассов. Наследование, настройка и расширение. Настройка конструкторов. Использование инструментов интроспекции. Специальные атрибуты класса. Атрибуты экземпляра или атрибуты класса. Сохранение объектов в базе данных.	https://edu.vsu.ru/course/view.php?id=15567
2.3	Детали реализации классов	Оператор class. Вызов конструкторов суперклассов. Построение дерева атрибутов. Специализации унаследованных методов. Методики связывания классов. Абстрактные суперклассы. Имена атрибутов: пространства имен объектов. Вложенные классы. Словари пространств имен.	https://edu.vsu.ru/course/view.php?id=15567
2.4	Перегрузка операций	Конструкторы и выражения: <code>__init__</code> и <code>__sub__</code> . Распространенные методы перегрузки операций. Индексирование и нарезание: <code>__getitem__</code> и <code>__setitem__</code> . Перехват срезов. Итерируемые объекты: <code>__iter__</code> и <code>__next__</code> . Множество итераторов в одном объекте.	https://edu.vsu.ru/course/view.php?id=15567
2.5	Проектирование с использованием классов	Python и объектно-ориентированное программирование. Объектно-ориентированное программирование и делегирование: промежуточные объекты-оболочки. Псевдозакрытые атрибуты классов. Множественное наследование: "подмешиваемые" классы.	https://edu.vsu.ru/course/view.php?id=15567
2.6	Расширенные возможности классов	Расширение встроенных типов. Расширение типов путем внедрения. Расширение типов путем создания подклассов. Модель классов "нового стиля". Расширения в классах нового стиля. Метод <code>__getattr__</code> и дескрипторы: инструменты для работы с атрибутами. Статические методы и методы классов	https://edu.vsu.ru/course/view.php?id=15567
2.7	Основы исключений	Роли, исполняемые исключениями. Стандартный обработчик исключений. Перехват исключений. Генерация исключений. Исключения, определяемые пользователем.	https://edu.vsu.ru/course/view.php?id=15567

2.8	Детали обработки исключений	Оператор try/except/finally. Оператор raise. Оператор assert. Диспетчеры контекстов with/as.	https://edu.vsu.ru/course/view.php?id=15567
2.9	Объекты исключений	Исключения на основе классов. Встроенные классы исключений. Категории встроенных исключений. Специальные данные и поведение.	https://edu.vsu.ru/course/view.php?id=15567
2.10	Проектирование с использованием исключений	Вложение обработчиков исключений. Идиомы исключений. Отладка с помощью внешних операторов try. Выполнение внутривещественных тестов. Отображение сообщений об ошибках и трассировок.	https://edu.vsu.ru/course/view.php?id=15567
2.11	Декораторы	Использование и определение декораторов. Декораторы функций. Декораторы классов. Вложение декораторов. Аргументы декораторов. Добавление аргументов к декоратору.	https://edu.vsu.ru/course/view.php?id=15567
2.12	Метаклассы	Модель метаклассов. Протокол оператора class. Объявление метаклассов. Реализация метаклассов. Методы метаклассов. Перегрузка операций в методах метакласса.	https://edu.vsu.ru/course/view.php?id=15567

13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (количество часов)				Всего
		Лекции	Практические	Лабораторные	Самостоятельная работа	
1	Объектно-ориентированное программирование: общая картина	2		1	2	5
2	Основы написания классов	2		2	6	10
3	Детали реализации классов	2		2	6	10
4	Перегрузка операций	2		2	12	16
5	Проектирование с использованием классов	2		4	14	20
6	Расширенные возможности классов	2		2	12	16
7	Основы исключений	2		2	6	10
8	Детали обработки исключений	2		2	6	10
9	Объекты исключений	2		2	6	10
10	Проектирование с использованием исключений	4		3	6	13
11	Декораторы	4		4	8	16
12	Метаклассы	2		2	4	8
	Итого:	28		28	88	144

14. Методические указания для обучающихся по освоению дисциплины

Освоение дисциплины складывается из аудиторной работы (учебной деятельности, выполняемой под руководством преподавателя) и внеаудиторной работы (учебной деятельности, реализуемой обучающимся самостоятельно).

Аудиторная работа состоит из работы на лекциях и выполнения практических (или лабораторных) заданий в объеме, предусмотренном учебным планом. Лекция представляет собой последовательное и систематическое изложение учебного материала, направленное на знакомство обучающихся с основными понятиями и теоретическими положениями изучаемой дисциплины. Лекционные занятия формируют

базу для практических (или лабораторных) занятий, на которых полученные теоретические знания применяются для решения конкретных практических задач. Обучающимся для успешного освоения дисциплины рекомендуется вести конспект лекций и практических (лабораторных) занятий.

Самостоятельная работа предполагает углублённое изучение отдельных разделов дисциплины с использованием литературы, рекомендованной преподавателем, а также конспектов лекций, презентационным материалом (при наличии) и конспектов практических (лабораторных) занятий. В качестве плана для самостоятельной работы может быть использован раздел 13.1 настоящей рабочей программы, в котором зафиксированы разделы дисциплины и их содержание. В разделе 13.2 рабочей программы определяется количество часов, отводимое на самостоятельную работу по каждому разделу дисциплины. Больше количество часов на самостоятельную работу отводится на наиболее трудные разделы дисциплины. Для самостоятельного изучения отдельных разделов дисциплины используется перечень литературы и других ресурсов, перечисленных в пунктах 15 и 16 настоящей рабочей программы.

Успешность освоения дисциплины определяется систематичностью и глубиной аудиторной и внеаудиторной работы обучающегося.

При использовании дистанционных образовательных технологий и электронного обучения выполнять все указания преподавателей, вовремя подключаться к online занятиям, ответственно подходить к заданиям для самостоятельной работы.

15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

а) основная литература:

№ п/п	Источник
1	Саммерфилд, М. Python на практике : учебное пособие / Саммерфилд М. – Москва : ДМК-пресс, 2014. – 338 с. – https://www.studentlibrary.ru/book/ISBN9785970600955.html
2	Шелудько, В.М. Основы программирования на языке высокого уровня Python : учебное пособие / Шелудько В.М. – Москва : ЮФУ, 2017. – 146 с. – https://www.studentlibrary.ru/book/ISBN9785927526499.html
3	Шелудько, В.М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули : учебное пособие / Шелудько В.М. – Москва : ЮФУ, 2017. – 107 с. – https://www.studentlibrary.ru/book/ISBN9785927526482.html

б) дополнительная литература:

№ п/п	Источник
1	Златопольский, Д.М. Основы программирования на языке Python : учебник / Златопольский Д.М. – Москва : ДМК-пресс, 2017. – 284 с. – https://www.studentlibrary.ru/book/ISBN9785970605523.html
2	Хилл, К. Научное программирование на Python [Электронный ресурс] / Хилл К. – Москва : ДМК Пресс, 2021/ – 646 с. – Книга из коллекции ДМК Пресс - Информатика https://e.lanbook.com/book/241031
3	Северанс, Ч. Р. Python для всех [Электронный ресурс] / Северанс Ч. Р. – Москва : ДМК Пресс, 2022. – 262 с. – Книга из коллекции ДМК Пресс - Информатика https://e.lanbook.com/book/241115
4	Уилкс, М. Профессиональная разработка на Python [Электронный ресурс] / Уилкс М. – Москва : ДМК Пресс, 2021. – 502 с. – Книга из коллекции ДМК Пресс - Информатика https://e.lanbook.com/book/241121
5	Златопольский, Д. М. Основы программирования на языке Python [Электронный ресурс] / Златопольский Д. М. 2-ое изд., испр. и доп. – Москва : ДМК Пресс, 2018. – 396 с. – Книга из коллекции ДМК Пресс - Информатика https://e.lanbook.com/book/131683
6	Груздев, А. В. Изучаем Pandas [Электронный ресурс] / Груздев А. В., Хейдт М. 2-ое изд., испр. и доп. – Москва : ДМК Пресс, 2019. – 700 с. – Книга из коллекции ДМК Пресс - Информатика https://e.lanbook.com/book/131693
7	Дауни, А. Б. Изучение сложных систем с помощью Python [Электронный ресурс] / Дауни А. Б. – Москва : ДМК Пресс, 2019. – 160 с. – Книга из коллекции ДМК Пресс - Информатика https://e.lanbook.com/book/131701

8	Саммерфилд, М. Python на практике [Электронный ресурс] / Саммерфилд М. ; Пер. с англ. Слинкин А.А. – Москва : ДМК Пресс, 2014. – 338 с. – Книга из коллекции ДМК Пресс - Информатика http://e.lanbook.com/books/element.php?pl1_id=66480
9	Лучано, Р. Python. К вершинам мастерства [Электронный ресурс] / Лучано Р. ; Пер. с англ. Слинкин А.А. – Москва : ДМК Пресс, 2016. – 768 с. – Книга из коллекции ДМК Пресс - Информатика https://e.lanbook.com/book/93273

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет)*:

№ п/п	Ресурс
1	Официальный сайт разработчика. – URL: https://www.python.org
2	Разработка программного обеспечения, технологии и наука. – URL: https://devpractice.ru/python-lessons
3	Учебный курс «Программирование на языке высокого уровня (Python)». – URL: https://www.yuripetrov.ru/edu/python/index.html
4	ЗНБ ВГУ: https://lib.vsu.ru/
5	Электронно-библиотечная система "Университетская библиотека online": http://biblioclub.ru/
6	Электронно-библиотечная система "Лань": https://e.lanbook.com/

16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
1	Бизли, Д. Python. Книга рецептов [Электронный ресурс] / Бизли Д., Джонс Б. К. – Москва : ДМК Пресс, 2019. – 646 с. – Книга из коллекции ДМК Пресс - Информатика https://e.lanbook.com/book/131723
2	Стивенсон, Б. Python. Сборник упражнений [Электронный ресурс] / Стивенсон Б. – Москва : ДМК Пресс, 2021. – 238 с. – Книга из коллекции ДМК Пресс - Информатика https://e.lanbook.com/book/241025

17. Образовательные технологии, используемые при реализации учебной дисциплины, включая дистанционные образовательные технологии (ДОТ, электронное обучение (ЭО), смешанное обучение):

При реализации дисциплины могут использоваться технологии электронного обучения и дистанционные образовательные технологии на базе портала edu.vsu.ru, а также другие доступные ресурсы сети Интернет.

18. Материально-техническое обеспечение дисциплины:

Аудитория для лекционных занятий: мультимедиа-проектор, экран для проектора, компьютер с выходом в сеть «Интернет». Специализированная мебель (столы ученические, стулья, доска). Программное обеспечение: LibreOffice v.5-7, программа для просмотра файлов формата pdf, браузер.

Компьютерный класс: специализированная мебель, персональные компьютеры на базе i3-9100-3,6ГГц, мониторы ЖК 19» (30 шт.), мультимедийный проектор, экран.

ПО: ОС Windows v.7, 8, 10, Набор утилит интерпретатор языка CPython, интерпретатор языка Anaconda, IDE PyCharm, редактор Jupiter.

19. Оценочные средства для проведения текущей и промежуточной аттестаций

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Наименование раздела дисциплины (модуля)	Компетенции	Индикаторы достижения компетенции	Оценочные средства
1	Объектно-ориентированное программирование: общая картина	ПК-2 ПК-3	ПК-2.1, ПК-2.2 ПК-2.3 ПК-3.1, ПК-3.2 ПК-3.3	Письменный опрос Лабораторные работы
2	Основы написания классов	ПК-2 ПК-3	ПК-2.1, ПК-2.2 ПК-2.3 ПК-3.1, ПК-3.2 ПК-3.3	Письменный опрос Лабораторные работы
3	Детали реализации классов	ПК-2 ПК-3	ПК-2.1, ПК-2.2 ПК-2.3 ПК-3.1, ПК-3.2 ПК-3.3	Письменный опрос Лабораторные работы
4	Перегрузка операций	ПК-2 ПК-3	ПК-2.1, ПК-2.2 ПК-2.3 ПК-3.1, ПК-3.2 ПК-3.3	Письменный опрос Лабораторные работы
5	Проектирование с использованием классов	ПК-2 ПК-3	ПК-2.1, ПК-2.2 ПК-2.3 ПК-3.1, ПК-3.2 ПК-3.3	Письменный опрос Лабораторные работы
6	Расширенные возможности классов	ПК-2 ПК-3	ПК-2.1, ПК-2.2 ПК-2.3 ПК-3.1, ПК-3.2 ПК-3.3	Письменный опрос Лабораторные работы
7	Основы исключений	ПК-2 ПК-3	ПК-2.1, ПК-2.2 ПК-2.3 ПК-3.1, ПК-3.2 ПК-3.3	Письменный опрос Лабораторные работы
8	Детали обработки исключений	ПК-2 ПК-3	ПК-2.1, ПК-2.2 ПК-2.3 ПК-3.1, ПК-3.2 ПК-3.3	Письменный опрос Лабораторные работы
9	Объекты исключений	ПК-2 ПК-3	ПК-2.1, ПК-2.2 ПК-2.3 ПК-3.1, ПК-3.2 ПК-3.3	Письменный опрос Лабораторные работы
10	Проектирование с использованием исключений	ПК-2 ПК-3	ПК-2.1, ПК-2.2 ПК-2.3 ПК-3.1, ПК-3.2 ПК-3.3	Письменный опрос Лабораторные работы
11	Декораторы	ПК-2 ПК-3	ПК-2.1, ПК-2.2 ПК-2.3 ПК-3.1, ПК-3.2 ПК-3.3	Письменный опрос Лабораторные работы
12	Метаклассы	ПК-2 ПК-3	ПК-2.1, ПК-2.2 ПК-2.3 ПК-3.1, ПК-3.2 ПК-3.3	Письменный опрос Лабораторные работы
Промежуточная аттестация форма контроля – зачёт с оценкой				Перечень вопросов к зачёту

20. Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

20.1. Текущий контроль успеваемости

Контроль успеваемости по дисциплине осуществляется с помощью письменного опроса и выполнения лабораторных работ

Перечень вопросов для письменных опросов

- 1.Изменение строкового представления экземпляров.
- 2.Настройка строкового форматирования.
- 3.Создание объектов, поддерживающих протокол менеджера контекста.
- 4.Экономия памяти при создании большого количества экземпляров.
- 5.Инкапсуляция имен в классе.
- 6.Создание управляемых атрибутов.
- 7.Вызов метода родительского класса.
- 8.Расширение свойства в подклассе.
- 9.Создание нового типа атрибута класса или экземпляра.
- 10.Использование лениво вычисляемых свойств.
- 11.Упрощение инициализации структур данных.
- 12.Определение интерфейса или абстрактного базового класса.
- 13.Реализации модели данных или системы типов.
- 14.Реализация собственных контейнеров.
- 15.Делегирование доступа к атрибуту.
- 16.Определение более одного конструктора в классе.
- 17.Создание экземпляра без вызова `init`.
- 18.Расширение классов с помощью миксин (`mixins`).
- 19.Реализация объектов с состоянием или конечных автоматов.
- 20.Вызов метода объекта с передачей имени метода в строке.
- 21.Управление памятью в циклических структурах данных.
- 22.Создание закешированных экземпляров.
- 23.Создание обертки для функции.
- 24.Сохранение метаданных функции при написании декораторов.
- 25.Снятие («разворачивание») декоратора.
- 26.Определение декораторов как классов.
- 27.Применение декораторов к методам класса и статическим методам.
- 28.Написание декораторов, которые добавляют аргументы обернутым функциям
- 29.Использование декораторов для исправления определений классов.
- 30.Использование метакласса для управления созданием экземпляров.
- 31.Захват порядка определения атрибутов класса.
- 32.Определение метакласса, принимающего необязательные аргументы.
- 33.Принудительная установка аргументной сигнатуры#при использовании. `*args` и `**kwargs`
- 34.Принуждение к использованию соглашений о кодировании в классах.
- 35.Программное определение классов.
- 36.Инициализация членов класса во время определения.
- 37.Реализация множественной диспетчеризации с помощью аннотаций функций.
- 38.Избежание повторяющихся методов свойств.
- 39.Выполнение кода с локальными побочными эффектами.
- 40.Парсинг и анализ исходного кода Python.
- 41.Дизассемблирование байт-кода Python.

42. Определение декоратора, принимающего аргументы.
43. Определение декоратора с настраиваемыми пользователем атрибутами.
44. Определение декоратора, принимающего необязательный аргумент.
45. Принудительная проверка типов в функции с использованием декоратора.
46. Определение декораторов как части класса.
47. Обработка множественных исключений.
48. Создание собственных исключений.
49. Возбуждение исключения в ответ на другое исключение.
50. Повторное возбуждение последнего исключения.
51. Вывод предупреждающих сообщений.
52. Отладка основных сбоев программы.
53. Тестирование отправки вывода в stdout.
54. Изменение объектов в юнит-тестах.
55. Проверка вызывающих исключения условий в рамках юнит-тестов.
56. Логирование вывода теста в файл.

Типовое задание для лабораторной работы

Лабораторная работа № 1

Цель работы: отработать на практике изменение строкового представления экземпляров с использованием методов `__str__()` и `__repr__()`.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей изменение строкового представления экземпляров с использованием методов `__str__()` и `__repr__()`. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую изменение строкового представления экземпляров с использованием методов `__str__()` и `__repr__()`. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 2

Цель работы: отработать на практике способы настройки строкового форматирования.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей настройку строкового форматирования. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую поддержку объектов кастомизированного форматирования через функцию `format()` и строковый метод. Класс и его методы должны

сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 3

Цель работы: отработать на практике создание объектов, поддерживающих протокол менеджера контекста.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей создание объектов, поддерживающих протокол менеджера контекста. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую создание объектов, поддерживающих протокол менеджера контекста. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 4

Цель работы: отработать на практике экономии памяти при создании большого количества экземпляров.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей экономии памяти при создании большого количества экземпляров.. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую экономии памяти при создании большого количества экземпляров.. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 5

Цель работы: отработать на практике инкапсуляцию имен в классе.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей инкапсуляцию имен в классе. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую инкапсуляцию имен в классе. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 6

Цель работы: отработать на практике создание управляемых атрибутов.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей создание управляемых атрибутов. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую создание управляемых атрибутов. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 7

Цель работы: отработать на практике вызов метода родительского класса.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей вызов метода родительского класса. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую вызов метода родительского класса. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 8

Цель работы: отработать на практике расширение свойства в подклассе.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей расширение свойства в подклассе. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую расширение свойства в подклассе. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 9

Цель работы: отработать на практике создание нового типа атрибута класса или экземпляра.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей создание нового типа атрибута класса или экземпляра. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую создание нового типа атрибута класса или экземпляра. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 10

Цель работы: отработать на практике использование лениво вычисляемых свойств.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей использование лениво вычисляемых свойств. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую использование лениво вычисляемых свойств. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 11

Цель работы: отработать на практике упрощение инициализации структур данных.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей упрощение инициализации

структур данных. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую упрощение инициализации структур данных. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 12

Цель работы: отработать на практике определение интерфейса или абстрактного базового класса.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей определение интерфейса или абстрактного базового класса. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую определение интерфейса или абстрактного базового класса. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 13

Цель работы: отработать на практике реализации модели данных или системы типов.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей модели данных или системы типов. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую модели данных или системы типов. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 14

Цель работы: отработать на практике реализацию собственных контейнеров.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей собственные контейнеры. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую собственные контейнеры. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 15

Цель работы: отработать на практике делегирование доступа к атрибуту.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей делегирование доступа к атрибуту. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую делегирование доступа к атрибуту. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 16

Цель работы: отработать на практике определение более одного конструктора в классе.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей определение более одного конструктора в классе. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую определение более одного конструктора в классе. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 17

Цель работы: отработать на практике создание экземпляра без вызова init.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей создание экземпляра без вызова `init`. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую создание экземпляра без вызова `init`. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 18

Цель работы: отработать на практике расширение классов с помощью миксин (`mixins`).

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей расширение классов с помощью миксин (`mixins`). Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую расширение классов с помощью миксин (`mixins`). Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 19

Цель работы: отработать на практике реализацию объектов с состоянием или конечных автоматов.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей объекты с состоянием или конечные автоматы. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую объекты с состоянием или конечные автоматы. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 20

Цель работы: отработать на практике вызов метода объекта с передачей имени метода в строке.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей вызов метода объекта с передачей имени метода в строке. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую вызов метода объекта с передачей имени метода в строке. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 21

Цель работы: отработать на практике работу с шаблонами проектирования.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей шаблон проектирования «Посетитель». Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую шаблон проектирования «Посетитель». Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 22

Цель работы: отработать на практике работу с шаблонами проектирования.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей шаблон «Посетитель» без рекурсии. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую шаблон «Посетитель» без рекурсии. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 23

Цель работы: отработать на практике управление памятью в циклических структурах данных.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей управление памятью в циклических структурах данных. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую управление памятью в циклических структурах данных. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 24

Цель работы: отработать на практике поддержку операция сравнения в классах.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей поддержку операция сравнения в классах. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую поддержку операция сравнения в классах. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

Лабораторная работа № 25

Цель работы: отработать на практике создание закешированных экземпляров.

Требования к выполнению работы: выполнение лабораторной работы предусматривает написание программы, реализующей создание закешированных экземпляров. Отчёт о работе проводится в виде собеседования и заключается в демонстрации работы и ответов на дополнительные вопросы.

Критерии оценки: для получения оценки «зачтено» необходимо показать высокий уровень владения теоретическим материалом, уметь объяснить принцип работы написанной программы, верно ответить на дополнительные вопросы.

Задание: написать программу, реализующую создание закешированных экземпляров. Класс и его методы должны сопровождаться документацией в соответствии с принятыми отраслевыми стандартами. Проверить работу программы на контрольном примере.

20.2. Промежуточная аттестация

Промежуточная аттестация по дисциплине осуществляется с помощью следующих оценочных средств: перечень вопросов к зачёту.

Перечень вопросов к зачету

- 1.Изменение строкового представления экземпляров.
- 2.Настройка строкового форматирования.
- 3.Создание объектов, поддерживающих протокол менеджера контекста.
- 4.Экономия памяти при создании большого количества экземпляров.
- 5.Инкапсуляция имен в классе.
- 6.Создание управляемых атрибутов.
- 7.Вызов метода родительского класса.
- 8.Расширение свойства в подклассе.
- 9.Создание нового типа атрибута класса или экземпляра.
- 10.Использование лениво вычисляемых свойств.
- 11.Упрощение инициализации структур данных.
- 12.Определение интерфейса или абстрактного базового класса.
- 13.Реализации модели данных или системы типов.
- 14.Реализация собственных контейнеров.
- 15.Делегирование доступа к атрибуту.
- 16.Определение более одного конструктора в классе.
- 17.Создание экземпляра без вызова init.
- 18.Расширение классов с помощью миксин (mixins).
- 19.Реализация объектов с состоянием или конечных автоматов.
- 20.Вызов метода объекта с передачей имени метода в строке.
- 21.Управление памятью в циклических структурах данных.
- 22.Создание закешированных экземпляров.
- 23.Создание обертки для функции.
- 24.Сохранение метаданных функции при написании декораторов.
- 25.Снятие («разворачивание») декоратора.
- 26.Определение декораторов как классов.
- 27.Применение декораторов к методам класса и статическим методам.
- 28.Написание декораторов, которые добавляют аргументы обернутым.
29. функциям
- 30.Использование декораторов для исправления определений классов.
- 31.Использование метакласса для управления созданием экземпляров.
- 32.Захват порядка определения атрибутов класса.
- 33.Определение метакласса, принимающего необязательные аргументы.
- 34.Принудительная установка аргументной сигнатуры при использовании *args и **kwargs.
- 35.Принуждение к использованию соглашений о кодировании в классах.
- 36.Программное определение классов.
- 37.Инициализация членов класса во время определения.
- 38.Реализация множественной диспетчеризации с помощью аннотаций
39. функций.
- 40.Избежание повторяющихся методов свойств.
- 41.Выполнение кода с локальными побочными эффектами.
- 42.Парсинг и анализ исходного кода Python.
- 43.Дизассемблирование байт-кода Python.
- 44.Определение декоратора, принимающего аргументы.
- 45.Определение декоратора с настраиваемыми пользователем атрибутами.
- 46.Определение декоратора, принимающего необязательный аргумент.

47. Принудительная проверка типов в функции с использованием декоратора.
48. Определение декораторов как части класса.
49. Обработка множественных исключений.
50. Создание собственных исключений.
51. Возбуждение исключения в ответ на другое исключение.
52. Повторное возбуждение последнего исключения.
53. Вывод предупреждающих сообщений.
54. Отладка основных сбоев программы.
55. Тестирование отправки вывода в stdout.
56. Изменение объектов в юнит-тестах.
57. Проверка вызывающих исключения условий в рамках юнит-тестов.
58. Логирование вывода теста в файл.

Для оценивания результатов обучения на зачёте с оценкой используется 4-балльная шкала: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
Дан полный, развёрнутый ответ на поставленный вопрос (вопросы), обучающийся свободно оперирует основными понятиями дисциплины, ориентируется в предметной области. Изложение материала не содержит ошибок, отличается последовательностью, грамотностью, логической стройностью.	Повышенный уровень	Отлично
Дан развёрнутый ответ на поставленный вопрос (вопросы), обучающийся свободно оперирует основными понятиями дисциплины, ориентируется в предметной области. Материал изложен в целом последовательно и грамотно, отсутствуют грубые ошибки, однако имеются отдельные неточности в определениях, вычислениях, доказательствах, изложениях положений теории.	Базовый уровень	Хорошо
Ответ на поставленный вопрос (вопросы) содержит изложение только базового теоретического материала, имеются ошибки в определениях, вычислениях, доказательствах, формулировках положений теории. Нарушена логическая последовательность в изложении материала.	Пороговый уровень	Удовлетворительно
Ответ на поставленный вопрос (вопросы) отсутствует, либо содержит грубые ошибки в определениях, вычислениях, доказательствах, формулировках положений теории. Обучающийся не владеет основными понятиями дисциплины. Отсутствует логическая последовательность в изложении материала.	–	Неудовлетворительно