

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ»

«УТВЕРЖДАЮ»

Проректор по учебной работе  Г.Ю. Нагорная
«16» 01 2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Объектно-ориентированное программирование

Уровень образовательной программы бакалавриат

Направление подготовки 01.03.02 Прикладная математика и информатика

Направленность(профиль) «Математические и информационные системы и технологии в астрономии»

Форма обучения: очная


Срок освоения ОП 4 года


Институт Цифровых технологий

Кафедра разработчик РПД Математика

Выпускающая кафедра Астрофизика

Начальник учебно-методического управления  Семенова Л. У.

Директор института ЦТ  Кумратова А. М.

И. О. заведующего выпускающей кафедрой  Валявин Г. Г.

г. Черкесск, 2026 г

СОДЕРЖАНИЕ

1	Цели освоения дисциплины	4
2	Место дисциплины в структуре образовательной программы	5
3	Планируемые результаты обучения по дисциплине	6
4	Структура и содержание дисциплины	7
	4.1. Объем дисциплины и виды учебной работы	7
	4.2. Содержание дисциплины	8
	4.2.1. Разделы (темы) дисциплины, виды учебной деятельности и формы контроля	8
	4.2.2. Лекционный курс	9
	4.2.3. Лабораторный практикум	11
	4.2.4. Практические занятия	14
	4.3. Самостоятельная работа обучающегося	15
5	Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	17
6	Образовательные технологии	27
7	Учебно-методическое и информационное обеспечение дисциплины	30
	7.1. Перечень основной и дополнительной учебной литературы	30
	7.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»	31
	7.3. Информационные технологии, лицензионное программное обеспечение	31
8	Материально-техническое обеспечение дисциплины	32
	8.1. Требования к аудиториям (помещениям, местам) для проведения занятий	32
	8.2. Требования к оборудованию рабочих мест преподавателя и обучающихся:	33
	8.3. Требования к специализированному оборудованию	33
9	Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья	34
	Приложение 1. Фонд оценочных средств	35

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины «Объектно – ориентированное программирование» является изучение основ объектно–ориентированного программирования и принципов создания приложений, поддерживающих требования интерфейса операционной среды WINDOWS. В качестве инструментального средства разработки приложений изучается RAD среда LAZARUS компании Borland Software Corporation (программа ориентирована на Lazarus 1.6) и среда MS Visual Studio компании Microsoft. Объектно–ориентированное программирование сегодня является наиболее мощным средством, позволяющим как моделировать объекты реального мира, так и генерировать виртуальные объекты, существующие только в электронной ауре компьютера. Использование визуального программирования существенно облегчает разработку интерфейса приложений, ориентированного на стандарт визуальных оболочек.

Рассмотрение конкретных приемов, как для создания отдельных объектных классов, так и построения виртуальной среды, насыщенной объектами различных классов, в структуру которых инкапсулированы возможности обмена данными и управления, как между собой, так и с внешними объектами. В результате обучающийся должен уметь пользоваться готовыми объектно-ориентированными конструкциями, расширяя их структуру и функционал и уметь разрабатывать в стиле объектного подхода собственные конструкции произвольной структурной сложности.

Задачи дисциплины:

- обучение основам объектно–ориентированного подхода в программировании информационных, вычислительных и имитационных систем;
- привить обучающимся знания использования основных компонент разработки приложений;
- дать опыт разработки собственных средств контроля и управления элементами программных систем;
- ознакомить обучающихся с принципами функционирования и управления специальными средствами WINDOWS–программирования (реализация многозадачности, СОМ технологии).

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

2.1. Дисциплина «Объектно – ориентированное программирование» относится к части, формируемой участниками образовательных Блока 1 Дисциплины (модули) имеет тесную связь с другими дисциплинами.

2.2. В таблице приведены предшествующие и последующие дисциплины, направленные на формирование компетенций дисциплины в соответствии с матрицей компетенций ОП.

Предшествующие и последующие дисциплины, направленные на формирование компетенций

№ п/п	Предшествующие дисциплины	Последующие дисциплины
1.	Основы информатики	Системы и методы компьютерного моделирования
2.	Процедурное программирование	Учебная практика
3.	Основы программирования	(Технологическая
4.		(проектно – технологическая) практика)

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Планируемые результаты освоения образовательной программы (ОП) – компетенции обучающихся определяются требованиями стандарта по направлению подготовки 01.03.02 Прикладная математика и информатика и формируются в соответствии с матрицей компетенций ОП

№ п/п	Номер/индекс компетенции	Наименование компетенции (или ее части)	Индикаторы достижения компетенций
1	2	3	4
1.	ПК-3	Способен осуществлять программную реализацию алгоритмов при разработке прикладных информационных систем	ПК-3.1 Обладает знаниями в области программирования и современных компьютерных технологий при разработке прикладных информационных систем. Способен осуществлять оптимизацию алгоритмов при создании прикладных программ или прикладных комплексов ПК-3.2 Анализирует стандартные технические требования к проектированию программного обеспечения. ПК-3.3 Способен осуществлять разработку технических требований к программному обеспечению, при его создании

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Вид учебной работы		Всего часов	Семестры*	
			№ 3	№4
			часов	часов
1	2	3	4	
Аудиторная контактная работа (всего)		108	54	54
В том числе:				
Лекции (Л)		36	18	18
Практические занятия (ПЗ), Семинары (С)				
Лабораторные работы (ЛР)		72	36	36
Внеаудиторная контактная работа, в том числе:		3,7	1,7	2
Индивидуальные и групповые консультации		3,7	1,7	2
Самостоятельная работа обучающегося (СРО)** (всего)		77	52	25
<i>Реферат (Реф)</i>		15	10	5
<i>Подготовка к лабораторным занятиям</i>		15	10	5
<i>Работа с книжными источниками</i>		15	10	5
<i>Работа с электронными источниками</i>		10	5	5
<i>Подготовка к промежуточному контролю (ППК)</i>		8	5	3
<i>Самоподготовка</i>		14	12	2
Промежуточная аттестация	зачет (За) в том числе:	За	За	-
	Прием зачета с оценкой, час	0,3	0,3	-
	экзамен (Э) в том числе:	Э(36)	-	Э (36)
	Прием экз., час.	0,5	-	0,5
	Консультация, час.	2	-	2
	СРО, час.	24,5	-	24,5
ИТОГО: Общая трудоемкость	часов	216	108	108
	зач. ед.	6	3	3

4.2. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.2.1. Разделы (темы) дисциплины, виды учебной деятельности и формы контроля

№ п/п	Наименование раздела (темы) дисциплины	Виды учебной деятельности, включая самостоятельную работу обучающихся (в часах)					Формы текущей и промежуточной аттестации
		Л	ЛР	ПЗ	СРО	всего	
1	2	3	4	5	6	7	8
Семестр 3							
1.	Раздел 1. Введение в объектно-ориентированное программирование.	2	2	-	18	22	Коллоквиум, контрольные вопросы, тестирование. Реферат
2.	Раздел 2. Классы и объекты. Атрибуты и методы. Инкапсуляция. Модификаторы доступа.	6	12	-	18	36	Коллоквиум, контрольные вопросы, тестирование. Реферат
3.	Раздел 3. Наследование. Создание иерархий классов. Статические методы. Абстрактные классы.	10	22	-	16	48	Коллоквиум, контрольные вопросы, тестирование. Реферат
4.	Внеаудиторная контактная работа					1,7	групповые и индивидуальные консультации
5.	Промежуточная аттестация					0,3	Зачет
Всего часов в 3 семестре		18	36	-	52	108	
Семестр 4							
6.	Раздел 4. Модули и пакеты. Установка пакетов.	6	4		10	20	Коллоквиум, контрольные вопросы, индивидуальные задания к лабораторным работам. Тестирование
7.	Раздел 5. Работа с файлами. Модули работы с файлами	6	16		5	27	Контрольные вопросы, индивидуальные задания к лабораторным работам, тестирование

8.	Раздел 6. Различные библиотеки Python	6	16		10	32	Контрольные вопросы, индивидуальные задания к лабораторным работам, тестирование
9.	Внеаудиторная контактная работа					2	групповые и индивидуальные консультации
10.	Промежуточная аттестация					0,5	Экзамен
11.						26,5	
Всего часов в 4 семестре		18	36		25	108	
ИТОГО:		36	72		77	216	

4.2.2. Лекционный курс

№ п/п	Наименование раздела дисциплины	Наименование темы лекции	Содержание лекции	Всего часов
1	2	3	4	5
Семестр 3,4				
1.	Раздел 1. Введение в объектно-ориентированное программирование. Раздел 2. Классы и объекты. Атрибуты и методы. Инкапсуляция. Модификаторы доступа.	Тема 1.1 Переход от структурного программирования к ООП. Преимущества ООП для создания сложных, масштабируемых и поддерживаемых приложений.	Исторический аспект возникновения объектно-ориентированного подхода в программировании. Принципы структурного программирования. Структурная декомпозиция. Виды структурирования данных. Модульный подход.	2
2.		Тема 2.1 Классы и объекты.	Классы и объекты. Поля и методы объектов. Жизненный цикл объектов. Области видимости. Конструкторы и деструкторы.	2
3.		Тема 2.2 Свойства объектов. Инкапсуляция.	Принцип инкапсуляции. Обычные атрибуты (прямой доступ) Методы getter/setter Свойства через property()	2
4.		Тема 2.3 Инкапсуляция. Модификаторы доступа	Публичный модификатор доступа. Соглашение о внутреннем использовании. Приватные члены.	2

			Защищенные члены.	
5.	Раздел 3. Наследование. Создание иерархий классов. Статические методы. Абстрактные классы.	Тема 3.1 Наследование. Множественное наследование	Наследование. Подкласс и суперкласс. Множественное наследование	2
		Тема 3.2 Переопределение функционала	Переопределение функционала базового класса. Вызов конструктора классов. Проверка типа объекта	4
		Тема 3.3 Атрибуты классов и статические методы	Атрибуты классов и статические методы. Статические методы. Аннотация @staticmethod	2
		Тема 3.4 Абстрактные классы	Перегрузка операторов. Истинность объекта. Операции обращения по индексу. Абстрактные классы и методы	2
6.	Раздел 4. Модули и пакеты. Установка пакетов.	Тема 4.1 Модули	Определение и подключение модулей Подключение функциональности модуля в глобальное пространство имен	2
		Тема 4.2 Генерация байткода модулей	Установка псевдонимов. Две стадии выполнения скрипта на языке Python. Ручная компиляция байткода	2
		Тема 4.3 Установка пакетов	Установка пакетов, pip и виртуальная среда. Менеджер пакетов. Установка виртуальной среды	2
7.	Раздел 5. Работа с файлами. Модули работы с файлами	Тема 5.1 Работа с файлами. Файлы CSV. Бинарные файлы	Открытие и закрытие файлов. Текстовые файлы. Чтение файла и запись. Работа со словарями. Модуль pickle. Модуль shelve. Чтение данных	2
		Тема 5.2 Модуль OS и работа с файловой системой	Встроенный модуль OS. Создание и удаление папки. Переименование файла. Удаление файла	2
		Тема 5.2 Запись и чтение архивных zip-файлов	Создание и закрытие файла. Запись файлов в архив. Получение информации о файлах в архиве	2
8.	Раздел 6. Различные библиотеки Python	Тема 6.1 Научные вычисления и Data Science	NumPy: Фундаментальная библиотека для работы с многомерными массивами и матрицами,	4

			математическими функциями. SciPy: Научные и инженерные вычисления (оптимизация, интеграция, линейная алгебра, статистика).	
		Тема 6.2 Визуализация данных	Matplotlib: Базовая и гибкая библиотека для построения статических, анимированных и интерактивных графиков. Seaborn: Высокоуровневая библиотека на основе Matplotlib для создания статистических графиков красивее и проще.	2
ИТОГО часов в 3,4 семестре:				36
ВСЕГО часов:				36

4.2.3. Лабораторный практикум

№ п/п	Наименование раздела дисциплины	Наименование лабораторной работы	Содержание лабораторной работы	Всего часов
1	2	3	4	5
Семестр 3				
	Раздел 1. Введение в объектно-ориентированное программирование.	Лабораторная работа №1 Организация работы с в Visual studio code	Создание консольного приложения в Visual studio code. Знакомство с функционалом Visual studio code	2
	Раздел 2. Классы и объекты. Атрибуты и методы. Инкапсуляция. Модификаторы доступа.	Лабораторная работа № 2 Введение в классы и объекты. Инкапсуляция	Проектировать простые классы, создавать объекты. Создание графического многооконного интерфейса в TkInter. Оформление и защита	4
		Лабораторная работа №3 Создание простых многооконных приложений под Windows в среде Visual studio code	Проектировать простые классы, создавать объекты, использовать модификаторы доступа (инкапсуляция). Концепция сокрытия данных (private/public). Реализовать класс, моделирующий реальный объект или сущность. Оформление и защита	4
		Лабораторная работа №4 Технология отладки и	Получение практических навыков по тестированию	4

		тестирования программ	и отладке программ. Оформление и защита.	
	Раздел 3. Наследование. Создание иерархий классов. Статические методы. Абстрактные классы.	Лабораторная работа № 5 Технология разработки, размещения и использование различных компонентов в TkInter	Получение практических навыков применения различных компонентов в TkInter Оформление и защита.	8
		Лабораторная работа №6 Наследование и полиморфизм.	Базовый (родительский) класс с общими полями и методами. Не менее 2-х производных (дочерних) классов, расширяющих базовый. Оформление и защита.	8
		Лабораторная работа №7 Абстрактные классы и интерфейсы	Переопределение одного метода в производных классах (@Override). В main продемонстрировать полиморфизм: создать массив/список типа базового класса, содержащий объекты производных классов, и вызвать у них переопределенный метод. Оформление и защита.	6
ИТОГО часов в 3 семестре:				36
	Раздел 4. Модули и пакеты. Установка пакетов.	Лабораторная работа №8 Работа с модулями. Установка пакетов	Принципы организации кода с использованием модулей и пакетов. Методы установки, обновления и удаления пакетов в Python. Создавание собственных модулей и пакетов. Практические навыки работы с pip, virtualenv/venv. Оформление и защита.	4
	Раздел 5. Работа с файлами. Модули работы с файлами	Лабораторная работа №9 Работа с файлами в Python	Принципы создания и использования модулей и пакетов в Python. Работа с файлами (чтение, запись, обработка данных). Применить полученные знания для решения индивидуальных задач. Оформление и защита.	8
		Лабораторная работа №10 Модули работы с файлами в Python	Основные методы работы с файлами в Python. Стандартные модули для работы с файлами различных форматов.	8

			Обработка исключения при работе с файловой системой Навыки создания собственных модулей для работы с файлами Оформление и защита.	
	Раздел 6. Различные библиотеки Python	Лабораторная работа №11 Работа с библиотеками NumPy, SciPy	Освоение основных возможностей библиотек NumPy и SciPy для научных вычислений, обработки данных и решения математических задач. Оформление и защита.	8
		Лабораторная работа №12 Визуализация данных. Библиотеки Matplotlib, Seaborn	Основные методы визуализации данных с использованием библиотек Matplotlib и Seaborn, научиться выбирать подходящие типы графиков для различных задач анализа данных. Оформление и защита.	8
ИТОГО часов в семестре:				36
ВСЕГО часов:				72

4.3. САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩЕГОСЯ

№ п/п	Наименование раздела (темы) дисциплины	№ п/п	Виды СРО	Всего часов
1	2	3	4	5
Семестр 3				
1.	Раздел 1. Введение в объектно-ориентированное программирование.	1.1.	Работа с книжными и электронными источниками. Подготовка к лабораторным занятиям. Реферат. Подготовка к промежуточному контролю. Самоподготовка	18
2.	Раздел 2. Классы и объекты. Атрибуты и методы. Инкапсуляция. Модификаторы доступа	2.1.	Работа с книжными и электронными источниками. Подготовка к лабораторным занятиям. Реферат. Подготовка к промежуточному контролю. Самоподготовка	6
		2.2.	Работа с книжными и электронными источниками. Подготовка к лабораторным занятиям. Реферат. Подготовка к	6

			промежуточному контролю. Самоподготовка	
		2.3.	Работа с книжными и электронными источниками. Подготовка к лабораторным занятиям. Реферат. Подготовка к промежуточному контролю. Самоподготовка	6
3.	Раздел 3. Наследование. Создание иерархий классов. Статические методы. Абстрактные классы.	3.1	Работа с книжными и электронными источниками. Подготовка к лабораторным занятиям. Реферат. Подготовка к промежуточному контролю. Самоподготовка	4
		3.2	Работа с книжными и электронными источниками. Подготовка к лабораторным занятиям. Реферат. Подготовка к промежуточному контролю. Самоподготовка	4
		3.3	Работа с книжными и электронными источниками. Подготовка к лабораторным занятиям. Реферат. Подготовка к промежуточному контролю. Самоподготовка	4
		3.4	Работа с книжными и электронными источниками. Подготовка к лабораторным занятиям. Реферат. Подготовка к промежуточному контролю. Самоподготовка	4
ИТОГО часов в 4 семестре:				52
Семестр 5				
4.	Раздел 4. Модули и пакеты. Установка пакетов.	4.1	Работа с книжными и электронными источниками. Подготовка к лабораторным занятиям. Реферат. Подготовка к промежуточному контролю. Самоподготовка	4
		4.2	Работа с книжными и электронными источниками. Подготовка к лабораторным занятиям. Реферат. Подготовка к промежуточному контролю. Самоподготовка	3
		4.3	Работа с книжными и электронными источниками. Подготовка к лабораторным занятиям. Реферат. Подготовка к промежуточному контролю. Самоподготовка	3
5.	Раздел 5. Работа с файлами. Модули работы с файлами.	5.1	Работа с книжными и электронными источниками. Подготовка к лабораторным занятиям. Реферат. Подготовка к промежуточному контролю. Самоподготовка	1
		5.2	Работа с книжными и электронными источниками. Подготовка к лабораторным занятиям. Реферат. Подготовка к промежуточному контролю. Самоподготовка	3
		5.3	Работа с книжными и электронными источниками. Подготовка к лабораторным занятиям. Реферат. Подготовка к промежуточному контролю. Самоподготовка	1

			занятиям. Реферат. Подготовка к промежуточному контролю. Самоподготовка	
6.	Раздел 6. Различные библиотеки Python	6.1	Работа с книжными и электронными источниками. Подготовка к лабораторным занятиям. Реферат. Подготовка к промежуточному контролю. Самоподготовка	5
		6.2	Работа с книжными и электронными источниками. Подготовка к лабораторным занятиям. Реферат. Подготовка к промежуточному контролю. Самоподготовка	5
ИТОГО часов в 5 семестре:				25
ВСЕГО часов:				77

5. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

5.1. Методические указания для подготовки обучающихся к лекционным занятиям

Какими бы замечательными качествами в области методики ни обладал лектор, какое бы большое значение на занятиях ни уделял лекции слушатель, глубокое понимание материала достигается только путем самостоятельной работы над ним.

Работа над конспектом лекции осуществляется по этапам:

- повторить изученный материал по конспекту;
- непонятные положения отметить на полях и уточнить;
- неоконченные фразы, пропущенные слова и другие недочеты в записях устранить, пользуясь материалами из учебника и других источников;
- завершить техническое оформление конспекта (подчеркивания, выделение главного, выделение разделов, подразделов и т.п.).

Самостоятельную работу следует начинать с доработки конспекта, желательно в тот же день, пока время не стерло содержание лекции из памяти (через 10 ч после лекции в памяти остается не более 30-40 % материала). Работа над конспектом не должна заканчиваться с прослушивания лекции. После лекции, в процессе самостоятельной работы, перед тем, как открыть тетрадь с конспектом, полезно мысленно восстановить в памяти содержание лекции, вспомнив ее структуру, основные положения и выводы.

С целью доработки необходимо прочитать записи, восстановить текст в памяти, а также исправить описки, расшифровать не принятые ранее сокращения, заполнить пропущенные места, понять текст, вникнуть в его смысл. Далее прочитать материал по рекомендуемой литературе, разрешая в ходе чтения, возникшие ранее затруднения, вопросы, а также дополнения и исправляя свои записи. Записи должны быть наглядными, для чего следует применять различные способы выделений. В ходе доработки конспекта углубляются, расширяются и закрепляются знания, а также дополняется, исправляется и совершенствуется конспект. Еще лучше, если вы переработаете конспект, дадите его в

новой систематизации записей. Это, несомненно, займет некоторое время, но материал вами будет хорошо проработан, а конспективная запись его приведена в удобный для запоминания вид. Введение заголовков, скобок, обобщающих знаков может значительно повысить качество записи. Этому может служить также подчеркивание отдельных мест конспекта красным карандашом, приведение на полях или на обратной стороне листа краткой схемы конспекта и др.

Подготовленный конспект и рекомендуемая литература используется при подготовке к практическому (семинарскому) занятию. Подготовка сводится к внимательному прочтению учебного материала, к выводу с карандашом в руках всех утверждений и формул, к решению примеров, задач, к ответам на вопросы, предложенные в конце лекции преподавателем или помещенные в рекомендуемой литературе. Примеры, задачи, вопросы по теме являются средством самоконтроля.

Непременным условием глубокого усвоения учебного материала является знание основ, на которых строится изложение материала. Обычно преподаватель напоминает, какой ранее изученный материал и в какой степени требуется подготовить к очередному занятию. Эта рекомендация, как и требование систематической и серьезной работы над всем лекционным курсом, подлежит безусловному выполнению. Потери логической связи как внутри темы, так и между ними приводит к негативным последствиям: материал дисциплины перестает основательно восприниматься, а творческий труд подменяется утомленным переписыванием. Обращение к ранее изученному материалу не только помогает восстановить в памяти известные положения, выводы, но и приводит разрозненные знания в систему, углубляет и расширяет их. Каждый возврат к старому материалу позволяет найти в нем что-то новое, переосмыслить его с иных позиций, определить для него наиболее подходящее место в уже имеющейся системе знаний. Неоднократное обращение к пройденному материалу является наиболее рациональной формой приобретения и закрепления знаний. Очень полезным, но, к сожалению, еще мало используемым в практике самостоятельной работы, является предварительное ознакомление с учебным материалом. Даже краткое, беглое знакомство с материалом очередной лекции дает многое. Обучающиеся получают общее представление о ее содержании и структуре, о главных и второстепенных вопросах, о терминах и определениях. Все это облегчает работу на лекции и делает ее целеустремленной.

5.2. Методические указания для подготовки обучающихся к лабораторным занятиям

Ведущей дидактической целью лабораторных занятий является систематизация и обобщение знаний по изучаемой теме, приобретение практических навыков по тому или другому разделу курса, закрепление практически полученных теоретических знаний.

В начале каждого лабораторного занятия кратко приводится теоретический материал, необходимый для решения задач по данной теме. После него предлагается решение этих задач и список заданий для самостоятельного выполнения.

Практическая работа включает в себя самоконтроль по предложенным вопросам, выполнение творческих и проверочных заданий, тестирование по теме.

Лабораторные работы сопровождают и поддерживают лекционный курс.

Количество лабораторных работ в строгом соответствии с содержанием курса. Каждая лабораторная предусматривает получение практических навыков по лекционным темам дисциплины «Объектно-ориентированное программирование». Для обучающихся подготовлен набор индивидуальных заданий по каждой лабораторной работе. В каждой лабораторной работе обучающийся оформляет полученные результаты. Также в текущей аттестации к лабораторным занятиям предусмотрена форма контроля в виде устной защиты каждого практического индивидуального задания по всем темам лабораторных занятий.

При проведении промежуточной и итоговой аттестации обучающихся важно всегда

помнить, что систематичность, объективность, аргументированность — главные принципы, на которых основаны контроль и оценка знаний обучающихся. Проверка, контроль и оценка знаний.

По окончании курса обучающимися сдается экзамен, в ходе которого они должны показать свои теоретические знания и практические навыки в Объектно – ориентированном программировании.

5.3 Методические рекомендации прохождения тестирования

Подготовку к итоговому тестированию необходимо осуществлять поэтапно.

На первом этапе необходимо повторить основные положения всех тем, детально разбирая наиболее сложные моменты. Непонятные вопросы необходимо выписывать, чтобы по ним можно было проконсультироваться с преподавателем перед прохождением итогового тестирования. Подготовку по темам каждой дидактической единицы целесообразно производить отдельно. На этом этапе необходимо использовать материалы лекционного курса, материалы семинарских занятий, тестовые задания для текущего контроля знаний, а также презентации лекционного курса.

На втором этапе подготовки предлагается без повторения теоретического материала дать ответы тестовые задания для рубежного контроля знаний. Если ответы на какие-то вопросы вызвали затруднение, необходимо еще раз повторить соответствующий теоретический материал.

Наконец, третий этап подготовки необходимо осуществить непосредственно накануне теста. На данном этапе необходимо аккуратно просмотреть весь лекционный курс.

В случае, если результаты выполнения тестового задания оказались неудовлетворительными, необходимо зафиксировать темы, на вопросы по которым были даны неверные ответы, и еще раз углубленно повторить соответствующие темы в соответствии с указанными выше тремя этапами подготовки к тестированию.

5.4 Методические рекомендации прохождения коллоквиума

Коллоквиумом называется собеседование преподавателя и студента по заранее определенным контрольным вопросам. Целью коллоквиума является формирование у студента навыков анализа теоретических проблем на основе самостоятельного изучения учебной и научной литературы. На коллоквиум выносятся крупные, проблемные, нередко спорные теоретические вопросы. Упор делается на монографические работы профессора-автора данного спецкурса. От студента требуется:

- владение изученным в ходе учебного процесса материалом, относящимся к рассматриваемой проблеме;
- знание разных точек зрения, высказанных в научной литературе по соответствующей проблеме, умение сопоставлять их между собой;
- наличие собственного мнения по обсуждаемым вопросам и умение его аргументировать.

Коллоквиум – это не только форма контроля, но и метод углубления, закрепления знаний студентов, так как в ходе собеседования преподаватель разъясняет сложные вопросы, возникающие у студента в процессе изучения данного источника. Однако коллоквиум не консультация и не экзамен. Его задача добиться глубокого изучения отобранного материала, пробудить у студента стремление к чтению дополнительной социологической литературы.

Подготовка к коллоквиуму.

Подготовка к коллоквиуму начинается с установочной консультации преподавателя, на которой он разъясняет развернутую тематику проблемы, рекомендует

литературу для изучения и объясняет процедуру проведения коллоквиума. Как правило, на самостоятельную подготовку к коллоквиуму студенту отводится 3 – 4 недели. Методические указания состоят из рекомендаций по изучению источников и литературы, вопросов для самопроверки и кратких конспектов ответа с перечислением основных фактов и событий, относящихся к пунктам плана каждой темы. Это должно помочь студентам целенаправленно организовать работу по овладению материалом и его запоминанию. При подготовке к коллоквиуму следует, прежде всего, просмотреть конспекты лекций и практических занятий и отметить в них имеющиеся вопросы коллоквиума. Если какие-то вопросы вынесены преподавателем на самостоятельное изучение, следует обратиться к учебной литературе, рекомендованной преподавателем в качестве источника сведений.

Коллоквиум проводится в форме индивидуальной беседы преподавателя с каждым студентом или беседы в небольших группах (2 – 3 человека). Обычно преподаватель задает несколько кратких конкретных вопросов, позволяющих выяснить степень добросовестности работы с литературой, проверяет конспект. Далее более подробно обсуждается какая-либо сторона проблемы, что позволяет оценить уровень понимания. По итогам коллоквиума выставляется дифференцированная оценка по пятибалльной системе.

5.6 Методические указания по подготовке к опросу (контрольные вопросы)

Самостоятельная работа обучающихся включает подготовку к опросу на практическом занятии. Опрос представляет собой форму текущего контроля успеваемости обучающегося по изучаемой дисциплине. При подготовке к опросу необходимо изучить материалы лекции, основную и дополнительную литературу, а также информацию с использованием Интернет-ресурсов по заявленной теме. Темы практических занятий, вопросы для обсуждения, а также контрольные вопросы даются в методических указаниях по соответствующим темам дисциплины. Обучающийся должен обратить внимание на основные термины и понятия по теме, на проблемные вопросы, подобрать дополнительную литературу для их освещения, составить тезисы выступления. Ответ обучающегося должен быть развернутым, аргументированным, логически выстроенным. При выставлении оценки учитывается правильность ответа по содержанию, самостоятельность суждений и выводов, умение анализировать и связывать теоретические положения с практикой.

5.5 Методические рекомендации прохождения коллоквиума

Коллоквиумом называется собеседование преподавателя и студента по заранее определенным контрольным вопросам. Целью коллоквиума является формирование у студента навыков анализа теоретических проблем на основе самостоятельного изучения учебной и научной литературы. На коллоквиум выносятся крупные, проблемные, нередко спорные теоретические вопросы. Упор делается на монографические работы профессора-автора данного спецкурса. От студента требуется:

- владение изученным в ходе учебного процесса материалом, относящимся к рассматриваемой проблеме;
- знание разных точек зрения, высказанных в научной литературе по соответствующей проблеме, умение сопоставлять их между собой;
- наличие собственного мнения по обсуждаемым вопросам и умение его аргументировать.

Коллоквиум – это не только форма контроля, но и метод углубления, закрепления знаний студентов, так как в ходе собеседования преподаватель разъясняет сложные вопросы, возникающие у студента в процессе изучения данного источника. Однако коллоквиум не консультация и не экзамен. Его задача добиться глубокого изучения

отобранного материала, пробудить у студента стремление к чтению дополнительной социологической литературы.

Подготовка к коллоквиуму.

Подготовка к коллоквиуму начинается с установочной консультации преподавателя, на которой он разъясняет развернутую тематику проблемы, рекомендует литературу для изучения и объясняет процедуру проведения коллоквиума. Как правило, на самостоятельную подготовку к коллоквиуму студенту отводится 3 – 4 недели. Методические указания состоят из рекомендаций по изучению источников и литературы, вопросов для самопроверки и кратких конспектов ответа с перечислением основных фактов и событий, относящихся к пунктам плана каждой темы. Это должно помочь студентам целенаправленно организовать работу по овладению материалом и его запоминанию. При подготовке к коллоквиуму следует, прежде всего, просмотреть конспекты лекций и практических занятий и отметить в них имеющиеся вопросы коллоквиума. Если какие-то вопросы вынесены преподавателем на самостоятельное изучение, следует обратиться к учебной литературе, рекомендованной преподавателем в качестве источника сведений.

Коллоквиум проводится в форме индивидуальной беседы преподавателя с каждым студентом или беседы в небольших группах (2 – 3 человека). Обычно преподаватель задает несколько кратких конкретных вопросов, позволяющих выяснить степень добросовестности работы с литературой, проверяет конспект. Далее более подробно обсуждается какая-либо сторона проблемы, что позволяет оценить уровень понимания. По итогам коллоквиума выставляется дифференцированная оценка по пятибалльной системе.

5.6 Методические указания по самостоятельной работе обучающихся

Самостоятельная работа обучающегося предполагает различные формы индивидуальной учебной деятельности: конспектирование научной литературы, сбор и анализ практического материала в СМИ, проектирование, выполнение тематических и творческих заданий и пр. Выбор форм и видов самостоятельной работы определяется индивидуально-личностным подходом к обучению совместно преподавателем и обучающимся. Формы текущего контроля успеваемости и промежуточной аттестации обучающихся.

Содержание внеаудиторной самостоятельной работы обучающихся по дисциплине «Объектно – ориентированное программирование» включает в себя различные виды деятельности:

- чтение текста (учебника, первоисточника, дополнительной литературы);
- составление плана текста;
- конспектирование текста;
- работа со словарями и справочниками;
- ознакомление с нормативными документами;
- исследовательская работа;
- использование аудио- и видеозаписи;
- работа с электронными информационными ресурсами;
- выполнение тестовых заданий;
- ответы на контрольные вопросы;
- аннотирование, реферирование, рецензирование текста;
- составление глоссария, кроссворда или библиографии по конкретной теме;
- решение вариативных задач и упражнений.

Рекомендации по подготовке реферата

Реферат является формой самостоятельной учебной работы по предмету, направленной на детальное знакомство с какой-либо темой в рамках данной дисциплины. Основная задача работы над рефератом по предмету — углубленное изучение определенной проблемы изучаемого курса, получение более полной информации по какому-либо его разделу.

При подготовке реферата необходимо использовать достаточное для раскрытия темы и анализа литературы количество источников, непосредственно относящихся к изучаемой теме. В качестве источников могут выступать публикации в виде книг и статей.

6. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

№ п/п	Виды учебной работы	Образовательные технологии	Всего часов
1	2	3	4
Семестр 3			
1	Тема 1.1 Переход от структурного программирования к ООП. Преимущества ООП для создания сложных, масштабируемых и поддерживаемых приложений.	Технологии развития критического мышления. Обзорная лекция.	2
2	Тема 2.1 Классы и объекты.	Лекция–информация.	2
3	Тема 2.2 Свойства объектов. Инкапсуляция.	Лекция – презентация с использованием Power Point.	2
4	Тема 2.3 Инкапсуляция. Модификаторы доступа	Лекция – презентация с использованием Power Point.	4
5	Тема 3.1 Наследование. Множественное наследование	Лекция – презентация с использованием Power Point.	2
6	Тема 3.2 Переопределение функционала	Лекция – презентация с использованием Power Point.	2
7	Тема 3.3 Атрибуты классов и статические методы	Лекция-информация.	6
8	Тема 3.4 Абстрактные классы	Использование компьютерных технологий при выполнении индивидуальных практических заданий.	6
9	Тема 4.1 Модули	Лекция – презентация с использованием Power Point.	6
10	Тема 4.2 Генерация байткода модулей	Использование компьютерных технологий при выполнении индивидуальных практических заданий	4
19	Лабораторная работа №1 Организация работы с в Visual studio code	Самостоятельная работа обучающихся по индивидуальным заданиям лабораторного практикума с ПЭВМ	2
20	Лабораторная работа № 2 Введение в классы и объекты. Инкапсуляция	Самостоятельная работа обучающихся по индивидуальным заданиям лабораторного практикума с ПЭВМ	4
21	Лабораторная работа №3 Создание простых многооконных приложений под Windows в среде Visual studio code	Самостоятельная работа обучающихся по индивидуальным заданиям лабораторного практикума с ПЭВМ.	4
22	Лабораторная работа №4 Технология отладки и тестирования программ	Самостоятельная работа обучающихся по индивидуальным заданиям лабораторного практикума с ПЭВМ.	4
23	Лабораторная работа № 5 Технология разработки, размещения и использование различных компонентов в TkInter	Самостоятельная работа обучающихся по индивидуальным заданиям лабораторного практикума с ПЭВМ.	4

Итого часов в 3 семестре:			56
Семестр 4			
28	Лабораторная работа №8 Работа с модулями. Установка пакетов	Самостоятельная работа обучающихся по индивидуальным заданиям лабораторного практикума с ПЭВМ.	6
29	Лабораторная работа №9 Работа с файлами в Python	Самостоятельная работа обучающихся по индивидуальным заданиям лабораторного практикума с ПЭВМ.	6
36	Лабораторная работа №10 Модули работы с файлами в Python	Самостоятельная работа обучающихся по индивидуальным заданиям лабораторного практикума с ПЭВМ.	6
37	Лабораторная работа №11 Работа с библиотеками NumPy, SciPy	Использование компьютерных программ. Учебно-исследовательская работа обучающихся.	6
38	Лабораторная работа №12 Визуализация данных. Библиотеки Matplotlib, Seaborn	Использование компьютерных программ. Учебно-исследовательская работа обучающихся.	6
Итого часов в 4 семестре:			30
Всего часов:			86

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

7.1. Перечень основной и дополнительной учебной литературы

Список основной литературы	
1.	Казанский А.А. Объектно-ориентированное программирование на языке Visual Basic 2008 в среде разработки Microsoft Visual Studio 2008 и .NET Framework. Часть 2 : учебное пособие и практикум / Казанский А.А.. — Москва : Московский государственный строительный университет, ЭБС АСВ, 2010. — 104 с. — Текст : электронный // IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/16368.html — Режим доступа: для авторизир. пользователей
2.	Казанский А.А. Объектно-ориентированное программирование на языке Microsoft Visual C# в среде разработки Microsoft Visual Studio 2008 и .NET Framework. 4.3 : учебное пособие и практикум / Казанский А.А.. — Москва : Московский государственный строительный университет, ЭБС АСВ, 2011. — 180 с. — Текст : электронный // IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/19258.html — Режим доступа: для авторизир. пользователей
3.	Николаев Е.И. Объектно-ориентированное программирование : учебное пособие / Николаев Е.И.. — Ставрополь : Северо-Кавказский федеральный университет, 2015. — 225 с. — Текст : электронный // IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/62967.html — Режим доступа: для авторизир. пользователей
4.	Букунов С.В. Основы объектно-ориентированного программирования : учебное пособие / Букунов С.В., Букунова О.В.. — Санкт-Петербург : Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2017. — 196 с. — ISBN 978-5-9227-0713-8. — Текст : электронный // IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/74339.html — Режим доступа: для авторизир. пользователей

5.	Лебедева Т.Н. Теория и практика объектно-ориентированного программирования : учебное пособие / Лебедева Т.Н.. — Челябинск, Саратов : Южно-Уральский институт управления и экономики, Ай Пи Эр Медиа, 2019. — 221 с. — ISBN 978-5-4486-0663-2. — Текст : электронный // IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/81498.html — Режим доступа: для авторизир. пользователей. - DOI: https://doi.org/10.23682/81498
6.	Гуськова О.И. Объектно ориентированное программирование в Java : учебное пособие / Гуськова О.И.. — Москва : Московский педагогический государственный университет, 2018. — 240 с. — ISBN 978-5-4263-0648-6. — Текст : электронный // IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/97750.html — Режим доступа: для авторизир. пользователей
Список дополнительной литературы	
7.	Лисицин Д.В. Объектно-ориентированное программирование : конспект лекций / Лисицин Д.В.. — Новосибирск : Новосибирский государственный технический университет, 2010. — 88 с. — ISBN 978-5-7782-1454-5. — Текст : электронный // IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/44970.html — Режим доступа: для авторизир. пользователей
8.	Николаев Е.И. Объектно-ориентированное программирование. Часть 1 : лабораторный практикум / Николаев Е.И.. — Ставрополь : Северо-Кавказский федеральный университет, 2015. — 183 с. — Текст : электронный // IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/62966.html — Режим доступа: для авторизир. пользователей
9.	Сорокин А.А. Объектно-ориентированное программирование. LAZARUS (Free Pascal) : лабораторный практикум / Сорокин А.А.. — Ставрополь : Северо-Кавказский федеральный университет, 2014. — 216 с. — Текст : электронный // IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/63109.html — Режим доступа: для авторизир. пользователей
10.	Николаев Е.И. Объектно-ориентированное программирование. Часть 2 : лабораторный практикум / Николаев Е.И.. — Ставрополь : Северо-Кавказский федеральный университет, 2015. — 156 с. — Текст : электронный // IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/63218.html — Режим доступа: для авторизир. пользователей
11.	Уйманова Н.А. Основы объектно-ориентированного программирования : практикум / Уйманова Н.А., Таспаева М.Г.. — Оренбург : Оренбургский государственный университет, ЭБС АСВ, 2017. — 156 с. — ISBN 978-5-7410-1993-1. — Текст : электронный // IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/78808.html — Режим доступа: для авторизир. пользователей

7.2 Интернет-ресурсы, справочные системы

1. Электронно-библиотечная система IPRbooks URL: [http:// www.iprbooks.ru/](http://www.iprbooks.ru/) ООО «Ай Пи Эр Медиа»

Лицензионное программное обеспечение	Реквизиты лицензий/ договоров
Антивирус Dr.Web Desktop Security Suite	Лицензионный договор № 621 Срок действия: с 25.09.2025 до 24.09.2026
Консультант Плюс	Договор № 7 от 15.01.2026 г.
Цифровой образовательный ресурс IPR SMART	Лицензионный договор № 12873/25П от 02.07.2025 г. Срок действия: с 01.07.2025 г. до 30.06.2026 г.

ЛИРА	Сублицензионный договор № 2066/А от 21.01.2014 г.
MATLAB	Гос. контракт № 0379100003114000018 от 16 мая 2014 г.
Кодекс	Лицензионное соглашение № 5/4072 от 29.03.2026 г.
Бесплатное ПО	
LibreOffice, OpenOffice, МойОфис, Sumatra PDF, 7-Zip, Adobe Acrobat Reader, Lazarus, Firebird, IBE Expert, VBA, MySQL, Virtual box, Visual Studio Code, StarUML – унифицированный язык моделирования, PostgreSQL. Учебная версия, Project, STDU Viewer	

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1. Требования к аудиториям (помещениям, местам) для проведения занятий

1. Учебная аудитория для проведения занятий лекционного типа:

Специализированная мебель:

Стол преподавательский - 1 шт., стул мягкий - 1 шт., доска меловая - 1 шт., парты - 10 шт., компьютерные столы - 11 шт., стулья - 21 шт.,

Лабораторное оборудование, технические средства обучения, служащие для предоставления учебной информации большой аудитории:

Персональный компьютер – 1 шт.

Экран рулонный настенный – 1 шт.

Проектор – 1 шт.

2. Учебная аудитория для проведения занятий семинарского типа, курсового проектирования (выполнение курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации

Специализированная мебель:

Стол преподавательский - 1 шт., стул мягкий - 1 шт., доска меловая - 1 шт., парты - 10 шт., компьютерные столы - 11 шт., стулья - 21 шт.,

Лабораторное оборудование, технические средства обучения, служащие для предоставления учебной информации большой аудитории:

Персональный компьютер – 1 шт.

Экран рулонный настенный – 1 шт.

Проектор – 1 шт.

3. Помещение для самостоятельной работы

Отдел обслуживания печатными изданиями

Специализированная мебель: Рабочие столы на 1 место – 21 шт. Стулья – 55 шт. Набор демонстрационного оборудования и учебно-наглядных пособий, обеспечивающих тематические иллюстрации: экран настенный – 1 шт.

Проектор – 1 шт. Ноутбук – 1 шт.

Информационно-библиографический отдел.

Специализированная мебель:

Рабочие столы на 1 место - 6 шт. Стулья - 6 шт.

Компьютерная техника с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду ФГБОУ ВО «СевКавГА»:

Персональный компьютер – 1 шт. Сканер – 1 шт. МФУ – 1 шт. Отдел обслуживания электронными изданиями Специализированная мебель:

Рабочие столы на 1 место – 24 шт. Стулья – 24 шт.

Набор демонстрационного оборудования и учебно-наглядных пособий, обеспечивающих тематические иллюстрации:

Интерактивная система - 1 шт. Монитор – 21 шт. Сетевой терминал -18 шт. Персональный компьютер -3 шт. МФУ – 2 шт. Принтер –1шт.

4. Помещение для хранения и профилактического обслуживания учебного оборудования

Специализированная мебель: Шкаф – 1 шт., стул -2 шт., кресло компьютерное – 2 шт., стол угловой компьютерный – 2 шт., тумбочки с ключом – 2 шт. Учебное пособие (персональный компьютер в комплекте) – 2 шт.

8.2. Требования к оборудованию рабочих мест преподавателя и обучающихся

Рабочее место преподавателя, оснащенное компьютером с доступом в Интернет.

Рабочие места обучающихся, оснащенные компьютерами с доступом в Интернет, предназначенные для работы в электронной образовательной среде, и т.п.

8.3. Требования к специализированному оборудованию нет

9. ОСОБЕННОСТИ РЕАЛИЗАЦИИ ДИСЦИПЛИНЫ ДЛЯ ИНВАЛИДОВ И ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Для обеспечения образования инвалидов и обучающихся с ограниченными возможностями здоровья разрабатывается (в случае необходимости) адаптированная образовательная программа, индивидуальный учебный план с учетом особенностей их психофизического развития и состояния здоровья, в частности применяется индивидуальный подход к освоению дисциплины, индивидуальные задания: рефераты, письменные работы и, наоборот, только устные ответы и диалоги, индивидуальные консультации, использование диктофона и других записывающих средств для воспроизведения лекционного и семинарского материала.

В целях обеспечения обучающихся инвалидов и лиц с ограниченными возможностями здоровья комплектуется фонд основной учебной литературой, адаптированной к ограничению электронных образовательных ресурсов, доступ к которым организован в БИЦ Академии. В библиотеке проводятся индивидуальные консультации для данной категории пользователей, оказывается помощь в регистрации и использовании сетевых и локальных электронных образовательных ресурсов, предоставляются места в читальном зале.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ПО ДИСЦИПЛИНЕ Объектно – ориентированное программирование

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

«Объектно – ориентированное программирование»

Компетенции, формируемые в процессе изучения дисциплины

Индекс	Формулировка компетенции
ПК-3	Способен осуществлять программную реализацию алгоритмов при разработке прикладных информационных систем

2. Этапы формирования компетенции в процессе освоения дисциплины

Основными этапами формирования указанных компетенций при изучении обучающимися дисциплины являются последовательное изучение содержательно связанных между собой разделов (тем) учебных занятий. Изучение каждого раздела (темы) предполагает овладение обучающимися необходимыми компетенциями. Результат аттестации обучающихся на различных этапах формирования компетенций показывает уровень освоения компетенций обучающимися.

Этапность формирования компетенций прямо связана с местом дисциплины в образовательной программе.

Разделы (темы) дисциплины	Формируемы компетенции (коды)
	ПК-3
Раздел 1. Введение в объектно-ориентированное программирование.	+
Раздел 2. Классы и объекты. Атрибуты и методы. Инкапсуляция. Модификаторы доступа.	+
Раздел 3. Наследование. Создание иерархий классов. Статические методы. Абстрактные классы.	+
Раздел 4. Модули и пакеты. Установка пакетов.	+
Раздел 5. Работа с файлами. Модули работы с файлами	+
Раздел 6. Различные библиотеки Python	+

3. Показатели, критерии и средства оценивания компетенций, формируемых в процессе изучения дисциплины

ПК-3 Способен осуществлять программную реализацию алгоритмов при разработке прикладных информационных систем

Индикаторы достижения компетенций	Критерии оценивания результатов обучения				Средства оценивания результатов обучения	
	Неудовл.	Удовлет.	Хорошо	Отлично	Текущий контроль	Промежуточная аттестация
ПК-3.1 Обладает знаниями в области программирования и современных компьютерных технологий при разработке прикладных информационных систем. Способен осуществлять оптимизацию алгоритмов при создании прикладных программ или прикладных комплексов	Не обладает знаниями в области программирования и современных компьютерных технологий при разработке прикладных информационных систем. Способен осуществлять оптимизацию алгоритмов при создании прикладных программ или прикладных комплексов	Обладает частичными знаниями в области программирования и современных компьютерных технологий при разработке прикладных информационных систем. Способен осуществлять оптимизацию алгоритмов при создании прикладных программ или прикладных комплексов	Знает методы и способы программирования и современных компьютерных технологий при разработке прикладных информационных систем. Способен осуществлять оптимизацию алгоритмов при создании прикладных программ или прикладных комплексов	Знает на отлично методы и способы программирования и современных компьютерных технологий при разработке прикладных информационных систем. Способен осуществлять оптимизацию алгоритмов при создании прикладных программ или прикладных комплексов	Коллоквиум, контрольные вопросы, тестирование. Реферат	Зачет , экзамен
ПК-3.2 Анализирует стандартные технические требования к проектированию программного обеспечения.	Не умеет анализировать стандартные технические требования к проектированию программного обеспечения.	Умеет частично анализировать стандартные технические требования к проектированию программного обеспечения.	Умеет с неточностями анализировать стандартные технические требования к проектированию программного обеспечения	Может на отлично анализировать стандартные технические требования к проектированию программного обеспечения	Коллоквиум, контрольные вопросы, тестирование. Реферат	Зачет , экзамен
ПК-3.3 Способен	Не может		Не уверено может		Коллоквиум,	Зачет , экзамен

<p>осуществлять разработку технических требований к программному обеспечению, при его создании</p>	<p>осуществлять разработку технических требований к программному обеспечению, при его создании</p>	<p>Частично может осуществить разработку технических требований к программному обеспечению, при его создании</p>	<p>осуществлять разработку технических требований к программному обеспечению, при его создании</p>	<p>Может осуществлять разработку технических требований к программному обеспечению, при его создании</p>	<p>контрольные вопросы, тестирование, индивидуальные задания к лабораторным работам.</p>	
----------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------	--

4. Комплект контрольно-оценочных средств по дисциплине

Вопросы к зачету

по дисциплине «Объектно – ориентированное программирование»

1. Понятие ООП. Назовите и кратко охарактеризуйте четыре основных принципа.
2. Класс и Объект (Экземпляр). В чем разница? Как создать простой класс и его экземпляр?
3. Конструктор и инициализатор `__init__`. Чем он отличается от `__new__`
4. Атрибуты класса и атрибуты экземпляра. В чем разница. Продемонстрируйте на примере.
5. Методы экземпляра `self`. Для чего он нужен.
6. Модификаторы доступа в Python. Как реализованы `public`, `protected` (`_`) и `private` (`__`) атрибуты и методы.
7. Инкапсуляция. Как она реализуется в Python с помощью свойств
8. Наследование. Как создать дочерний класс. Что такое метод `super()` и для чего он используется.
9. Полиморфизм. Объясните на примере методов. Что такое переопределение методов (`method overriding`).
10. Абстракция. Абстрактный класс? Как создать абстрактный метод с помощью модуля `abc`?
11. Дать понятие «магические» методы. Приведите примеры самых распространенных.
12. Композиция и Агрегация. В чем разница между этими двумя типами отношений "has-a".
12. Классовые и статические методы. Декораторы `@classmethod` и `@staticmethod`.
13. Множественное наследование и MRO (Method Resolution Order).
14. Переменные только для чтения. Как создать свойство только для чтения с помощью декоратора `@property`?
15. Что такое "слоты" (`__slots__`)? Для чего их используют? Каковы преимущества и недостатки?

Вопросы к экзамену

по дисциплине «Объектно – ориентированное программирование»

1. Предпосылки появления и развития объектно-ориентированного подхода в программировании.
2. Основные особенности ООП.
3. Ключевые понятия ООП.
4. Область действия полей объекта и параметр Self. Метод класса.
5. Понятие наследования и его назначение. Присваивание объектов.
6. Полиморфизм. Статические и виртуальные методы.
7. Внутренняя структура объекта.
8. Конструкторы и деструкторы, их назначение и правила использования.
9. Виртуальные и динамические методы, их назначение и механизмы вызова.
10. Основные подходы к созданию многооконных приложений в Tkinter.
11. Опишите различия между главным окном и дочерними окнами.
12. Модальные и немодальные диалоги.
13. Приведите примеры использования каждого типа.
14. Как создаются модальные окна в Tkinter?
15. Методы управления жизненным циклом окон.
16. Как программно закрыть, скрыть или уничтожить окно?
17. Опишите способы передачи данных между окнами в многооконном приложении.
18. Как организовать центрирование окна относительно главного окна или экрана?
19. Напишите код для центрирования дочернего окна.
20. Какие шаблоны проектирования рекомендуется использовать при разработке многооконных приложений?
21. Почему важно разделять логику и интерфейс?
22. Перечислите основные модули для работы со стандартными диалогами в Tkinter.
23. Какие типы диалогов предоставляет каждый модуль?
24. Как настроить фильтры файлов в диалогах открытия и сохранения?
25. Приведите пример настройки для текстовых и графических файлов.
26. Какие методы модуля messagebox используются для вывода информации и получения ответа от пользователя?
27. В чем разница между askyesno() и askquestion()?
28. Как работает диалог выбора цвета?
29. Какие данные возвращает функция askcolor()?
30. Как использовать выбранный цвет для изменения элементов интерфейса?
31. Для каких задач предназначен модуль simpledialog? Какие типы ввода данных он поддерживает?
32. Как обработать некорректный ввод пользователя?
33. Как создать собственное диалоговое окно для специфических задач?
34. Какие элементы управления следует включать в пользовательские диалоги?

Перечень практический заданий к экзамену

Требования:

Многооконный графический интерфейс

Не менее 3-5 полей (характеристик объекта).

Конструктор по умолчанию и конструктор с параметрами.

Не менее 2-3 методов, описывающих поведение объекта.

Создание нескольких объектов в main и демонстрация работы методов.

Индивидуальные задания:

1. Book (название, автор, год, методы: показать информацию, издать книгу).
2. Car (марка, модель, год, цвет, пробег, методы: поехать, покрасить).
3. Student (имя, фамилия, ID, группа, средний балл, методы: перевести в новую группу).
4. Fraction (дробь: числитель, знаменатель, методы: сложение, вычитание, вывод на экран).
5. Sotrudnik (имя, фамилия, должность, оклад, стаж: минимальный стаж, минимальный оклад)
6. Dekanat (название, количество студентов, количество преподавателей: подсчет отличников, средний балл по институту)
7. Programmist (имя, фамилия, уровень, место работы, стаж: расчет заработной платы, обучение)
8. Dog (кличка, порода, возраст, вес, фио хозяина: расчет нормы кормления, оповещение о прививках)
9. Uchenik (имя, фамилия, класс, оценки: средний балл оценок, количество пропусков)
10. Figura (название, длины сторон, цвет закраски: периметр, площадь)

СЕВЕРО-КАВКАЗСКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ

Кафедра Математика

20_ - 20_ учебный год

Экзаменационный билет № 1

по дисциплине «Объектно – ориентированное программирование»
для обучающихся направления подготовки 01.03.02 Прикладная математика
и информатика

1. Методы управления жизненным циклом окон.
2. Опишите различия между главным окном и дочерними окнами.
3. Многооконный графический интерфейс
Не менее 3-5 полей (характеристик объекта).
Конструктор по умолчанию и конструктор с параметрами.
Не менее 2-3 методов, описывающих поведение объекта.
Создание нескольких объектов в main и демонстрация работы методов.
Book (название, автор, год, методы: показать информацию, издать книгу).

Зав. кафедрой «Математика»

Кочкаров А.М.

Контрольные вопросы

по дисциплине «Объектно – ориентированное программирование»

Вопросы к разделу 1.

1. Что такое объектно-ориентированное программирование (ООП)?
2. Назовите и кратко охарактеризуйте четыре основных принципа.
3. Класс и Объект (Экземпляр). В чем разница? Как создать простой класс и его экземпляр?
4. Какие преимущества дает использование ООП?
5. Что такое атрибуты и методы?
6. Что такое конструктор и деструктор?
7. Как реализуется инкапсуляция в Python?
8. Конструктор и инициализатор `__init__`. Чем он отличается от `__new__`

9. Атрибуты класса и атрибуты экземпляра. В чем разница. Продемонстрируйте на примере.
10. Методы экземпляра `self`. Для чего он нужен.
11. Модификаторы доступа в Python. Как реализованы `public`, `protected` (`_`) и `private` (`__`) атрибуты и методы.

Вопросы к разделу 2.

1. Что такое класс в объектно-ориентированном программировании?
2. Чем объект отличается от класса?
3. Что такое атрибуты и методы класса?
4. Что такое конструктор класса и для чего он используется?
5. Что такое инкапсуляция и какие преимущества она дает?
6. Какие модификаторы доступа существуют и чем они отличаются?
7. Что такое свойства (`property`) и для чего они используются?
8. Что такое статические методы и методы класса?

Вопросы к разделу 3.

1. Что такое наследование в ООП и какие преимущества оно дает?
2. В чем разница между переопределением и перегрузкой методов?
3. Что такое абстрактный класс и абстрактный метод?
4. Какие особенности статических методов и полей?
5. Что такое цепочка наследования и множественное наследование?
6. Что такое конструкторы в иерархии наследования?

Вопросы к разделу 4.

1. Конструкторы и деструкторы, их назначение и правила использования.
2. Виртуальные и динамические методы, их назначение и механизмы вызова.
3. Основные подходы к созданию многооконных приложений в Tkinter.
4. Опишите различия между главным окном и дочерними окнами.
5. Модальные и немодальные диалоги.
6. Приведите примеры использования каждого типа.
7. Как создаются модальные окна в Tkinter?

Вопросы к разделу 5.

1. Какие основные режимы открытия файлов существуют в Python и чем они отличаются?
2. Почему рекомендуется использовать конструкцию `with open() as file:`?
3. Как определить кодировку текстового файла перед его чтением?
4. В чем разница между методами `read()`, `readline()` и `readlines()`?
5. Как организовать обработку исключений при работе с файлами?
6. Какие модули стандартной библиотеки Python используются для работы с JSON и CSV?
7. Как рекурсивно обойти все файлы в директории и поддиректориях?
8. В чем разница между абсолютным и относительным путем к файлу?
9. Как безопасно удалить файл с проверкой его существования?
10. Какие методы используются для позиционирования внутри файла?

Вопросы к разделу 6.

1. Какие преимущества дает использование NumPy вместо стандартных списков Python?
2. Как эффективно работать с большими массивами данных?
3. В каких случаях следует использовать разреженные матрицы?

4. Как выбрать подходящий метод оптимизации для конкретной задачи?
5. Какие методы интерполяции наиболее точны для различных типов данных?
6. Что такое ndarray и чем он отличается от обычного списка Python?
7. Как создать массив размерности 3×4, заполненный случайными числами?
8. Что такое вещание (broadcasting) в NumPy?
9. Как выполнить поэлементное умножение двух массивов?
10. Какие методы есть для сохранения и загрузки массивов?
11. Каковы основные различия между NumPy и SciPy?
12. Как решить систему линейных уравнений с помощью SciPy?
13. Какие методы численного интегрирования предоставляет SciPy?
14. Как выполнить интерполяцию данных с помощью SciPy?
15. Какие статистические тесты доступны в SciPy?
16. Что такое Figure и Axes в Matplotlib?
17. Как создать несколько subplot на одном рисунке?
18. Какие основные параметры управляют внешним видом линии на графике?
19. Как добавить легенду к графику?
20. Как сохранить график в файл?
21. Чем отличается bar от barh?
22. Как изменить размер шрифта на осях?
23. Что такое tight_layout() и для чего он используется?

Вопросы для коллоквиумов

по дисциплине «Объектно – ориентированное программирование»

Раздел 1.

12. Что такое объектно-ориентированное программирование (ООП)?
13. Назовите и кратко охарактеризуйте четыре основных принципа.
14. Класс и Объект (Экземпляр). В чем разница? Как создать простой класс и его экземпляр?
15. Какие преимущества дает использование ООП?
16. Что такое атрибуты и методы?
17. Что такое конструктор и деструктор?
18. Как реализуется инкапсуляция в Python?
19. Конструктор и инициализатор __init__. Чем он отличается от __new__
20. Атрибуты класса и атрибуты экземпляра. В чем разница. Продемонстрируйте на примере.
21. Методы экземпляра self. Для чего он нужен.
22. Модификаторы доступа в Python. Как реализованы public, protected (__) и private (__) атрибуты и методы.

Раздел 2.

1. Дать понятие «магические» методы. Приведите примеры самых распространенных.
2. Композиция и Агрегация. В чем разница между этими двумя типами отношений "has-a".
3. Классовые и статические методы. Декораторы @classmethod и @staticmethod.

4. Множественное наследование и MRO (Method Resolution Order).
5. Переменные только для чтения. Как создать свойство только для чтения с помощью декоратора `@property`?
6. Что такое "слоты" (`__slots__`)? Для чего их используют? Каковы преимущества и недостатки?

Раздел 3.

1. Предпосылки появления и развития объектно-ориентированного подхода в программировании.
2. Основные особенности ООП.
3. Ключевые понятия ООП.
4. Область действия полей объекта и параметр `Self`. Метод класса.
5. Понятие наследования и его назначение. Присваивание объектов.
6. Полиморфизм. Статические и виртуальные методы.
7. Внутренняя структура объекта.
8. Методы управления жизненным циклом окон.
9. В чем разница между `askyesno()` и `askquestion()`?
10. Как работает диалог выбора цвета?
11. Какие данные возвращает функция `askcolor()`?
12. Как использовать выбранный цвет для изменения элементов интерфейса?
13. Для каких задач предназначен модуль `simpledialog`? Какие типы ввода данных он поддерживает?
14. Как обработать некорректный ввод пользователя?
15. Как создать собственное диалоговое окно для специфических задач?
16. Какие элементы управления следует включать в пользовательские диалоги

Разделу 4.

1. Конструкторы и деструкторы, их назначение и правила использования.
2. Виртуальные и динамические методы, их назначение и механизмы вызова.
3. Основные подходы к созданию многооконных приложений в Tkinter.
4. Опишите различия между главным окном и дочерними окнами.
5. Модальные и немодальные диалоги.
6. Приведите примеры использования каждого типа.
7. Как создаются модальные окна в Tkinter?

Раздел 5.

1. Как программно закрыть, скрыть или уничтожить окно?
2. Опишите способы передачи данных между окнами в многооконном приложении.
3. Как организовать центрирование окна относительно главного окна или экрана?
4. Напишите код для центрирования дочернего окна.
5. Какие шаблоны проектирования рекомендуется использовать при разработке многооконных приложений?
6. Почему важно разделять логику и интерфейс?
7. Перечислите основные модули для работы со стандартными диалогами в Tkinter.
8. Какие типы диалогов предоставляет каждый модуль?
9. Как настроить фильтры файлов в диалогах открытия и сохранения?
10. Приведите пример настройки для текстовых и графических файлов.
11. Какие методы модуля `messagebox` используются для вывода информации и получения ответа от пользователя?

Раздел 6.

1. Какие методы интерполяции наиболее точны для различных типов данных?
2. Что такое ndarray и чем он отличается от обычного списка Python?
3. Как создать массив размерности 3×4 , заполненный случайными числами?
4. Что такое вещание (broadcasting) в NumPy?
5. Как выполнить поэлементное умножение двух массивов?
6. Какие методы есть для сохранения и загрузки массивов?
7. Каковы основные различия между NumPy и SciPy?
8. Как решить систему линейных уравнений с помощью SciPy?
9. Какие методы численного интегрирования предоставляет SciPy?
10. Как выполнить интерполяцию данных с помощью SciPy?
11. Какие статистические тесты доступны в SciPy?
12. Что такое Figure и Axes в Matplotlib?
13. Как создать несколько subplot на одном рисунке?
14. Какие основные параметры управляют внешним видом линии на графике?

Комплект заданий для лабораторной работы

по дисциплине «Объектно – ориентированное программирование»

Лабораторная работа №1 Введение в классы и объекты. Инкапсуляция.

Разработать программное приложение по следующим требованиям:

Многооконный графический интерфейс

Не менее 3-5 полей (характеристик объекта).

Конструктор по умолчанию и конструктор с параметрами.

Не менее 2-3 методов, описывающих поведение объекта.

Создание нескольких объектов в main и демонстрация работы методов.

Индивидуальные задания:

1. Book (название, автор, год, методы: показать информацию, издать книгу).
2. Car (марка, модель, год, цвет, пробег, методы: поехать, покрасить).
3. Student (имя, фамилия, ID, группа, средний балл, методы: перевести в новую группу).
4. Fraction (дробь: числитель, знаменатель, методы: сложение, вычитание, вывод на экран).
5. Sotrudnik (имя, фамилия, должность, оклад, стаж: минимальный стаж, минимальный оклад)
6. Dekanat (название, количество студентов, количество преподавателей: подсчет отличников, средний балл по институту)
7. Programmist (имя, фамилия, уровень, место работы, стаж: расчет заработной платы, обучение)
8. Dog (кличка, порода, возраст, вес, фио хозяина: расчет нормы кормления, оповещение о прививках)
9. Uchenik (имя, фамилия, класс, оценки: средний балл оценок, количество пропусков)
10. Figura (название, длины сторон, цвет закраски: периметр, площадь)

Лабораторная работа №2 Создание простых многооконных приложений под Windows в среде Visual studio code.

Индивидуальные задания

Задание 1: "Конвертер валют"

Создайте приложение с двумя окнами:

- Главное окно: поле ввода суммы, выбор валюты из списка, кнопка "Конвертировать"
- Второе окно: отображает результат конвертации по фиксированному курсу
- Добавьте кнопку "История конвертаций", открывающую третье окно с таблицей предыдущих операций

Задание 2: "Редактор текста"

Реализуйте текстовый редактор:

- Главное окно: текстовое поле с возможностью редактирования
- Меню "Файл" с пунктами: "Новый", "Открыть", "Сохранить", "Сохранить как"
- Окно "О программе" с информацией об авторе
- Окно "Настройки" для изменения шрифта и цвета текста

Задание 3: "Калькулятор ИМТ"

Разработайте приложение для расчета индекса массы тела:

- Первое окно: ввод роста (см) и веса (кг), кнопка расчета
- Второе окно: графическое отображение результата с цветовой индикацией (недостаток/норма/избыток)
- Третье окно: рекомендации по результату расчета

Задание 4: "Планировщик задач"

Создайте планировщик с тремя окнами:

1. Список задач с чекбоксами выполнено/не выполнено
 2. Окно добавления/редактирования задачи
 3. Окно статистики: выполнено/невыполнено/просрочено
- Используйте Treeview для отображения списка

Задание 5: "Галерея изображений"

Приложение для просмотра изображений:

- Главное окно: список файлов изображений в папке
- Второе окно: полноразмерный просмотр выбранного изображения
- Третье окно: информация о файле (размер, разрешение, дата создания)
- Реализуйте кнопки "Следующее", "Предыдущее"

Задание 6: "Анкета пользователя"

Многооконное приложение для сбора данных:

1. Окно 1: Основная информация (ФИО, возраст)
 2. Окно 2: Контактные данные
 3. Окно 3: Дополнительная информация
 4. Окно 4: Просмотр и подтверждение всех введенных данных
- Добавьте навигацию "Назад"/"Вперед"

Задание 7: "Игра "Угадай число"

Игровое приложение:

- Главное окно: ввод предполагаемого числа, кнопка "Проверить"
- Второе окно: статистика игры (количество попыток, история догадок)
- Третье окно: настройки (диапазон чисел, максимальное количество попыток)
- Информационные окна при угадывании/проигрыше

Задание 8: "Погодное приложение"

Приложение для отображения погоды:

- Главное окно: текущая погода (температура, влажность, давление)
- Второе окно: прогноз на 3 дня в виде таблицы
- Третье окно: настройки (выбор города, единицы измерения)
- Используйте иконки для отображения погодных условий

Задание 9: "Шифровальщик текста"

Приложение для шифрования:

- Главное окно: ввод текста, выбор алгоритма (Цезарь, XOR)
- Второе окно: настройки параметров шифрования (сдвиг, ключ)
- Третье окно: история операций
- Кнопки "Зашифровать", "Расшифровать", "Сохранить в файл"

Задание 10: "Система тестирования"

Приложение для проведения тестов:

1. Окно регистрации пользователя
 2. Окно выбора теста
 3. Окно прохождения теста (вопрос + варианты ответов)
 4. Окно результатов
 5. Окно статистики по всем попыткам
- Используйте JSON файлы для хранения вопросов и результатов

Лабораторная работа №4 Технология отладки и тестирования программ

Индивидуальные задания

Общее требование: Для каждой программы создать не менее 5 тестовых случаев, использовать отладчик для поиска и устранения ошибок.

- 1. Калькулятор дробей**
Реализовать калькулятор для арифметических операций с дробями. Обеспечить сокращение дробей и обработку деления на ноль.
- 2. Поиск в массиве**
Реализовать функции линейного и бинарного поиска. Добавить обработку некорректных входных данных.
- 3. Валидатор дат**
Написать программу для проверки корректности даты в формате ДД.ММ.ГГГГ с учетом високосных лет.
- 4. Шифр Цезаря**
Реализовать шифрование и дешифрование текста с заданным сдвигом. Обеспечить работу с кириллицей и латиницей.
- 5. Работа с CSV-файлами**
Создать программу для чтения, фильтрации и записи данных в CSV-файл. Обработать возможные ошибки ввода-вывода.
- 6. Стек с обработкой исключений**
Реализовать структуру данных "стек" с методами push, pop, peek. Добавить пользовательские исключения для пустого стека.
- 7. Конвертер температур**
Реализовать конвертацию между шкалами Цельсия, Фаренгейта и Кельвина. Добавить проверку на абсолютный ноль.
- 8. Поиск палиндромов**
Написать функцию для проверки строки на палиндром (игнорируя регистр и пробелы). Протестировать на различных входных данных.
- 9. Генератор паролей**
Создать программу генерации безопасных паролей с заданной длиной и набором символов. Добавить тесты для проверки корректности.
- 10. Упрощение дроби**
Реализовать функцию сокращения дроби, представленной числителем и знаменателем. Добавить обработку отрицательных чисел и нулевого знаменателя.

Лабораторная работа №5 Технология разработки, размещения и использование различных компонентов в TkInter

Индивидуальные задания:

- 1. Калькулятор** – создайте простой калькулятор с кнопками цифр и операций.
- 2. Текстовый редактор** – поле Text с кнопками "Открыть", "Сохранить".
- 3. Список дел** – добавление, удаление задач в Listbox.
- 4. Конвертер валют** – поля ввода, выбор валют через Combobox.
- 5. Настройки шрифта** – изменение шрифта текста через Scale и Radiobutton.
- 6. График функции** – ввод формулы и отрисовка с помощью matplotlib (интеграция с TkInter).
- 7. Игра "Угадай число"** – с подсказками "больше/меньше".
- 8. Анкета** – форма с полями ввода, переключателями и кнопкой отправки.
- 9. Часы** – цифровые часы с обновлением через after().
- 10. Палитра цветов** – выбор цвета через Scale для RGB и отображение.

Лабораторная работа №6 Наследование и полиморфизм.

Индивидуальные задания

Общее условие:

Создайте иерархию классов согласно вашему варианту. Реализуйте полиморфные методы, продемонстрируйте работу системы.

Варианты заданий:

Вариант 1: Система учета сотрудников компании

- Базовый класс: Employee
- Подклассы: Manager, Developer, Designer
- Полиморфный метод: calculate_salary()

Вариант 2: Библиотека медиафайлов

- Базовый класс: MediaFile
- Подклассы: AudioFile, VideoFile, ImageFile
- Полиморфный метод: play(), get_info()

Вариант 3: Система банковских счетов

- Базовый класс: BankAccount
- Подклассы: SavingsAccount, CheckingAccount, CreditAccount
- Полиморфный метод: calculate_interest(), withdraw()

Вариант 4: Транспортные средства

- Базовый класс: Vehicle
- Подклассы: Car, Motorcycle, Bicycle, Truck
- Полиморфный метод: start_engine(), fuel_consumption()

Вариант 5: Геометрические тела

- Базовый класс: Solid
- Подклассы: Cube, Sphere, Cylinder, Pyramid
- Полиморфный метод: volume(), surface_area()

Вариант 6: Система заказов в ресторане

- Базовый класс: MenuItem
- Подклассы: Appetizer, MainCourse, Dessert, Beverage
- Полиморфный метод: prepare(), calculate_cost()

Вариант 7: Иерархия животных

- Базовый класс: Animal
- Подклассы: Mammal, Bird, Fish, Reptile
- Полиморфный метод: move(), eat()

Вариант 8: Система учета учебных курсов

- Базовый класс: Course
- Подклассы: LectureCourse, LabCourse, SeminarCourse, OnlineCourse
- Полиморфный метод: conduct_class(), grade_student()

Вариант 9: Игровые персонажи

- Базовый класс: GameCharacter
- Подклассы: Warrior, Mage, Archer, Healer
- Полиморфный метод: attack(), defend(), use_special_ability()

Вариант 10: Система учета недвижимости

- Базовый класс: Property
- Подклассы: Apartment, House, CommercialSpace, LandPlot
- Полиморфный метод: calculate_tax(), get_description()

Лабораторная работа №7 Абстрактные классы и интерфейсы

Индивидуальные задания

1. **Система фигур:** Создайте абстрактный класс `Shape` с методами `area()` и `perimeter()`. Реализуйте классы `Circle`, `Triangle`, `Square`. Добавьте метод `display_info()`, который выводит информацию о фигуре.
2. **Банковские счета:** Создайте абстрактный класс `BankAccount` с методами `deposit()`, `withdraw()`, `get_balance()`. Реализуйте классы `SavingsAccount` (нельзя снимать больше, чем есть) и `CreditAccount` (можно уходить в минус до лимита).
3. **Животные:** Создайте интерфейс `Animal` с методами `make_sound()` и `move()`. Реализуйте классы `Dog`, `Cat`, `Bird`. Добавьте класс `Zoo`, который может содержать животных и выводить информацию о них.
4. **Уведомления:** Создайте абстрактный класс `Notification` с методом `send()`. Реализуйте классы `EmailNotification`, `SMSNotification`, `PushNotification`. Добавьте возможность отправки разных типов уведомлений.
5. **Геометрические тела:** Создайте абстрактный класс `Solid` с методами `volume()` и `surface_area()`. Реализуйте классы `Cube`, `Sphere`, `Cylinder`.
6. **Транспортные средства:** Создайте интерфейс `Vehicle` с методами `start()`, `stop()`, `fuel_type()`. Реализуйте классы `Car`, `Bicycle`, `Motorcycle`. Учтите, что велосипед не использует топливо.
7. **Сотрудники компании:** Создайте абстрактный класс `Employee` с методами `calculate_salary()` и `get_role()`. Реализуйте классы `Manager`, `Developer`, `Intern`. Для каждого типа сотрудника своя логика расчета зарплаты.
8. **Медиаплеер:** Создайте интерфейс `MediaPlayer` с методами `play()`, `pause()`, `stop()`. Реализуйте классы `AudioPlayer`, `VideoPlayer`. Добавьте специфические методы для каждого типа.
9. **Фильтры изображений:** Создайте абстрактный класс `ImageFilter` с методом `apply()`. Реализуйте классы `BlurFilter`, `SharpenFilter`, `GrayscaleFilter`. Создайте конвейер обработки изображений.
10. **Платежные системы:** Создайте интерфейс `PaymentSystem` с методами `process_payment()` и `refund()`. Реализуйте классы `CreditCardPayment`, `PayPalPayment`, `CryptoPayment`. Добавьте обработку ошибок при оплате.

Рекомендации по выполнению

1. Для каждого задания создайте отдельный файл.
2. Сначала проектируйте иерархию классов на бумаге.
3. Используйте аннотации типов для улучшения читаемости кода.
4. Проверяйте работу каждого класса с помощью unit-тестов.
5. Документируйте код с помощью docstrings.

Лабораторная работа №8 Работа с модулями. Установка пакетов

Индивидуальные задания

Задание 1

Создайте модуль `geometry.py` с функциями для вычисления:

- Площади круга
- Площади прямоугольника
- Площади треугольника по формуле Герона

Задание 2

Напишите программу, которая использует модуль `datetime` для:

- Вывода текущей даты и времени
- Определения дня недели
- Вычисления разницы между двумя датами

Задание 3

Создайте пакет `statistics` с модулями:

- `descriptive.py` (среднее, медиана, мода)
- `probability.py` (факториал, сочетания)
- `tests.py` (тестирование функций)

Задание 4

Установите библиотеку `requests` с помощью `pip` и напишите программу, которая:

- Отправляет GET-запрос к публичному API
- Обрабатывает ответ в формате JSON
- Выводит определенные данные из ответа

Задание 5

Создайте виртуальное окружение и установите в него:

- Библиотеку для работы с данными (`pandas`)
- Библиотеку для визуализации (`matplotlib`)
- Создайте файл `requirements.txt` с зависимостями

Задание 6

Напишите модуль для шифрования/дешифрования текста с использованием:

- Шифра Цезаря
- Азбуки Морзе
- Простого заменного шифра

Задание 7

Создайте модуль `file_operations.py` с функциями для:

- Чтения и записи текстовых файлов
- Поиска строк по шаблону
- Подсчета частоты слов в тексте

Задание 8

Используя модуль `random`, создайте программу для:

- Генерации случайных паролей заданной длины
- Симуляции броска игральных костей
- Лотереи (выбор случайных победителей из списка)

Задание 9

Установите библиотеку `pillow` и создайте программу для:

- Изменения размера изображений
- Конвертации между форматами
- Наложения водяных знаков

Задание 10

Создайте собственный пакет `myutilities` с модулями:

- `validation.py` (валидация email, пароля, номера телефона)
- `conversion.py` (конвертация валют, единиц измерения)
- `helpers.py` (различные вспомогательные функции)

Лабораторная работа № 9 Работа с файлами в Python

Индивидуальные задания

Задание 1: Анализ текстового файла

Создайте программу, которая:

- Считывает текст из файла input.txt
- Подсчитывает количество строк, слов и символов
- Записывает статистику в файл report.txt

Задание 2: Фильтрация данных

Создайте программу, которая:

- Читает файл data.txt с числами (по одному в строке)
- Записывает в even.txt только четные числа
- Записывает в odd.txt только нечетные числа

Задание 3: Объединение файлов

Напишите программу, которая:

- Берет 3 файла (part1.txt, part2.txt, part3.txt)
- Объединяет их содержимое
- Сохраняет результат в full_text.txt

Задание 4: Поиск в файле

Создайте программу для поиска:

- Пользователь вводит слово для поиска
- Программа ищет это слово в text.txt
- Выводит номера строк, содержащих это слово

Задание 5: Шифрование файла

Реализуйте программу:

- Читает secret.txt
- Шифрует текст (например, сдвиг букв на +1)
- Сохраняет зашифрованный текст в encrypted.txt

Задание 6: Лог-анализатор

Напишите программу:

- Анализирует файл server.log
- Находит все строки с ошибками (содержащие "ERROR")
- Сохраняет их в errors.log

Задание 7: CSV обработчик

Создайте программу для работы с CSV:

- Читает students.csv (имя, оценка, группа)
- Вычисляет среднюю оценку по группам
- Записывает результат в statistics.csv

Задание 8: Резервное копирование

Напишите программу:

- Создает копию указанного файла
- Добавляет к имени копии timestamp
- original.txt → original_20241205_1430.txt

Задание 9: Разделитель файла

Создайте программу:

- Делит большой файл big_file.txt на части по N строк

- Создает файлы part_1.txt, part_2.txt, etc.

Задание 10: JSON конвертер

Напишите программу:

- Читает текстовый файл с данными в определенном формате
- Конвертирует в JSON
- Сохраняет как data.json

Лабораторная работа № 10 Модули работы с файлами в Python

Индивидуальные задания

Задание 1: Анализатор текста

Создайте программу, которая:

- Читает текст из файла input.txt
- Подсчитывает количество слов, строк и символов
- Находит самое длинное слово
- Сохраняет статистику в report.txt

Задание 2: Фильтр данных

Дан CSV файл students.csv с колонками: name,age,grade

Отфильтруйте студентов с оценкой выше 4.0

Сохраните результат в honor_students.json

Задание 3: Организатор файлов

Напишите скрипт, который:

- Рекурсивно обходит указанную директорию
- Сортирует файлы по расширениям
- Создает папки для каждого расширения
- Перемещает файлы в соответствующие папки

Задание 4: Лог-анализатор

Проанализируйте файл server.log:

- Найдите все ошибки (строки с "ERROR")
- Подсчитайте количество ошибок по типам
- Экспортируйте отчет в CSV

Задание 5: Резервное копирование

Создайте утилиту бэкапа:

- Копирует указанные файлы в папку backup/YYYY-MM-DD
- Архивирует папку в ZIP
- Удаляет бэкапы старше 7 дней

Задание 6: Конвертер форматов

Конвертируйте JSON файл в CSV и наоборот

Реализуйте поддержку аргументов командной строки

Задание 7: Поиск дубликатов

Найдите дубликаты файлов в директории:

- Сравните файлы по размеру и хешу MD5
- Выведите список дубликатов с путями
- Предложите варианты удаления

Задание 8: Шифрование файлов

Реализуйте простое шифрование XOR:

- Чтение бинарного файла
- Применение ключа шифрования

- Сохранение зашифрованного файла
- Режим дешифровки

Задание 9: Слияние файлов

Объедините несколько текстовых файлов в один:

- Добавляйте название исходного файла как заголовок
- Удаляйте пустые строки
- Сохраняйте порядок файлов

Задание 10: Монитор изменений

Напишите программу, которая:

- Мониторит изменения в указанной папке
- Логирует создание, изменение, удаление файлов
- Сохраняет историю в SQLite базе данных

Лабораторная работа №11 Работа с библиотеками NumPy, SciPy

Индивидуальные задания

Вариант 1: Статистический анализ данных

- Сгенерируйте массив из 1000 случайных чисел с нормальным распределением
- Вычислите среднее, медиану, стандартное отклонение
- Постройте гистограмму (используйте matplotlib)

Вариант 2: Решение уравнений

- Решите нелинейное уравнение: $x^3 - 2x^2 + 4 = 0$
- **Найдите минимум функции $f(x) = x^4 - 3x^2 + x$**

Вариант 3: Работа с полиномами

- Создайте полином 4-й степени с заданными коэффициентами
- Вычислите корни полинома
- Найдите значение производной в точке $x = 2$

Вариант 4: Обработка сигналов

- Создайте сигнал из суммы двух синусоид
- Примените преобразование Фурье
- Отфильтруйте высокочастотные составляющие

Вариант 5: Оптимизация

- Решите задачу линейного программирования
- Минимизируйте функцию с ограничениями
- Используйте метод симплекса

Вариант 6: Численное интегрирование

- Вычислите определенный интеграл от сложной функции
- Сравните различные методы интегрирования
- Оцените погрешность вычислений

Вариант 7: Работа с разреженными матрицами

- Создайте разреженную матрицу 100x100
- Выполните операции с разреженными матрицами
- Сравните производительность с плотными матрицами

Вариант 8: Интерполяция данных

- Создайте набор экспериментальных данных с шумом
- Выполните линейную и кубическую интерполяцию
- Сравните результаты с исходной функцией

Вариант 9: Специальные функции

- Исследуйте функции Бесселя различных порядков
- Постройте графики гамма-функции
- Вычислите значения интегральной экспоненты

Вариант 10: Работа с изображениями

- Загрузите изображение (используйте `scipy.misc`)
- Примените различные фильтры (размытие, детекция краев)
- Измените размер и ориентацию изображения

Лабораторная работа №12 Визуализация данных. Библиотеки Matplotlib, Seaborn

Индивидуальные задания

Каждому студенту предлагается выполнить одно задание из списка ниже. Данные можно сгенерировать случайно или использовать встроенные датасеты (например, `iris`, `tips`, `titanic` из Seaborn).

1. **Линейные графики**
Постройте 3 линейных графика на одной `figure`, иллюстрирующих изменение функций $\sin(x)$, $\cos(x)$ и $\sin(x)*\cos(x)$ на интервале $[0, 4\pi]$. Добавьте легенду, названия осей и заголовков.
2. **Столбчатые диаграммы**
Создайте столбчатую диаграмму для 5 категорий с случайными значениями. Добавьте подписи значений над столбцами и измените цветовую палитру.
3. **Круговая диаграмма**
Постройте круговую диаграмму для отображения долей (например, распределение бюджета). Выделите один сегмент, добавьте проценты с округлением до целых.
4. **Гистограмма**
Сгенерируйте 1000 случайных чисел с нормальным распределением и постройте гистограмму с 20 интервалами. Добавьте кривую плотности.
5. **Диаграмма рассеяния**
Создайте `scatter plot` для двух наборов случайных данных (500 точек). Настройте размер, цвет и прозрачность точек. Добавьте линию тренда.
6. **Boxplot**
Постройте `boxplot` для 4 наборов случайных данных с разными распределениями. Подпишите группы и добавьте заголовок.
7. **Heatmap**
Создайте тепловую карту для корреляционной матрицы датасета `iris`. Добавьте аннотации (значения коэффициентов).
8. **Violin plot**
Используя датасет `tips`, постройте `violin plot` для распределения суммы счёта (`total_bill`) по дням недели (`day`).
9. **Pairplot**
Для датасета `penguins` постройте `pairplot`, раскрасив точки по виду пингвинов. Уберите верхние треугольные графики.
10. **Jointplot**
Создайте `jointplot` для двух переменных из датасета `mpg` (например, `mpg` и `horsepower`). Добавьте регрессионную линию и гистограммы распределений.

Комплект тестовых вопросов

по дисциплине «Объектно – ориентированное программирование»

Раздел 1,2

1. Какой из перечисленных принципов НЕ является принципом ООП?

- A) Инкапсуляция
- B) Наследование
- C) Рекурсия
- D) Полиморфизм

2. Что такое объект в ООП? _____

3. Какой метод вызывается при создании объекта в Python?

- A) `__new__()`
- B) `__init__()`
- C) `__create__()`
- D) `__start__()`

4. Как обозначается защищенный атрибут в Python?

- A) Без подчеркивания: `attribute`
- B) С одним подчеркиванием: `_attribute`
- C) С двумя подчеркиваниями: `__attribute`
- D) С тремя подчеркиваниями: `___attribute`

5. Что выведет следующий код?

```
class Dog:
    def __init__(self, name):
        self.name = name

    def bark(self):
        return "Гав!"

my_dog = Dog("Бобик")
print(my_dog.name)
```

6. Как правильно обратиться к методу объекта?

- A) `object::method()`
- B) `object.method()`
- C) `method(object)`
- D) `object->method()`

7. Какой декоратор используется для создания свойства только для чтения?

- A) @property
- B) @readonly
- C) @getter
- D) @staticmethod

8. Что такое полиморфизм? _____

9. Какой принцип ООП нарушен в следующем коде?

```
class BankAccount:
    def __init__(self, balance):
        self.balance = balance # Публичный доступ к балансу

account = BankAccount(1000)
account.balance = -500 # Можем установить отрицательный баланс
```

- A) Наследование
- B) Инкапсуляция
- C) Полиморфизм
- D) Абстракция

10. Какой модификатор доступа позволяет обращаться к атрибуту только внутри класса?

- A) Public
- B) Protected
- C) Private
- D) Static

11. Что произойдет при попытке доступа к приватному атрибуту извне класса?

- A) Успешный доступ
- B) Ошибка компиляции
- C) Ошибка выполнения AttributeError
- D) Вернется значение None

12. Как правильно организовать доступ к приватному атрибуту?

- A) Сделать его публичным
- B) Использовать геттеры/сеттеры
- C) Обращаться напрямую
- D) Использовать глобальные переменные

13. Какой декоратор используется для создания свойства-геттера?

- A) @getter
- B) @property
- C) @attribute
- D) @getattr

14. Что такое капсула в контексте инкапсуляции?

- A) Метод класса
- B) Объект класса
- C) Интерфейс взаимодействия
- D) Контейнер для данных и методов

15. Какой модификатор следует использовать для атрибута, который нужен только внутри класса и его подклассов?

- A) Public
- B) Protected
- C) Private
- D) Final

Раздел 3

16. Что такое абстракция в ООП? _____

17. Какой будет результат выполнения кода?

```
class Counter:
    count = 0

    def __init__(self):
        Counter.count += 1
```

```
c1 = Counter()
c2 = Counter()
c3 = Counter()
print(Counter.count)
A) 0
B) 1
C) 3
D) Ошибка
```

18. Что такое классическая переменная?

- A) Переменная, доступная только внутри одного метода
- B) Переменная, общая для всех экземпляров класса
- C) Переменная, которая меняется для каждого объекта
- D) Переменная, объявленная вне класса

19. Как создать статический метод?

- A) Использовать декоратор @staticmethod
- B) Использовать декоратор @classmethod
- C) Объявить метод без параметра self
- D) Использовать ключевое слово static

20. Что делает метод __str__?

- A) Вызывается при создании объекта
- B) Возвращает строковое представление объекта
- C) Сравнивает два объекта
- D) Удаляет объект

21. Как правильно инкапсулировать атрибут для чтения извне, но изменения только через метод?

- A) Сделать атрибут публичным

- B) Использовать @property без сеттера
- C) Использовать @property с сеттером
- D) Сделать атрибут приватным

22. Какой декоратор используется для создания статического метода?

- A) @staticmethod
- B) @classmethod
- C) @abstractmethod
- D) @static

23. Какой параметр получает статический метод? _____

24. Что выведет этот код?

```
class MathUtils:  
    @staticmethod  
    def multiply(x, y):  
        return x * y
```

```
result = MathUtils.multiply(4, 5)
```

```
print(result)
```

- A) 20
- B) Ошибка
- C) None
- D) 9

25. Можно ли вызвать статический метод через экземпляр класса?

- A) Нет, только через класс
- B) Да, можно
- C) Только если метод публичный
- D) Только если метод защищенный

26. В чем основное отличие статического метода от метода класса?

- A) Статический метод получает cls
- B) Метод класса не получает параметров
- C) Статический метод не получает неявных параметров
- D) Нет отличий

27. Из какого модуля нужно импортировать ABC и abstractmethod?

- A) abstract
- B) abc
- C) base
- D) abstracts

28. Может ли абстрактный класс содержать неабстрактные методы? _____

30. Что такое "интерфейс" в контексте абстрактных классов?

- A) Графический интерфейс пользователя

- B) Набор абстрактных методов, которые должны быть реализованы
- C) Родительский класс
- D) Встроенный метод Python

31. Когда использовать абстрактные классы, а когда интерфейсы?

- A) В Python нет интерфейсов, только абстрактные классы
- B) Абстрактные классы для частичной реализации, интерфейсы для контракта
- C) Интерфейсы для наследования, абстрактные классы для композиции
- D) Разницы нет

Раздел 4

32. Что такое модуль в Python?

- A) Файл с расширением .py
- B) Встроенная функция Python
- C) Тип данных

33. Как установить пакет с помощью pip?

- A) pip download numpy
- B) pip install numpy
- C) pip get numpy

35. Какой командой можно посмотреть список установленных пакетов?

- A) pip show
- B) pip list
- C) pip packages

36. Для чего используется виртуальное окружение? _____

37. Как импортировать модуль под псевдонимом?

- A) import module as m
- B) alias module m
- C) from module import m

Раздел 5

38. Какой режим открытия файла позволяет только читать файл и устанавливает указатель в начало?

- а) 'w'
- б) 'a'
- в) 'r'
- г) 'x'

39. Что произойдет, если открыть несуществующий файл в режиме 'r'?

- а) Будет создан новый файл.
- б) Программа завершится с ошибкой FileNotFoundError.

- в) Файл откроется в пустом состоянии.
- г) Ничего, файл будет проигнорирован.

40. Какой метод используется для чтения всего содержимого файла в строку?

- а) `readline()`
- б) `read(size)`
- в) `read()`
- г) `readall()`

41. Для чего используется метод `seek(0)`?

- а) Для закрытия файла.
- б) Для перемещения указателя файла в начало.
- в) Для удаления содержимого файла.
- г) Для поиска строки в файле.

42. Какой из этих блоков кода гарантированно закроет файл, даже если произойдет ошибка?

а)

```
file = open('test.txt', 'r')
data = file.read()
file.close()
```

б)

```
with open('test.txt', 'r') as file:
    data = file.read()
```

- а) Только первый
- б) Только второй
- в) Оба
- г) Ни один

43. Какой модуль следует использовать для работы с путями файловой системы кросс-платформенным способом?

- а) `os.path`
- б) `sys.path`
- в) `pathlib` (оба верные, но `pathlib` — более современный)
- г) `glob`

44. Что делает функция `os.remove('file.txt')`?

- а) Переименовывает файл.
- б) Перемещает файл в корзину.
- в) Удаляет файл.
- г) Создает копию файла.

45. Какой метод из модуля `shutil` используется для рекурсивного копирования всей директории?

- a) `shutil.copy()`
- б) `shutil.copy2()`
- в) `shutil.copypath()`
- г) `shutil.move()`

46. Что возвращает метод `readline()` при достижении конца файла?

- a) `None`
- б) Пустую строку "
- в) `False`
- г) Вызывает исключение `EOFError`

47. Какой режим открытия файла ('w', 'a', 'r+', 'x') НЕ сотрет содержимое существующего файла?

- a) 'w'
- б) 'a'
- в) 'r+'
- г) 'x'

Раздел 6

48. Как создать массив NumPy из списка?

- A) `np.array(list)`
- B) `np.create(list)`
- C) `np.makearray(list)`

49. Как получить размерность массива?

- A) `array.size`
- B) `array.shape`
- C) `array.dim`

50. Как транспонировать матрицу?

- A) `matrix.transpose()`
- B) `matrix.T`
- C) Оба варианта верны

51. Какой модуль SciPy используется для решения СЛАУ?

- A) `scipy.linalg`
- B) `scipy.linear`
- C) `scipy.equations`

52. Какая функция используется для вычисления определенного интеграла?

- A) `scipy.integrate.quad`
- B) `scipy.calc.integral`
- C) `scipy.math.integrate`

53. Какая библиотека является наиболее популярной для создания статических визуализаций в Python?

- A) Plotly
- B) Seaborn
- C) Matplotlib
- D) Bokeh

54. Какой командой в Matplotlib можно создать линейный график?

- A) plt.bar()
- B) plt.scatter()
- C) plt.hist()
- D) plt.plot()

55. Какой параметр в Seaborn используется для разделения данных по категориям с помощью цвета?

- A) size
- B) style
- C) hue
- D) palette

56. Какая библиотека лучше всего подходит для создания интерактивных визуализаций, которые можно встраивать в веб-страницы?

- A) Matplotlib
- B) Seaborn
- C) Plotly
- D) Pandas

57. Что делает метод plt.subplots() в Matplotlib?

- A) Создает один график
- B) Сохраняет график в файл
- C) Создает сетку подграфиков
- D) Изменяет размер графика

58. Какой тип визуализации лучше всего подходит для отображения распределения непрерывной переменной?

- A) Круговая диаграмма
- B) Столбчатая диаграмма
- C) Гистограмма
- D) Линейный график

59. Какой командой можно сохранить график Matplotlib в файл с высоким разрешением?

- A) plt.show()
- B) plt.savefig('plot.png', dpi=300)
- C) plt.export('plot.png')
- D) plt.write('plot.png')

60. Что такое "Figure" и "Axes" в терминологии Matplotlib?

- A) Figure - ось X, Axes - ось Y
- B) Figure - весь график, Axes - область для построения данных
- C) Figure - заголовок, Axes - легенда
- D) Figure - сетка, Axes - цвета

61. Какой метод Seaborn используется для построения попарных отношений в наборе данных?

- A) sns.pairplot()
- B) sns.jointplot()
- C) sns.relplot()
- D) sns.catplot()

62. Какой тип визуализации наиболее эффективен для отображения корреляционной матрицы?

- A) Линейный график
- B) Столбчатая диаграмма
- C) Тепловая карта (heatmap)
- D) Круговая диаграмма

63. Как добавить заголовок к графику в Matplotlib?

- A) plt.title('Мой график')
- B) plt.header('Мой график')
- C) plt.name('Мой график')
- D) plt.caption('Мой график')

64. Какая библиотека построена поверх Matplotlib и предоставляет более привлекательные стили по умолчанию?

- A) Plotly
- B) Vokeh
- C) Seaborn
- D) ggplot

65. Что делает параметр alpha в функциях построения графиков Matplotlib?

- A) Задаёт толщину линии
- B) Задаёт прозрачность
- C) Задаёт цвет
- D) Задаёт стиль линии

Темы рефератов

по дисциплине «Объектно – ориентированное программирование»

1. Принципы объектно-ориентированного программирования в Python: сравнительный

- анализ. (Общий обзор всех четырёх принципов)
2. Инкапсуляция в Python: особенности реализации через механизм name mangling и свойства (@property). (Почему в Python нет "полной" инкапсуляции и как это обойти)
 3. Наследование в Python: простое, множественное и иерархия классов. Роль метода super() и MRO (Method Resolution Order). (Можно сделать акцент на проблемах ромбовидного наследования)
 4. Полиморфизм в Python: утиная типизация (duck typing) и перегрузка операторов (магические методы). (Как Python реализует полиморфизм без строгих интерфейсов)
 5. Абстракция в Python: абстрактные базовые классы (ABC) и модуль abc. Сравнение с интерфейсами в других языках.
 6. Магические методы (dunder methods) в Python: создание интуитивно понятных и интегрируемых объектов.
 7. Метод разрешения порядка (MRO) в Python: алгоритм C3 и его практическое значение при множественном наследовании.
 8. Декрипторы в Python: глубокий взгляд на работу @property, @classmethod и @staticmethod.
 9. Метапрограммирование в Python: метаклассы (тип type) и их использование для регистрации классов, валидации и ORM.
 10. Модель данных Python: как реализация специальных методов (__len__, __getitem__, __iter__) позволяет создавать Pythonic-коллекции.
 11. Контекстные менеджеры в Python: реализация с помощью магических методов __enter__ и __exit__.
 12. Миксины (Mixins) в Python: шаблон для повторного использования кода и расширения функциональности классов.
 13. Паттерны проектирования (GoF) в Python: идиоматическая реализация на примере Singleton, Factory, Observer, Strategy. (Можно взять 1-2 паттера и разобрать подробно)
 14. Принципы SOLID в контексте объектно-ориентированного программирования на Python.
 15. Композиция против наследования в Python: выбор подхода для повышения гибкости и уменьшения связанности кода.
 16. Dependency Injection (Внедрение зависимостей) в Python-приложениях: простые подходы без использования фреймворков.
 17. Объектно-ориентированный и процедурный подход: сравнительный анализ на примере решения конкретной задачи (например, разработки простой игры).
 18. ООП в современных Python-фреймворках: анализ использования классов и наследования в Django (модели, представления) или Flask (blueprints, расширения).
 19. Создание собственного ORM (Object-Relational Mapping) на Python: изучение принципов ООР через практическую задачу.
 20. Протоколы в Python (структурная типизация): современная альтернатива ABC для реализации полиморфизма.
 21. Дата-классы (dataclasses) в Python: упрощение создания классов-контейнеров данных. Сравнение с обычными классами и namedtuple.
 22. Принцип единственной ответственности (Single Responsibility Principle) на практике: рефакторинг "божественного объекта" (God Object).

5. Методические материалы, определяющие процедуры оценивания компетенции

5.1 Критерии оценивания качества выполнения лабораторного практикума

Оценка «**зачтено**» выставляется обучающемуся, если лабораторная работа выполнена правильно и обучающийся ответил на все вопросы, поставленные преподавателем на защите.

Оценка «**не зачтено**» выставляется обучающемуся, если лабораторная работа выполнена не правильно или обучающийся не проявил глубоких теоретических знаний при защите работы

5.2 Критерии оценивания качества устного ответа

Оценка «**отлично**» выставляется за глубокое знание предусмотренного программой материала, за умение четко, лаконично и логически последовательно отвечать на поставленные вопросы.

Оценка «**хорошо**» – за твердое знание основного (программного) материала, за грамотные, без существенных неточностей ответы на поставленные вопросы.

Оценка «**удовлетворительно**» – за общее знание только основного материала, за ответы, содержащие неточности или слабо аргументированные, с нарушением последовательности изложения материала.

Оценка «**неудовлетворительно**» – за незнание значительной части программного материала, за существенные ошибки в ответах на вопросы, за неумение ориентироваться в материале, за незнание основных понятий дисциплины.

5.3 Критерии оценивания тестирования

При тестировании все верные ответы берутся за 100%.

90%-100% отлично

75%-90% хорошо

60%-75% удовлетворительно

менее 60% неудовлетворительно

5.4 Критерии оценивания результатов освоения дисциплины

Оценка «**отлично**» выставляется за глубокое знание предусмотренного программой материала, содержащегося в основных и дополнительных рекомендованных литературных источниках, за умение четко, лаконично и логически последовательно отвечать на поставленные вопросы, за умение анализировать изучаемые явления в их взаимосвязи и диалектическом развитии, применять теоретические положения при решении практических задач.

Оценка «**хорошо**» – за твердое знание основного (программного) материала, включая расчеты (при необходимости), за грамотные, без существенных неточностей ответы на поставленные вопросы, за умение применять теоретические положения для решения практических задач.

Оценка «**удовлетворительно**» – за общее знание только основного материала, за ответы, содержащие неточности или слабо аргументированные, с нарушением последовательности изложения материала, за слабое применение теоретических положений при решении практических задач.

Оценка «**неудовлетворительно**» – за незнание значительной части программного материала, за существенные ошибки в ответах на вопросы, за неумение ориентироваться в расчетах, за незнание основных понятий дисциплины