

Федеральное государственное образовательное бюджетное учреждение
высшего образования
**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финуниверситет)**

Тульский филиал Финуниверситета

Кафедра «Математика и информатика»

УТВЕРЖДАЮ
Директор филиала

Г.В. Кузнецов
24 декабря 2024 г.

Р.А. Жуков

АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ В ЯЗЫКЕ PYTHON

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки
09.03.02 Информационные системы и технологии,
образовательная программа «Информационные технологии в экономике»,
профиль «Информационные технологии в управлении цифровой
экономикой»

Рекомендовано Ученым советом Тульского филиала Финуниверситета
(протокол от 24 декабря 2024 г. № 22)

Одобрено кафедрой «Математика и информатика»
(протокол от 02 декабря 2024 г. № 5)

Тула 2024

СОДЕРЖАНИЕ

№ п/п	Наименования	Стр.
1.	Название дисциплины	3
2.	Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	3
3.	Место дисциплины в структуре образовательной программы.	4
4.	Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	4
5.	Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объёмов (в академических часах) и видов учебных занятий	4
5.1.	Содержание дисциплины	4
5.2.	Учебно-тематический план	7
5.3.	Содержание семинаров, практических занятий	8
6.	Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	10
6.1.	Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы.	10
6.2.	Перечень вопросов, заданий, тем для подготовки к текущему контролю (согласно таблице 2)	11
7.	Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	13
8.	Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	18
9.	Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	19
10.	Методические указания для обучающихся по освоению дисциплины	19
11.	Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационно справочных систем (при необходимости)	20
12.	Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	21

1. Наименование дисциплины

Алгоритмы и структуры данных в языке Python.

2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	1. Разрабатывает алгоритмы решения простых информационных задач и выражает их на языке программирования.	Знать: типы и структуры данных, используемые в языке Python. Уметь: разрабатывать алгоритмы решения простых информационных задач и выражает их на языке Python.
		2. Анализирует алгоритмы в части производительности, оптимальности, вырабатывает рекомендации для оптимизации алгоритмов программ.	Знать: подходы к оценке алгоритмов языка Python в части производительности и оптимальности. Уметь: анализировать алгоритмы языка Python и вырабатывать рекомендации для оптимизации алгоритмов программ.
		3. Проводит ручное и автоматизированное тестирование программных продуктов по методам черного и белого ящика, составляет набор тестовых случаев.	Знать: технологии тестирования программных продуктов на языке Python по методам черного и белого ящика. Уметь: выбирать структуры данных и алгоритмы в языке Python, позволяющие решить поставленную задачу наиболее оптимальным способом.
ОПК-7	Способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем	1. Демонстрирует знания: основных платформ, технологий и инструментальных программно-аппаратных средств для реализации информационных систем.	Знать: основные платформы, технологии и инструментальные программно-аппаратные средства на примере языка Python. Уметь: использовать основные платформы, технологии и инструментальные программно-аппаратные средства на примере языка Python.
		2. Осуществляет выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем.	Знать: основные платформы, технологии и инструментальные программно-аппаратные средства на примере языка Python. Уметь: осуществлять выбор платформ и инструментальных программно-аппаратных средств на примере языка Python.
		3. Применяет технологии и инструментальные	Знать: технологии и инструментальные программно-

		программно-аппаратные средства для реализации информационных систем.	аппаратные средства для реализации информационных систем. Уметь: использовать технологии и инструментальные программно-аппаратные средства для реализации информационных систем.
--	--	--	--

3. Место дисциплины в структуре образовательной программы

Дисциплина «Алгоритмы и структуры данных в языке Python» относится к общепрофессиональному циклу обязательной части учебного плана направления подготовки 09.03.02 Информационные системы и технологии.

4. Объем дисциплины в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся

Таблица 1

Вид учебной работы по дисциплине	Всего (в з.е. и часах)	Семестр 1 (в часах)	Семестр 2 (в часах)
Общая трудоемкость дисциплины	8 з.е./ 288	144	144
Контактная работа - Аудиторные занятия	100	50	50
Лекции	32	16	16
Семинары, практические занятия	68	34	34
Самостоятельная работа	188	94	94
Вид текущего контроля	2 контрольные работы	контрольная работа	контрольная работа
Вид промежуточной аттестации	Зачет, экзамен	Зачет	Экзамен

5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий

5.1. Содержание дисциплины

Тема 1. Введение в программирование на Python

Общая информация о языке Python. История языка программирования, его связь с другими языками программирования, распространенность Python и основные сферы его применения. Знакомство с первыми примерами кода на Python. Философия Python.

Базовая информация о языке Python. Основные типы данных. Основные числовые типы данных и операции над ними. Математические операции над числовыми типами данных. Преобразование типов данных. Переменные и специфика их объявления. Статическая и динамическая типизация. Работа с переменными. Управление памятью и сборка мусора в Python. Именованные переменные.

Работа со строками: создание строк, специальные символы. Индексирование строк, получение срезов строк. Основные функции для работы со строками. Вывод на экран (работа с функцией print) и форматирование строк.

Различные подходы к форматированию строк, форматирование с помощью f-строк. Расширенное форматирование в Python.

Тема 2. Управляющие конструкции, списки и кортежи

Управляющие конструкции в Python. Булев тип: объявление и операции. Операции сравнения в Python. Условные операторы в Python. Реализация задачи case в Python.

Циклы в Python: while, for. Специфика циклов в Python. Функции range и enumerate и их использование в циклах.

Списки и кортежи в Python. Специфика списков и их отличие от массивов. Создание списка, оперирование вложенными списками, копирование списков, операции над списками: индексация и срезы; изменение списка; поиск, сортировка и обход; изменение списка. Кортежи в Python: синтаксис, специфика использования.

Тема 3. Словари, множества и выражения-генераторы

Словари Python. Словари: семантика, синтаксис создания, операции над словарями, перебор элементов словаря.

Множества в Python. Множества: семантика, синтаксис создания, операции над словарями, перебор элементов словаря. Специфика операций с множествами в Python.

Выражения-генераторы в Python. Выражения-генераторы для списков: семантика и синтаксис. Пример: задача приведения списка к «плоскому» виду. Выражения-генераторы для множеств и словарей. Кейсы использования и производительность решений с использованием выражений-генераторов.

Тема 4. Функции

Функции в Python: общая семантика. Создание функции и ее вызов. Расположение определений функций. Анонимные функции в Python. Необязательные параметры функций и сопоставление по ключам. Возвращение нескольких значений из функции. Распаковка и упаковка параметров функции. Аннотации и документирование функций. Глобальные и локальные переменные.

Тема 5. Работа с файлами и обработка исключительных ситуаций

Обработка исключений в Python: кейсы для использования. Инструкция try ... except ... else ... finally. Классы встроенных исключений. Создание пользовательских исключений. Инструкция assert.

Работа с файлами в Python. Концепция файла в современных ОС и языках программирования. Операции с файлами: открытие/закрытие файла, чтение и записи и другие методы для работы с файлами. Инструкция with ... as и ее использование для файлов. Сохранение объектов в файл с помощью модуля pickle и shelve. Модуль CSV.

Тема 6. Модули и пакеты

Модули и пакеты в Python: подход к структурированию программного кода с помощью модулей и пакетов. Синтаксис импортирования в Python. Создание и работа с пакетами в Python. Повторная загрузка модулей.

Написание и запуск скриптов на Python. Установка модулей из глобального репозитория

Тема 7. Введение в объектно-ориентированное программирование

Предпосылки и история появления ООП. Объекты и классы в ООП. Принципы и основные механизмы ООП. Логика работы абстракции, инкапсуляции, наследования и полиморфизма.

Python как объектно-ориентированный язык программирования. Базовые возможности ООП в Python: создание классов и объектов; наследование и полиморфизм; функция `super()`; проверка принадлежности к классу. Базовые типы в Python.

Тема 8. Объектно-ориентированное программирование в Python

Методы классов и статические переменные и методы в Python. Управление доступом к атрибутам класса в Python. Динамические операции с атрибутами и интроспекция в Python. Использование специальных методов для расширенного функционала пользовательских классов. Кейс построения иерархии классов.

Тема 9. Введение в функциональное программирование

Парадигмы и идиомы программирования, общая концепция функционального программирования. Функциональные языки программирования.

Функции «граждане первого класса», функции высшего порядка, замыкания, функции без побочных эффектов, рекурсия, хвостовая рекурсия. Неизменяемые структуры данных. Идиомы, распространенные в функциональных языках программирования: итераторы, последовательности, ленивые вычисления, сопоставление с образцом, монады.

Элементы функционального программирования в Python: функции – граждане первого класса; глобальные и локальные переменные в Python; вложенные функции и замыкания в Python.

Декораторы в Python: использование и создание собственных декораторов.

Тема 10. Функциональное программирование в Python

Подход: `map`, `filter`, `reduce`. Реализация функций `map`, `filter`, `reduce` в Python.

Итераторы в Python, итерируемый тип данных. Модуль `itertools`.

Функции-генераторы и выражения-генераторы в Python.

Тема 11. Структуры данных: массивы, стеки, очереди, списки.

Введение в анализ сложности алгоритмов.

Массивы и их отличие от списков в Python. Динамические массивы, сложность операций работы с динамическими массивами.

Стек, операции со стеком. Реализации стека. Очередь, операции с очередью. Реализация очереди. Связные списки, варианты связанных списков.

Тема 12. Алгоритмы поиска и сортировки

Поиск в списках/массивах, бинарный поиск. Сортировка и ее использование в прикладных задачах.

Простые методы сортировки: обменные сортировки (с различными вариациями); сортировка выбором (извлечением); сортировка включением (вставками).

Эффективные методы сортировки: быстрая сортировка; сортировка Шелла; сортировка слиянием. Сравнение различных сортировок.

5.2 Учебно-тематический план

Таблица 2

№ п/п	Наименование тем (разделов) дисциплины	Трудоемкость в часах					Формы текущего контроля успеваемости
		Всего	Контактная работа – Аудиторная работа			Самост. работа	
			Общая	Лекции и	Семинары, практические занятия		
1	Введение в программирование на Python	24	6	2	4	18	Выполнение индивидуальн ых заданий
2	Управляющие конструкции, списки и кортежи	24	6	2	4	18	Выполнение индивидуальн ых заданий
3	Словари, множества и выражения- генераторы	24	10	2	8	14	Выполнение индивидуальн ых заданий
4	Функции	24	8	2	6	16	Выполнение индивидуальн ых заданий
5	Работа с файлами и обработка исключительных ситуаций	24	10	4	6	14	Выполнение индивидуальн ых заданий
6	Модули и пакеты	24	10	4	6	14	Выполнение индивидуальн ых заданий
	Всего в 1 семестре	144	50	16	34	94	Контрольная работа
7	Введение в объектно- ориентированное программирование	24	6	2	4	18	Выполнение индивидуальн ых заданий и заданий в малых группах
8	Объектно- ориентированное программирование в Python	24	6	2	4	18	Выполнение индивидуальн ых заданий
9	Введение в функциональное программирование	24	10	2	8	14	Выполнение индивидуальн ых заданий
10	Функциональное программирование в Python	24	8	2	6	16	Выполнение индивидуальн ых заданий
11	Структуры данных: массивы, стеки, очереди, списки	24	10	4	6	14	Выполнение индивидуальн ых заданий
12	Алгоритмы поиска и	24	10	4	6	14	Выполнение

	сортировки						индивидуальных заданий
	Всего во 2 семестре	144	50	16	34	94	Контрольная работа
	В целом по дисциплине	288	100	32	68	188	Согласно учебному плану: две контрольные работы
	Итого в %	100	35	32	68	65	-

5.3. Содержание семинаров, практических занятий

Таблица 3

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях, рекомендуемые источники из разделов 8, 9 (указывается раздел и порядковый номер источника)	Формы проведения занятий
Тема 1. Введение в программирование на Python	Установка Python, установка дистрибутива Anaconda. Работа в интерактивном режиме интерпретатора. Интерактивная оболочка IPython notebook: принципы работы и применения. Рекомендуемые источники: 8.1-4; 9.	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Тема 2. Управляющие конструкции, списки и кортежи	Базовые числовые типы, строки, списки, словари, переменные, базовые операторы. Рекомендуемые источники: 8.1-4; 9.	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Тема 3. Словари, множества и выражения-генераторы	Множества в Python. Множества: семантика, синтаксис создания, операции над словарями, перебор элементов словаря. Специфика операций с множествами в Python. Выражения-генераторы в Python. Выражения-генераторы для списков: семантика и синтаксис. Рекомендуемые источники: 8.1-4; 9.	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Тема 4. Функции	Создание функций, область видимости переменной, передача аргументов в функцию. Лямбда-функции. Рекомендуемые источники: 8.1-4; 9.	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Тема 5. Работа с файлами и обработка	Исключения. Инструкция try...except...else...finally. Классы	Выполнение практических заданий

исключительных ситуаций	встроенных исключений. Создание пользовательских исключений. Инструкция <code>assert</code> . Работа с файлами в Python, операции с файлами: открытие/закрытие файла, чтение и записи и другие методы для работы с файлами. Инструкция <code>with ... as</code> и ее использование для файлов. Рекомендуемые источники: 8.1-4; 9.	под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Тема 6. Модули и пакеты	Устройство модулей и пакетов, инструкции <code>import</code> и <code>from</code> . Создание собственных модулей и пакетов. Рекомендуемые источники: 8.1-4; 9.	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Тема 7. Введение в объектно-ориентированное программирование	Базовые возможности ООП в Python: создание классов и объектов; наследование и полиморфизм; функция <code>super()</code> . Рекомендуемые источники: 8.1-4; 9.	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Тема 8. Объектно-ориентированное программирование в Python	Методы классов и статические переменные и методы в Python. Управление доступом к атрибутам класса в Python. Динамические операции с атрибутами и интроспекция в Python. Использование специальных методов для расширенного функционала пользовательских классов. Кейс построения иерархии классов. Рекомендуемые источники: 8.1-4; 9.	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Тема 9. Введение в функциональное программирование	Элементы функционального программирования в Python: функции – граждане первого класса; глобальные и локальные переменные в Python; вложенные функции и замыкания в Python. Декораторы в Python: использование и создание собственных декораторов. Рекомендуемые источники: 8.1-4; 9.	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Тема 10. Функциональное программирование в Python	Реализация функций <code>map</code> , <code>filter</code> , <code>reduce</code> в Python. Итераторы в Python, итерируемый тип данных. Модуль <code>itertools</code> . Рекомендуемые источники: 8.1-4; 9.	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Тема 11. Структуры данных: массивы, стеки, очереди,	Массивы и их отличие от списков в Python. Динамические массивы, сложность операций работы с	Выполнение практических заданий под руководством

списки	динамическими массивами. Стек, операции со стеком. Реализации стека. Очередь, операции с очередью. Реализация очереди. Связные списки, варианты связанных списков. Рекомендуемые источники: 8.1-4; 9.	преподавателя и самостоятельно. Решение задач. Разбор решений
Тема 12. Алгоритмы поиска и сортировки	Реализация на Python простых методов сортировки: обменные сортировки (с различными вариациями); сортировка выбором (извлечением); сортировка включением (вставками). Реализация на Python эффективных методов сортировки: быстрая сортировка; сортировка Шелла; сортировка слиянием. Рекомендуемые источники: 8.1-4; 9.	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений

6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Таблица 4

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Тема 1. Введение в программирование на Python	Среда программирования. Использование документации.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
Тема 2. Управляющие конструкции, списки и кортежи	Оперирование вложенными списками, копирование списков, некоторые операции над списками.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
Тема 3. Словари, множества и выражения-генераторы	Выражения-генераторы в Python. Выражения-генераторы для словарей и множеств: семантика, синтаксис и практическое использование.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
Тема 4. Функции	Возвращение нескольких значений из функции. Распаковка и упаковка параметров функции.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
Тема 5. Работа с файлами и обработка исключительных ситуаций	Сохранение объектов в файл с помощью модуля pickle и shelve. Модуль CSV.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
Тема 6. Модули и пакеты	Написание и запуск скриптов на Python. Установка модулей из глобального репозитория.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
Тема 7. Введение в объектно-ориентированное	Базовые типы в Python: взгляд с точки зрения ООП. Методы базовых типов.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
программирование		Notebook.
Тема 8. Объектно-ориентированное программирование в Python	Проверка принадлежности к классу и интроспекция в Python.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
Тема 9. Введение в функциональное программирование	Реализация декораторов с параметрами в Python.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
Тема 10. Функциональное программирование в Python	Функции-генераторы и выражения-генераторы в Python.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
Тема 11. Структуры данных: массивы, стеки, очереди, списки	Реализация различных вариантов связанных списков на Python.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
Тема 12. Алгоритмы поиска и сортировки	Сравнение различных сортировок (простых и эффективных) с использованием их реализаций на Python.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.

6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю (согласно таблице 2)

Текущий контроль осуществляется в ходе учебного процесса и контроля самостоятельной работы обучающихся, по результатам выполнения контрольных работ. Основными формами текущего контроля знаний являются:

- обсуждение вопросов и задач, вынесенных в планах семинарских занятий в качестве самостоятельных заданий;
- решение задач в аудитории по теме занятия, их обсуждение;
- подготовка, обсуждение вопросов и задач внеаудиторных самостоятельных и аудиторных контрольных работ.

Примеры заданий контрольных работ

Контрольная работа 1

1. В строке содержащей последовательность слов, разделенных запятыми удалить все нечетные слова. Ответ представить в виде строки. Пример: строка 'SIX,SEVEN,EIGHT,NINE,TEN' будет преобразована в: 'SIX,EIGHT,TEN'.

2. Из списка списков элементами которого являются текстовые символы собрать строку, в которой вложенные списки объединены в слова, а слова через запятую объединены в строку. Пример список вида [['E', 'e', 'n', 'y'], ['m', 'e', 'e', 'n', 'y'], ['m', 'i', 'n', 'e', 'y'], ['m', 'o', 'e']] будет преобразован в строку

‘Eeny, meeny, miney, moe’

Контрольная работа 2

1. Создать иерархию классов для фруктов, продающихся в магазине. Иерархия должна содержать не менее 3 классов. Объекты должны содержать не менее 2х атрибутов и 2х методов. Реализовать механизм автоматического подсчета количества всех созданных фруктов и автоматического присвоения каждому фрукту уникального идентификатора. Необходимо заполнить список представителями всех классов (всего не менее 10 объектов) и продемонстрировать работу созданного механизма.
2. Реализовать декоратор с именем `not_none`, который генерирует исключительную ситуацию если декорируемая функция вернула значения `None`. (20 баллов)
3. Используя генератор словарей (и не используя код вне него) инвертировать словарь, т.е. сделать ключи словаря, его значениями и наоборот. Значения, которые в исходном словаре повторяются не добавлять в итоговый словарь. Пример: `{'a':1, 'b':3, 'c':4, 'd':3}` -> `{1:'a', 4:'c'}`.
4. Реализовать декоратор с именем `print_type`, выводящий на печать тип значения, возвращаемого декорируемой функцией.

Примерные вопросы к контрольным работам

Контрольная работа 1

1. Операции над основными числовыми типами данных.
2. Операции над булевскими переменными.
3. Динамическая типизация в Python.
4. Преобразование типов в Python.
5. Создание строк в Python.
6. Организация и пример цикла `while` в Python.
7. Организация и пример цикла `for` в Python.
8. Операции над словарями в Python.
9. Выражения-генераторы для списков в Python.
10. Необязательные параметры функций в Python.
11. Заpackовка и распаковка параметров в Python.
12. Аннотации и документирование функций.
13. Создание объектов в Python.
14. Управление доступом к атрибутам класса в Python.
15. Выполнение интроспекции в Python.
16. Замыкания в Python.
17. Использование декораторов в Python.
18. Создание собственных декораторов в Python.
19. Компилятор и интерпретатор. Достоинства и недостатки.
20. Назовите и дайте краткую характеристику основных классов языков программирования.
21. Встроенные числовые типы языка Python.

Контрольная работа 2

1. Списки. Создание, основные операции.
2. Основные методы списка.
3. Кортежи. Создание, основные методы и операции.
4. Словари. Создание, основные операции.
5. Методы для работы со словарями.
6. Множества. Создание, основные методы и операции.
7. Переменные. Правила именования переменных.
8. Динамическая типизация.
9. Операторы сравнения и логические операторы.
10. Инструкция if...else.
11. Инструкция цикла while.
12. Инструкция цикла for.
13. Создание и вызов функции.
14. Передача аргументов функцию.
15. Функции-генераторы.
16. Лямбда-функции.
17. Модули. Инструкции import и from.
18. Базовые принципы объектно-ориентированного программирования.
19. Класс, метод класса, атрибут класса. Определение класса и создание экземпляра класса.
20. Конструктор и деструктор.
21. Наследование.
22. Абстрактные методы класса.
23. Статические методы класса.
24. Свойства класса.
25. Исключения. Обработка исключений.
26. Пользовательские исключения.

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях кафедры.

7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине содержится в разделе «2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине».

Таблица 5

Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые задания
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	Индикатор 1. Разрабатывает алгоритмы решения простых информационных задач и выражает их на языке программирования .	Знать: типы и структуры данных, используемые в языке Python. Уметь: разрабатывать алгоритмы решения простых информационных задач и выражает их на языке Python	Задание 1. Напишите скрипт на Python обрабатывающий текстовый файл так, что в нем все заглавные буквы преобразуются в строчные. Задание 2. Установите пакет Anaconda запустите программу Jupyter Notebook составьте тестовый код на Python.
	Индикатор 2. Анализирует алгоритмы в части производительности, оптимальности, вырабатывает рекомендации для оптимизации алгоритмов программ.	Знать: подходы к оценке алгоритмов языка Python в части производительности и оптимальности. Уметь: анализировать алгоритмы языка Python и вырабатывать рекомендации для оптимизации алгоритмов программ.	Задание 1. Выберите библиотеку Python для хэширования данных в целях быстрого поиска текстовых строк. Задание 2. Реализуйте программу на Python которой в качестве аргумента командной строки передается имя CSV-файла, в первом столбце которого находятся числа, которые необходимо отсортировать. Программа создает новый файл, в котором первый столбец отсортирован.
	Индикатор 3. Проводит ручное и автоматизированное тестирование программных продуктов по методам черного и белого ящика, составляет набор тестовых случаев.	Знать: технологии тестирования программных продуктов на языке Python по методам черного и белого ящика. Уметь: выбирать структуры данных и алгоритмы в языке Python, позволяющие решить	Задание 1. Студенты проходили тестирование, результаты которого были сохранены в списке вида ["Иванов 75", "Петров 63", "Петров 84"]. Студент мог проходить тестирование несколько раз. Выведите на экран список студентов и максимальный балл, полученный им при тестировании, в виде таблицы.

		поставленную задачу наиболее оптимальным способом.	
ОПК-7. Способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем	Индикатор 1. Демонстрирует знания: основных платформ, технологий и инструментальных программно-аппаратных средств для реализации информационных систем.	Знать: основные платформы, технологии и инструментальные программно-аппаратные средства на примере языка Python. Уметь: использовать основные платформы, технологии и инструментальные программно-аппаратные средства на примере языка Python.	Задание 1. Опишите структуру CSV файла фиксирующего факты продаж, необходимого для подсчета суммарных продаж по различным географическим подразделениям фирмы. Задание 2. Опишите закономерности/вариабельность в синтаксических конструкциях Python для работы со списками и словарями.
	Индикатор 2. Осуществляет выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем.	Знать: основные платформы, технологии и инструментальные программно-аппаратные средства на примере языка Python. Уметь: осуществлять выбор платформ и инструментальных программно-аппаратных средств на примере языка Python.	Задание 1. Создать иерархию классов для фруктов, продающихся в магазине. Иерархия должна содержать не менее 3 классов. Объекты должны содержать не менее 2х атрибутов и 2х методов. Реализовать механизм автоматического подсчета количества всех созданных фруктов и автоматического присвоения каждому фрукту уникального идентификатора. Необходимо заполнить список представителями всех классов (всего не менее 10 объектов) и продемонстрировать работу созданного механизма. Задание 2. Опишите подходы используемых в Python для группировки функций ориентированных на обработку однотипных объектов и решение близких задач.
	Индикатор 3.	Знать: технологии	Задание 1. Сформируйте

	Применяет технологии и инструментальные программно-аппаратные средства для реализации информационных систем.	и инструментальные программно-аппаратные средства для реализации информационных систем. Уметь: использовать технологии и инструментальные программно-аппаратные средства для реализации информационных систем.	собственные суждения о применимости Python для решения задач машинного обучения. Задание 2. Выскажите свою точку зрения о применимости Python для создания крупной корпоративной транзакционной системы на языке программирования Python.
--	---	--	--

Перечень примерных вопросов к зачету

1. История развития языков программирования.
2. Понятие языка программирования. Синтаксис и семантика языка. Способы реализации языков: компиляция, интерпретация, смешанный подход.
3. Язык программирования Python и его место среди других языков программирования. Установка Python.
4. Работа в интерактивном режиме интерпретатора. Среда программирования. Использование документации.
5. Работа с числами. Базовые числовые типы `int` и `float`. Числовые литералы. Операторы для работы с числовыми объектами. Форматы чисел. Встроенные функции и модули для работы с числами. Преобразование и смешивание в выражениях значений разных типов.
6. Строки. Литералы строк. Типы `str`, `bytes`, `bytearray`. Операции над строками: конкатенация, повторение, доступ по индексу, получение подстроки, проверка вхождения.
7. Форматирование строк. Функции и методы для работы со строками. Регулярные выражения.
8. Списки. Создание списка. Операции над списками. Перебор элементов списка. Многомерные списки. Методы списков. Кортежи.
9. Словари. Создание словаря. Операции над словарями. Перебор элементов словаря. Методы для работы со словарями. Множества.
10. Переменные. Правила именования переменных. Присваивание значения переменным. Динамическая типизация.
11. Понятие о счетчике ссылок и сборке мусора. Проверка и преобразование типов данных. Удаление переменной.
12. Структура программы. Комментарии. Блок. Правила оформления отступов.

13. Инструкция присваивания. Групповое присваивание. Комбинированные инструкции присваивания. Инструкции выражений. Функция print.

Перечень примерных вопросов к экзамену

1. Операторы сравнения. Логические операторы. Инструкция ветвления if...else. Инструкция цикла while. Инструкция цикла for.
2. Инструкции break, continue, pass, else.
3. Функции. Создание функций. Область видимости переменной. Передача аргументов в функцию. Инструкция return. Вызов функции.
4. Лямбда-функции. Функции-генераторы. Инструкция yield.
5. Модули. Инструкции import и from. Обзор стандартной библиотеки Python.
6. Базовые принципы объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм.
7. Классы в языке Python. Инструкция class. Создание экземпляра класса. Атрибуты класса и экземпляра класса. Методы класса.
8. Python. Конструктор и деструктор. Статические методы. Закрытые методы и атрибуты. Свойства. Наследование. Множественное наследование. Перегрузка операторов.
9. Событийно-ориентированное программирование. Событие. Обработчик события. Цикл обработки событий.
10. Инструменты для создания графических интерфейсов пользователя (GUI).
11. Создание базового окна. Элементы графического интерфейса (виджеты).
12. Создание и конфигурирование виджета. Менеджер размещения.
13. Использование элементов Кнопка, Надпись, Текстовое поле, Флажок, Переключатель.
14. Связывание виджетов по событиям с функциями.
15. Создание графических объектов. Поле Canvas. Редактирование объектов. Перемещение, изменение размера, удаление.
16. Инструкции break, continue, pass, else.
17. Функции. Создание функций. Область видимости переменной. Передача аргументов в функцию. Инструкция return. Вызов функции.
18. Лямбда-функции. Функции-генераторы. Инструкция yield.
19. Модули. Инструкции import и from. Обзор стандартной библиотеки Python.
20. Базовые принципы объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм.
21. Классы в языке Python. Инструкция class. Создание экземпляра класса. Атрибуты класса и экземпляра класса. Методы класса.
22. Python. Конструктор и деструктор. Статические методы. Закрытые методы и атрибуты. Свойства. Наследование. Множественное наследование. Перегрузка операторов.

23. Событийно-ориентированное программирование. Событие. Обработчик события. Цикл обработки событий.
24. Инструменты для создания графических интерфейсов пользователя (GUI).
25. Создание базового окна. Элементы графического интерфейса (виджеты).
26. Создание и конфигурирование виджета. Менеджер размещения.

Пример экзаменационного билета

Федеральное государственное образовательное бюджетное учреждение
высшего образования

«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)

Кафедра «Математика и информатика»

Дисциплина «Алгоритмы и структуры данных в языке Python»

Филиал Тульский

Семестр: 2 Направление подготовки: Информационные системы и технологии

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

1. Создание базового окна. Элементы графического интерфейса (виджеты) (10 баллов).
2. Модули. Инструкции `import` и `from`. Обзор стандартной библиотеки Python (10 баллов).
3. Напишите скрипт, реализующий последовательность действий по созданию таблицы прихода / расхода / списания Товара1 в течение месяца в файл `mov.csv`, организовать считывание из этого файла данных по строкам и столбцам в структуру `table`. (40 баллов)

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература

1. Жуков Р.А. Язык программирования Python: практикум: учебное пособие/ Р.А. Жуков. - Москва: ИНФРА-М, 2024. - 216 с. + Доп. материалы [Электронный ресурс]. Текст: электронный. URL: <https://znanium.ru/catalog/product/2139862>. – Режим доступа: по подписке.

Дополнительная литература

2. Андреева О.В. Основы алгоритмизации и программирования на языке Python: учебник/О.В. Андреева О.И. Ремизова. Москва: Издательский Дом НИТУ «МИСиС», 2022. 149 с. Текст: электронный. URL: <https://znanium.com/catalog/product/1914825> – Режим доступа: по подписке.
3. Карякин М.И. Технологии программирования и компьютерный практикум на языке Python: учебное пособие/М.И. Карякин, К.А. Ватульян, Р.М. Мнухи; Южный федеральный университет. - Ростов-на-Дону; Таганрог: Издательство Южного федерального университета, 2022. 242 с. Текст:

электронный. URL: <https://znanium.com/catalog/product/2057604>. – Режим доступа: по подписке.

4. Солем Я.Э. Программирование компьютерного зрения на языке Python: практическое руководство/Я.Э. Солем; пер. с англ. А.А. Слинкина. 2-е изд. Москва: ДМК Пресс, 2023. 313 с. Текст: электронный. URL: <https://znanium.ru/catalog/product/2102597>. – Режим доступа: по подписке.

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>
2. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
3. Электронно-библиотечная система «Университетская библиотека ОНЛАЙН» <http://biblioclub.ru/>
4. Электронно-библиотечная система Znanium <http://www.znanium.com>
5. Электронно-библиотечная система издательства «ЮРАЙТ» <https://www.urait.ru/>
6. Электронно-библиотечная система издательства «Лань» <https://e.lanbook.com/>
7. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>
8. Национальная электронная библиотека <http://нэб.рф/>

10. Методические указания для обучающихся по освоению дисциплины

Основная цель освоения содержания дисциплины получить теоретические знания в ходе лекционных занятий и семинаров, закрепить их в процессе систематической самостоятельной работы и сформировать знания, умения и владения в соответствии с компетенциями, содержащимися в рабочей программе по дисциплине.

Самостоятельная работа студента в процессе освоения дисциплины включает в себя:

- подготовку к семинарским занятиям;
- работу с электронными учебными ресурсами;
- изучение основной и дополнительной литературы по дисциплине;
- индивидуальные или групповые консультации по наиболее сложным вопросам;
- подготовка к зачету/экзамену.

возникшие вопросы и своевременно обращаться для разъяснения к лектору или преподавателю практических занятий (по графику их консультаций).

В рамках изучения предусмотрено выполнение контрольной работы.

Организация выполнения контрольной работы включает несколько этапов. Прежде всего, студент должен изучить рекомендуемую учебную литературу и внимательно ознакомиться с методическими указаниями по выполнению контрольной работы.

Выполненная и надлежащим образом оформленная работа сдается на кафедру «Математика и информатика» в сроки, установленные учебным

графиком. Если студент не допущен к защите, то работа должна быть доработана согласно замечаниям руководителя.

Критерии оценивания контрольной работы:

- 1) четкость структуры работы, представленной в содержании (рубрики дают однозначное понимание раскрываемой тематики и оформлены стилями заголовков соответствующего уровня);
- 2) наличие всех разделов, предложенных в методических указаниях, таблиц, иллюстраций (схем, рисунков и т.п.), ссылок на источники заимствованного материала;
- 3) наличие списка использованной литературы, оформленного в соответствии с правилами библиографического описания;
- 4) соответствие оформления контрольной работы общим требованиям.

Необходимо стремиться к ясности, краткости и самостоятельности изложения материала.

Цитаты и заимствованный материал должны сопровождаться ссылками на источники, описание которых необходимо привести в списке использованной литературы (например: [2, с. 15]).

Сокращение слов и использование аббревиатур, за исключением общепринятых, в контрольной работе не допускаются. Общепринятые аббревиатуры необходимо расшифровать в тексте контрольной работы при их первом употреблении.

Цифровой материал целесообразно представлять в виде таблиц. Для этого в правом верхнем углу помещают надпись «Таблица» и указывают ее порядковый номер. Таблицу снабжают тематическим заголовком, который располагают посередине страницы.

Приводимые в работе иллюстрации (схемы, диаграммы, графики, технические рисунки) должны быть выполнены четко, аккуратно и разборчиво. Рисунок должен иметь номер и подрисуночную подпись (например: Рис. 4. График функции $f(x)$).

На все таблицы и рисунки необходимо сделать ссылки в тексте работы.

На последней странице студент должен поставить подпись и дату сдачи контрольной работы на рецензирование.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем

11.1. Комплект лицензионного программного обеспечения

лицензионное отечественное:

1. Антивирусная защита Kaspersky Endpoint Security.
2. Astra Linux.
3. LibreOffice.

лицензионное импортное:

ПО для ТСО лиц с ограниченными возможностями здоровья
свободно распространяемое:

Архиватор KDE (Ark).

Интегрированная среда разработки для Python «PyCharm»
Интегрированная среда разработки и обучения для Python «IDLE»
Интерактивная веб-среда разработки для Python «Jupyter Notebook»

11.2. Современные профессиональные базы данных и информационные справочные системы

1. Информационно-правовая система «Гарант».
2. Информационно-правовая система «Консультант Плюс»
<http://www.consultant.ru>.
3. Информационно-образовательные ресурсы личных кабинетов обучающихся платформы Финансового университета - <https://org.fa.ru/>.

11.3. Сертифицированные программные и аппаратные средства защиты информации

Не используются.

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Учебная аудитория для проведения учебных занятий, предусмотренных программой бакалавриата, оснащенная оборудованием и техническими средствами обучения:

Специализированная мебель:

Парты студенческие двухместные – 32 шт.

Парты студенческие трехместные – 16 шт.

Стол преподавателя – 1 шт.

Стул для преподавателя – 3 шт.

Учебная доска меловая - 1 шт.

Кафедра переносная – 1 шт.

Технические средства обучения:

Компьютер – 1 шт.

Проектор – 1 шт.

Интерактивный экран с акустикой – 1 шт.

Учебная аудитория для проведения учебных занятий, предусмотренных программой бакалавриата, оснащенная оборудованием и техническими средствами обучения:

Специализированная мебель:

Столы для автоматизированных рабочих мест (одноместные) - 24 шт.

Стулья – 23 шт.

Стол письменный – 2 шт.

Кресло офисное – 1 шт.

Стол преподавателя – 1 шт.

Стол с двумя тумбами – 1 шт.

Тумба – 3 шт.

Учебная доска меловая - 1 шт.

Шкаф – 2 шт.

Технические средства обучения:

Моноблок – 21 шт.

Экран – 1 шт.

Колонки - 1 шт.

Помещение для самостоятельной работы.

Компьютерный класс.

Специализированная мебель:

Столы для автоматизированных рабочих мест (одноместные) - 11 шт.

Стулья – 13 шт.

Стол письменный – 2 шт.

Кресло офисное – 2 шт.

Стол преподавателя – 1 шт.

Стол под телевизор – 1 шт.

Шкаф – 1 шт.

Технические средства обучения:

Моноблок – 11 шт.

ЖК панель 55” – 1 шт.

Принтер - 1 шт.

Ноутбук - 1 шт.

Колонки - 1 шт.

Помещение для самостоятельной работы обучающихся оснащено компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно - образовательную среду Финансового университета.