

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

Тульский филиал Финуниверситета

Кафедра «Математика и информатика»

УТВЕРЖДАЮ



Директор Тульского филиала

Г.В. Кузнецов

30 мая 2023 г.

Приложение к рабочей программе дисциплины

Алгоритмы и структуры данных в языке Python

Направление подготовки 38.03.05 – Бизнес-информатика,
образовательная программа «Цифровая трансформация управления
бизнесом»,

Профиль: ИТ-менеджмент в бизнесе

Фамилия И.О. научно-педагогического работника филиала, ответственного
за актуализацию РПД Р.А. Жуков

Год утверждения рабочей программы дисциплины – 2021

*Одобрено кафедрой «Математика и информатика»
Протокол от 21 мая 2023 г. № 10*

Тула 2023

СОДЕРЖАНИЕ ПРИЛОЖЕНИЯ

№ п/п	РАЗДЕЛЫ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ С ВНЕСЁННЫМИ ИЗМЕНЕНИЯМИ И ДОПЛНЕНИЯМИ	стр.
2.	Перечень планируемых результатов освоения образовательной программы с указанием индикаторов их достижения, соотнесенных с планируемыми результатами обучения по дисциплине	3
4.	Объем дисциплины в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	4
5.2	Учебно-тематический план	5
5.3.	Содержание семинаров, практических занятий.....	6
6.	Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.....	8
6.1.	Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	8
6.2.	Перечень вопросов, заданий, тем для подготовки к текущему контролю.	9
7.	Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине.....	10
7.1	Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, знаний и умений.....	10
7.2.	Описание показателей и критериев оценивания компетенций, описание шкалы оценивания	16
8.	Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	16
10.	Методические указания для обучающихся по освоению дисциплины.....	17
11.	Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	19
11.1.	Комплект лицензионного программного обеспечения	19
11.2.	Современные профессиональные базы данных и информационные справочные системы	19
11.3.	Сертифицированные программные и аппаратные средства защиты информации	19
12.	Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	19

2. Перечень планируемых результатов освоения образовательной программы с указанием индикаторов их достижения, соотнесенных с планируемыми результатами обучения по дисциплине

В результате изучения дисциплины «Алгоритмы и структуры данных в языке Python» у студентов должны быть сформированы следующие компетенции:

Таблица 1

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
УК-4	Способность использовать прикладное программное обеспечение при решении профессиональных задач	1.Использует основные методы и средства получения, представления, хранения и обработки данных.	Знать: типы и структуры данных, используемые в языке Python. Уметь: выбирать структуры данных и алгоритмы, позволяющие решить поставленную задачу наиболее оптимальным способом.
		2.Демонстрирует владение профессиональными пакетами прикладных программ.	Знать: типы и структуры данных, используемые в языке Python. Уметь: выбирать алгоритмы, позволяющие решить поставленную задачу наиболее оптимальным способом.
		3.Выбирает необходимое прикладное программное обеспечение в зависимости от решаемой задачи.	Знать: технологии обработки, анализа и интерпретации данных различной природы. Уметь: применять алгоритмы для поиска и выявления зависимостей в данных.
		4.Использует прикладное программное обеспечение для решения конкретных прикладных задач.	Знать: технологии обработки, анализа и интерпретации данных различной природы. Уметь: применять алгоритмы для поиска и выявления зависимостей в данных.
УК-10	Способность осуществлять поиск, критически анализировать, обобщать и систематизировать информацию, использовать системный подход для решения поставленных задач	1. Четко описывает состав и структуру требуемых данных и информации, грамотно реализует процессы их сбора, обработки и интерпретации.	Знать: возможности языка программирования Python для решения математических и научных задач. Уметь: четко описывать состав и структуру данных и информации, используемых в решаемой задаче, грамотно реализовать процесс сбора требуемых данных и информации, их обработки и интерпретации.
		2. Обосновывает существование происходящего, выявляет закономерности,	Знать: возможности языка программирования Python для решения математических и научных задач.

		понимает природу вариабельности.	Уметь: технологии создания программных решений на современных языках программирования.
		3. Формирует признак классификации, выделяет соответствующие ему группы однородных «объектов», идентифицирует общие свойства элементов этих групп, оценивает полноту результатов классификации, показывает прикладное назначение классификационных групп.	Знать: возможности языка программирования Python для решения математических и научных задач. Уметь: технологии создания программных решений на современных языках программирования.
		4. Грамотно, логично, аргументировано формирует собственные суждения и оценки. Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности.	Знать: фактологические основы формирования суждений и оценок. Уметь: формализовывать постановку прикладных задач исследования с целью программирования решения.
		5. Аргументированно и логично представляет свою точку зрения посредством и на основе системного описания.	Знать: системное описание процессов и явлений от простого к сложному, от общего к частному. Уметь: формализовывать постановку прикладных задач исследования с целью программирования решения.

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

Таблица 2

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 1 (в часах)	Семестр 2 (в часах)
Общая трудоемкость дисциплины	6/216	108	108
Контактная работа- Аудиторные занятия	100	50	34
Лекции	0	0	0
Семинары, практические занятия	84	50	34

Самостоятельная работа	132	58	74
Вид текущего контроля	контрольные работы	контрольная работа	контрольная работа
Вид промежуточной аттестации	1-Зачет 2 -экзамен	Зачет	Экзамен

5.2 Учебно-тематический план

№ п/п	Наименование тем (разделов) дисциплины	Трудоемкость в часах					Формы текущего контроля успеваемости
		Всего	Контактная работа - Аудиторная работа			Самостоят ельная работа	
			Общая, в т.ч.:	Лекции	Семинары, практические занятия		
1	Введение в программирование на Python	16	8	-	8	8	Обсуждение теоретических вопросов. Решение задач (программирование) по тематике практического занятия.
2	Управляющие конструкции, списки и кортежи	16	8	-	8	8	
3	Словари, множества и выражения-генераторы	18	8	-	8	10	
4	Функции	18	8	-	8	10	
5	Работа с файлами и обработка исклю- чительных ситуаций	18	8	-	8	10	
6	Модули и пакеты	20	10	-	10	10	
7	Введение в объектно- ориентированное программирование	20	8	-	6	12	
8	Объектно- ориентированное программирование в Python	20	8	-	6	12	
9	Введение в функциональное программирование	16	4	-	4	12	
10	Функциональное программирование в Python	18	6	-	6	12	
11	Структуры данных: массивы, стеки, очереди, списки	24	8	-	6	14	
12	Алгоритмы поиска и сортировки	24	8	-	6	14	
	В целом по дисциплине	216	84	-	84	132	Согласно учебному плану: контрольные работы
Итого в %:		100	39	-	39	54	

5.3. Содержание семинаров, практических занятий

Таблица 4

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарских, практических занятиях, рекомендуемые источники из разделов 8, 9 (указывается раздел и порядковый номер источника)	Формы проведения занятий
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; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Управляющие конструкции, списки и кортежи	Базовые числовые типы, строки, списки, словари, переменные, базовые операторы. Рекомендуемые источники: 8.1; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
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; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Функции	Создание функций, область видимости переменной, передача аргументов в функцию. Лямбда-функции. Рекомендуемые источники: 8.1; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Работа с файлами и обработка исключительных ситуаций	Исключения. Инструкция try...except...else...finally. Классы встроенных исключений. Создание пользовательских исключений. Инструкция assert. Работа с файлами в Python, операции с файлами: открытие/закрытие файла, чтение и записи и другие методы для работы с файлами. Инструкция with ... as и ее использование для файлов. Рекомендуемые источники: 8.1; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений

Модули и пакеты	Устройство модулей и пакетов, инструкции <code>import</code> и <code>from</code> . Создание собственных модулей и пакетов. Рекомендуемые источники: 8.1; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
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; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
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; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
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; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
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; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Структуры данных: массивы, стеки, очереди, списки	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. Рекомендуемые источники: 8.1; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений

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; 8.2; 8.3	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 5

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Введение в программирование на Python	Среда программирования. Использование документации.	Работа с литературой. Выполнение заданий для самостоятельной работы
Управляющие конструкции, списки и кортежи	Оперирование вложенными списками, копирование списков, некоторые операции над списками.	Работа с литературой. Выполнение заданий для самостоятельной работы
Словари, множества и выражения-генераторы	Выражения-генераторы в Python. Выражения-генераторы для словарей и множеств: семантика, синтаксис и практическое использование.	Работа с литературой. Выполнение заданий для самостоятельной работы
Функции	Возвращение нескольких значений из функции. Распаковка и запаковка параметров функции.	Работа с литературой. Выполнение заданий для самостоятельной работы
Работа с файлами и обработка исключительных ситуаций	Сохранение объектов в файл с помощью модуля pickle и shelve. Модуль CSV.	Работа с литературой. Выполнение заданий для самостоятельной работы
Модули и пакеты	Написание и запуск скриптов на Python. Установка модулей из глобального репозитория.	Работа с литературой. Выполнение заданий для самостоятельной работы
Введение в объектно-ориентированное программирование	Базовые типы в Python: взгляд с точки зрения ООП. Методы базовых типов.	Работа с литературой. Выполнение заданий для самостоятельной работы
Объектно-ориентированное программирование в Python	Проверка принадлежности к классу и интроспекция в Python.	Работа с литературой. Выполнение заданий для самостоятельной работы
Введение в	Реализация декораторов с	Работа с литературой.

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
функциональное программирование	параметрами в Python.	Выполнение заданий для самостоятельной работы
Функциональное программирование в Python	Функции-генераторы и выражения-генераторы в Python.	Работа с литературой. Выполнение заданий для самостоятельной работы
Структуры данных: массивы, стеки, очереди, списки	Реализация различных вариантов связанных списков на Python.	Работа с литературой. Выполнение заданий для самостоятельной работы
Алгоритмы поиска и сортировки	Сравнение различных сортировок (простых и эффективных) с использованием их реализаций на Python.	Работа с литературой. Выполнение заданий для самостоятельной работы

6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю

На практических занятиях используются методические материалы, выдаваемые преподавателем в виде файлов, а также материалы с сайтов, указанных в разделах 8 и 9 («Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины» и «Перечень ресурсов информационно-телекоммуникационной сети Интернет, необходимых для освоения дисциплины»).

Задания для самостоятельной работы содержатся в методических материалах, выдаваемых преподавателем на практических занятиях.

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,мое'

Контрольная работа 2

3. 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.

4. Реализовать декоратор с именем `not_none`, который генерирует исключительную ситуацию если декорируемая функция вернула значения `None`. (20 баллов)

5. 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'}`

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».

7.1 Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, знаний и умений

Компетенция УК-4 Способность использовать прикладное программное обеспечение при решении профессиональных задач

Индикатор 1 Использует основные методы и средства получения, представления, хранения и обработки данных.

Знать: типы и структуры данных, используемые в языке Python.

Уметь: выбирать структуры данных и алгоритмы, позволяющие решить поставленную задачу наиболее оптимальным способом.

Задание

Напишите скрипт, реализующий следующую последовательность действий:

1. Генерация исходных данных по поставкам и продажам магазина (Товар1, ... Товар N, по дням недели) в текстовые файлы `in.txt`, `out.txt`
2. Копирование файлов с исходными данными в папку выполнения расчета
3. Считывание исходных данных из файлов в переменные среды R.

Индикатор 2 Демонстрирует владение профессиональными пакетами прикладных программ.

Знать: типы и структуры данных, используемые в языке Python.

Уметь: выбирать алгоритмы, позволяющие решить поставленную задачу наиболее оптимальным способом.

Задание

Имеется список строк вида «дд.мм.гггг», например, «02.02.1988» или «22.06.2016». Создайте новый список, в котором элементы будут отсортированы в порядке убывания года.

Индикатор 3. Выбирает необходимое прикладное программное обеспечение в зависимости от решаемой задачи.

Знать: технологии обработки, анализа и интерпретации данных различной природы.

Уметь: применять алгоритмы для поиска и выявления зависимостей в данных.

Задание

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.

Индикатор 4. Использует прикладное программное обеспечение для решения конкретных прикладных задач.

Знать: технологии обработки, анализа и интерпретации данных различной природы.

Уметь: применять алгоритмы для поиска и выявления зависимостей в данных.

Задание

Напишите функцию с 1 параметром, которая рассчитывает ставку налога на легковой автомобиль в зависимости от мощности двигателя по следующему правилу:

до 100 л.с. включительно - 2,5 руб.

свыше 100 л.с. до 150 л.с. включительно - 3,5 руб.

свыше 150 л.с. до 200 л.с. включительно - 5,0 руб.

свыше 200 л.с. до 250 л.с. включительно - 7,5 руб.

свыше 250 л.с. - 15,0 руб.

Мощность двигателя является параметром функции. Функция возвращает найденную ставку налога.

Вызовите функцию несколько раз с различными значениями параметров.

Компетенция	УК-10	Способность	осуществлять	поиск,
критически	анализировать,	обобщать	и	систематизировать
информацию,	использовать	системный	подход	для
поставленных задач				решения

Индикатор 1. Четко описывает состав и структуру требуемых данных и информации, грамотно реализует процессы их сбора, обработки и интерпретации

Знать: возможности языка программирования Python для решения математических и научных задач.

Уметь: четко описывать состав и структуру данных и информации, используемых в решаемой задаче, грамотно реализовать процесс сбора требуемых данных и информации, их обработки и интерпретации.

Задание

Напишите скрипт, реализующий последовательность действий по расчету экономических показателей:

$Pr = TR - TC$ (где TR (totalrevenue) – выручка от продажи товара

TC (totalcost) – затраты

$TR = Q_sale * P$ (где Q_sale (quatitysale) – количество проданных контейнеров

P (price) – розничная цена контейнера

$TC = Q_supply * P_supply + Q_util * P_util$

где Q_supply – количество закупленных магазином контейнеров

P_supply – цена закупки контейнера магазином

Q_util – количество непроданных контейнеров

P_util – цена утилизации контейнера

Индикатор 2. Обосновывает сущность происходящего, выявляет закономерности, понимает природу вариабельности

Знать: возможности языка программирования Python для решения математических и научных задач.

Уметь: технологии создания программных решений на современных языках программирования.

Задание

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.

Индикатор 3. Формулирует признак классификации, выделяет соответствующие ему группы однородных «объектов», идентифицирует общие свойства элементов этих групп, оценивает полноту результатов классификации, показывает прикладное назначение классификационных групп.

Знать: возможности языка программирования Python для решения математических и научных задач.

Уметь: технологии создания программных решений на современных языках программирования.

Задание

Напишите скрипт, реализующий последовательность действий по

табличному и графическому анализу выручки от продажи Товара 1 в течение недели.

Индикатор 4. Грамотно, логично, аргументировано формирует собственные суждения и оценки. Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности.

Знать: фактологические основы формирования суждений и оценок.

Уметь: формализовывать постановку прикладных задач исследования с целью программирования решения.

Задание

Имеется текстовый файл, содержащий информацию о товарах и ценах на них. Каждая строка файла имеет вид: НАЗВАНИЕ ТОВАРА: ЦЕНА. Используя данный файл, найдите цену указанного товара, или выдайте сообщение о том, что цена не известна.

Индикатор 5. Аргументированно и логично представляет свою точку зрения посредством и на основе системного описания.

Знать: системное описание процессов и явлений от простого к сложному, от общего к частному.

Уметь: формализовывать постановку прикладных задач исследования с целью программирования решения.

Задание

Создайте класс Покупка, имеющий:

- закрытый атрибут L - список кортежей, в котором хранится информация о купленных товарах. Кортеж содержит название, цену и количество товара;
- метод init;
- метод put - добавляет в список покупок новый товар. Метод имеет 3 параметра: название, цена и количество товара;
- метод Стоимость - вычисляет стоимость покупки;
- метод Инфо - выводит на экран информацию о покупке: названия товаров, их цену и количество.

Создайте экземпляр класса, продемонстрируйте работу с методами.

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: Бизнес-информатика

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

1. Создание базового окна. Элементы графического интерфейса (виджеты).
2. Модули. Инструкции import и from. Обзор стандартной библиотеки Python.
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)

7.2. Описание показателей и критериев оценивания компетенций, описание шкалы оценивания

Порядок проведения промежуточной аттестации и система оценивания результатов промежуточной аттестации по итогам семестра (модуля) по программам бакалавриата и магистратуры в соответствии с учебными планами по направлениям подготовки высшего образования утверждена приказом ректора Финуниверситета по основной деятельности № 0557/о от 23.03.2017.

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

Основная литература

1. Жуков Р. А. Язык программирования Python: практикум [Электронный ресурс]: учебное пособие. М.: ИНФРА-М, 2021. 216 с. ISBN 978-5-16-016971-2. <https://znanium.com/catalog/product/1412168> (дата обращения: 26.05.2021).

Дополнительная литература

2. Гуриков С. Р. Основы алгоритмизации и программирования на Python [Электронный ресурс]: учебное пособие. М.: ФОРУМ; ИНФРА-М, 2020. 343 с. ISBN 978-5-00091-487-8. <https://znanium.com/catalog/product/1206074> (дата обращения: 26.05.2021).

3. Федоров Д. Ю. Программирование на языке высокого уровня Python [Электронный ресурс]: учебное пособие. 2-е изд., перераб. и доп. М.: Юрайт, 2021. 161 с. ISBN 978-5-534-10971-9. <https://ez.el.fa.ru:2428/bcode/472985> (дата обращения: 26.05.2021).

4. Гниденко И. Г., Павлов Ф. Ф., Федоров Д. Ю. Технологии и методы программирования [Электронный ресурс]: учебное пособие. М.: Юрайт, 2021. 235 с. ISBN 978-5-534-02816-4. <https://ez.el.fa.ru:2428/bcode/469759> (дата обращения: 26.05.2021).

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

1. Сайт Департамента анализа данных и машинного обучения Финансового университета.

2. Электронная библиотека Финансового университета (ЭБ) <http://elibrary.ru/>

3. Электронно-библиотечная система BOOK.RU <http://www.book.ru>

4. Электронно-библиотечная система «Университетская библиотека ОНЛАЙН» <http://biblioclub.ru/>
5. Электронно-библиотечная система Znanium <http://www.znanium.com>
6. Электронно-библиотечная система издательства «ЮРАЙТ» <https://www.urait.ru/>
7. Электронно-библиотечная система издательства «Лань» <https://e.lanbook.com/>
8. Деловая онлайн-библиотека Alpina Digital <http://lib.alpinadigital.ru/>
9. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>
10. 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;
- выполнение расчетно-аналитической работы;
- индивидуальные или групповые консультации по наиболее сложным вопросам;
- подготовка к экзамену.

В рамках изучения предусмотрено выполнение расчетно-аналитической работы.

Студенты, не выполнившие расчетно-аналитическую работу или не прошедшие собеседование по ней, к экзамену по дисциплине не допускаются.

Организация выполнения расчетно-аналитической 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 расчетно-аналитической 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 расчетно-аналитической работы общим требованиям.

Расчетно-аналитическая работа оформляется в тетради или на компьютере с использованием текстового редактора:

- гарнитура шрифта – Times New Roman;
- размер шрифта – 14;
- межстрочный интервал – полуторный;
- размеры полей: левого – 3 см, верхнего – 1,5, правого – 1, нижнего – 1,5 см;
- ориентация – книжная;
- форматирование основного текста и ссылок – по ширине;
- цвет шрифта – черный;
- абзацный отступ – 1,5 см.
- нумерация страниц – справа, внизу.

Объем расчетно-аналитической работы не должен превышать 25 страниц машинописного текста формата А4.

Каждую структурную часть расчетно-аналитической работы нужно начинать с нового листа. Точка в конце заголовка структурной части работы не ставится.

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.

Расчетно-аналитическая работа представляется на рецензирование в сброшюрованном виде (листы должны быть скреплены по левому краю).

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

1. Компьютерные программы общего назначения: Windows, Microsoft Office.
2. Антивирус Kaspersky EndPoint Security.

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. Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>
4. Система комплексного раскрытия информации «СКРИН» -
<http://www.skrin.ru/>
5. Федеральная служба государственной статистики: <http://www.gks.ru>
6. Информационная система СПАРК.
7. Информационная система Bloomberg.
8. Информационная система Thomson Reuters

11.3. Сертифицированные программные и аппаратные средства защиты информации

Не используются.

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

Образовательный процесс по учебной дисциплине осуществляется в учебных аудиториях для проведения учебных занятий. При освоении дисциплины используются технические средства мультимедийной техники

аудиторий. Для проведения учебных занятий используются аудитории, оборудованные следующими техническими средствами: мультимедиа-проектором, экраном настенным или доской интерактивной с встроенной акустической системой для воспроизведения аудио файла, персональным (-ными) компьютером (-рами) с доступом к Internet-ресурсам и электронно-образовательной среде Финансового университета.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения в сети «Интернет» и обеспечением доступа в электронно-образовательную среду Финансового университета.