

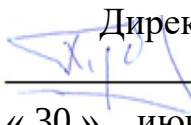
**Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финуниверситет)**

Владикавказский филиал Финуниверситета

Кафедра «Корпоративные инфокоммуникационные системы»

УТВЕРЖДАЮ

Директор филиала

 Т.А. Хубаев

« 30 » июня 2023 г.

М.А. Ковалева

Алгоритмы и структуры данных в языке Python

Оценочные материалы для проведения текущего контроля успеваемости и
промежуточной аттестации студентов,
обучающихся по направлению подготовки
38.03.05 – Бизнес-информатика,
образовательная программа «Цифровая трансформация управления бизнесом»,
профиль «ИТ- менеджмент в бизнесе»

Одобрено кафедрой «Корпоративные инфокоммуникационные системы»

Протокол от «22» июня 2023 г. № 12.

Владикавказ 2023

СОДЕРЖАНИЕ

1.Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	3
2. Показатели и критерии оценивания компетенций на различных этапах их формирования, описание шкал оценивания.....	4
3. Оценочный лист результатов обучения к дисциплине	7
4. Контрольные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний	18

1.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
УК-4	Способность использовать прикладное программное обеспечение при решении профессиональных задач	1. Использует основные методы и средства получения, представления, хранения и обработки данных.	1. Знать: методы и средства получения, представления, хранения и обработки данных. Уметь: получать, представлять, хранить и обрабатывать данные.
		2. Демонстрирует владение профессиональными пакетами прикладных программ.	2. Знать: основные профессиональные пакеты прикладных программ. Уметь: использовать профессиональные пакеты прикладных программ.
		3. Выбирает необходимое прикладное программное обеспечение в зависимости от решаемой задачи.	3.Знать: рынок прикладного ПО. Уметь: формировать критерии выборы и осуществлять выбор ПО.
		4. Использует прикладное программное обеспечение для решения конкретных прикладных задач	4.Знать: прикладное ПО. Уметь: решать конкретные прикладные задачи
УК-10	Способность осуществлять поиск, критически анализировать, обобщать и систематизировать информацию, использовать системный подход для решения	1. Четко описывает состав и структуру требуемых данных и информации, грамотно реализует процессы их сбора, обработки и интерпретации	1.Знать: процессы сбора, обработки и интерпретации информации Уметь: описывать описывает состав и структуру требуемых данных и информации
		2.Обосновывает сущность происходящего, выявляет закономерности, понимает природу вариабельности	2.Знать: принципы описания закономерностей и вариабельности Уметь: обосновывать сущность происходящего

	поставленных задач	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: подходы к выполнению классификации Уметь: выполнять классификацию и оценивать полноту выполнения классификации.
		4.Грамотно, логично, аргументировано формирует собственные суждения и оценки. Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности	Знать: методы аргументации и оценкам. Уметь: формировать суждения и оценки.
		5. Аргументированно и логично представляет свою точку зрения посредством и на основе системного описания.	Знать: подходы к описанию собственного мнения. Уметь: представлять свою точку зрения.

2. Показатели и критерии оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Уровень сформированности компетенций			
«Минимальный уровень не достигнут» Компетенции не сформированы. Знания отсутствуют, умения и навыки не сформированы.	«Минимальный уровень» Компетенции сформированы. Сформированы базовые структуры знаний. Умения фрагментарны и носят репродуктивный характер. Демонстрируется низкий уровень самостоятельности практического навыка.	«Средний уровень» Компетенции сформированы. Знания обширные, системные. Умения носят репродуктивный характер, применяются к решению типовых заданий. Демонстрируется достаточный уровень самостоятельности устойчивого практического навыка.	«Высокий уровень» Компетенции сформированы. Знания твердые, аргументированные, всесторонние. Умения успешно применяются к решению как типовых, так и нестандартных творческих заданий. Демонстрируется высокий уровень самостоятельности, высокая адаптивность практического навыка
Описание критериев оценивания			

Обучающийся демонстрирует: - существенные пробелы в знаниях учебного материала; - допускаются принципиальные ошибки при ответе на основные вопросы билета, отсутствует знание и понимание основных понятий и категорий; - непонимание сущности дополнительных вопросов в рамках заданий билета; - отсутствие умения выполнять практические задания, предусмотренные программой дисциплины; - отсутствие готовности (способности) к дискуссии и низкую степень контактности.	Обучающийся демонстрирует: - знания теоретического материала; - неполные ответы на основные вопросы, ошибки в ответе, недостаточное понимание сущности излагаемых вопросов; - неуверенные и неточные ответы на дополнительные вопросы; - недостаточное владение литературой, рекомендованной программой дисциплины; - умение без грубых ошибок решать практические задания, которые следует выполнить.	Обучающийся демонстрирует: - знание и понимание основных вопросов контролируемого объема программного материала; - твердые знания теоретического материала. - способность устанавливать и объяснять связь практики и теории, выявлять противоречия, проблемы и тенденции развития; - правильные и конкретные, без грубых ошибок, ответы на поставленные вопросы; - умение решать практические задания, которые следует выполнить; - владение основной литературой, рекомендованной программой дисциплины; - наличие собственной обоснованной позиции по обсуждаемым вопросам. Возможны незначительные оговорки и неточности в раскрытии отдельных положений вопросов билета, присутствует неуверенность в ответах на	Обучающийся демонстрирует: - глубокие, всесторонние и аргументированные знания программного материала; - полное понимание сущности и взаимосвязи рассматриваемых процессов и явлений, точное знание основных понятий в рамках обсуждаемых заданий; - способность устанавливать и объяснять связь практики и теории; - логически последовательные, содержательные, конкретные и исчерпывающие ответы на все задания билета, а также дополнительные вопросы экзаменатора; - умение решать практические задания; - свободное использование в ответах на вопросы материалов рекомендованной основной и дополнительной литературы.
--	---	--	---

Оценка «неудовлетворите льно»	Оценка «удовлетворительно» / «зачтено»	Оценка «хорошо» / «зачтено»	Оценка «отлично» / «зачтено»
--	---	--	---

3. Оценочный лист результатов обучения к дисциплине

Примерные виды работ обучающегося, формирующие текущий контроль по дисциплине

№ п/п	Вид учебной деятельности	Баллы	Максимум за семестр (модуль)
1	Ведение конспекта лекции / семинара и работа с ним	0,1	2
2	Активное вовлечение в интерактивный процесс	0,3	3
3	Ответы на вопросы на семинарском занятии	0,3	5
4	Реферативный обзор дополнительного материала по теме	0,3	3
5	Написание реферата по заданной теме	0,3	3
6	Аналитический разбор научных публикаций по проблеме	0,4	5
7	Анализ конкретной ситуации и подготовка аналитической записки	0,4	4
8	Выполнение и сдача контрольной работы	0,8	4
9	Блиц-опрос по теме	0,2	2
10	Аудиторная контрольная работа	0,3	3
11	Презентация по заданной теме (разделу)	0,4	4
12	Выполнение домашней контрольной работы	0,2	2
...			
	Всего за семестр (модуль)		40

Критерии балльной оценки включают качество подготовки обучающихся к семинарским занятиям, выполнения различных видов самостоятельной работы (при этом результаты регулярно проводимого экспресс-контроля успеваемости обучающихся должны составлять не менее 50% балльной оценки), а также посещение аудиторных занятий (не более 15% балльной оценки).

При подведении итогов текущего контроля успеваемости в середине семестра его результаты в 20-балльной оценке вносятся в ведомости текущего контроля успеваемости и использованием ТЭКУ (в филиалах – в ведомости текущего контроля успеваемости) и на соответствующей странице предметно-группового журнала преподавателям, ведущим аудиторные занятия.

Обучающийся, получивший от 7 до 20 баллов, считается аттестованным, получивший от 0 до 6 баллов – не аттестованным.

Результаты текущего контроля успеваемости за вторую половину семестра / в модуле преподаватели, ведущие аудиторные занятия, сообщают обучающимся на последнем занятии и вносят в ведомости (в колонку «баллы, работа в семестре») в день проведения зачета/экзамена.

Итоговая сумма баллов, полученная обучающимся по результатам текущего контроля успеваемости в семестре/модуле и промежуточной аттестации, преобразуется в пятибалльную систему оценок.

Порядок перевода 100-бальной оценки в пятибалльную

100-бальная система	5-бальная система
86-100	отлично
70-85	хорошо
50-69	удовлетворительно
менее 50	неудовлетворительно
50-100	зачтено
менее 50	не зачтено

4. Контрольные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний

Тема 1. Введение в программирование на Python

Тематика текущего опроса, презентаций, дискуссий УК-4(1,2,3,4), УК-10(1,2,3,4,5)

1. Python - мультипарадигмальный язык программирования
2. Итераторы для словарей
3. Ключи и значения
4. Список построений
5. Синтаксис понимания
6. Синтаксис нарезки списка
7. Изменение списка

Контрольные вопросы по теме семинарского занятия УК-4(1,2,3,4), УК-10(1,2,3,4,5)

Установка Python, установка дистрибутива Anaconda.

Работа в интерактивном режиме интерпретатора.

Интерактивная оболочка IPython notebook: принципы работы и применения

Практическое задание 1. УК-4(1,2,3,4), УК-10(1,2,3,4,5)

Установите пакет Anaconda запустите программу Jupyter Notebook составьте тестовый код на Python

Тема 2. Управляющие конструкции, списки и кортежи

**Тематика текущего опроса, презентаций, дискуссий УК-4(1,2,3,4),
УК-10(1,2,3,4,5)**

1. Анонимные функции
2. Перебор словарей
3. Понимание понимания
4. Введение в Python
5. Строки, списки и кортежи

**Контрольные вопросы по теме семинарского занятия УК-4(1,2,3,4),
УК-10(1,2,3,4,5)**

Базовые числовые типы, строки, списки, словари, переменные, базовые операторы.

Практическое задание 1. УК-4(1,2,3,4), УК-10(1,2,3,4,5)

Есть список `a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]`.

Выведите все элементы, которые меньше 8..

Практическое задание 2. УК-4(1,2,3,4), УК-10(1,2,3,4,5)

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'`

Тема 3. Словари, множества и выражения-генераторы

**Тематика текущего опроса, презентаций, дискуссий УК-4(1,2,3,4),
УК-10(1,2,3,4,5)**

1. Словари и наборы
2. Условное выполнение и циклы
3. Понимание
4. Функции
5. Модули
6. Области и пространства имен
7. Промежуточный Python
8. Обработка файлов

9. Объектно-ориентированное программирование
10. Итератор, генератор, декораторы
11. Лямбда-выражения
12. Регулярное выражение

**Контрольные вопросы по теме семинарского занятия УК-4(1,2,3,4),
УК-10(1,2,3,4,5)**

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.

Практическое задание 1. УК-4(1,2,3,4), УК-10(1,2,3,4,5)

Определить минимальное и максимальное значение среди чисел a, b, c, d.

Практическое задание 2. УК-4(1,2,3,4), УК-10(1,2,3,4,5)

Анаграммами называются слова, образованные путем взаимной перестановки букв. В английском языке, например, анаграммами являются слова «live» и «evil», а в русском – «выбор» и «обрыв». Напишите программу, которая будет запрашивать у пользователя два слова, определять, являются ли они анаграммами, и выводить на экран ответ.

Тема 4. Функции

**Тематика текущего опроса, презентаций, дискуссий УК-4(1,2,3,4),
УК-10(1,2,3,4,5)**

1. Функции
2. Модули
3. Области и пространства имен
4. Объявление и вызов функции в Python.
5. Параметры функции со значением по умолчанию и комментирование функции.

6. Получение информации о функции.
7. Способы передачи параметров при вызове функции.

**Контрольные вопросы по теме семинарского занятия УК-4(1,2,3,4),
УК-10(1,2,3,4,5)**

Создание функций, область видимости переменной, передача аргументов в функцию.

Лямбда-функции.

Практическое задание 1. УК-4(1,2,3,4), УК-10(1,2,3,4,5)

По номеру года - 2022 определите, является ли данный год високосным.

Тема 5. Работа с файлами и обработка исключительных ситуаций

**Тематика текущего опроса, презентаций, дискуссий УК-4(1,2,3,4),
УК-10(1,2,3,4,5)**

1. Обработка файлов
2. Итератор, генератор, декораторы
3. Базовые операции для работы с файлами в Python.
4. Использование инструкции with ... as на примере работы с файлами.

**Контрольные вопросы по теме семинарского занятия УК-4(1,2,3,4),
УК-10(1,2,3,4,5)**

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..

Практическое задание 1. УК-4(1,2,3,4), УК-10(1,2,3,4,5)

С клавиатуры вводятся n целых чисел. Найти минимальное положительное и максимальное отрицательное числа.

Тема 6. Модули и пакеты

**Тематика текущего опроса, презентаций, дискуссий УК-4(1,2,3,4),
УК-10(1,2,3,4,5)**

1. Что такое декоратор?
2. Определите класс car с двумя атрибутами: color и speed. Затем создайте экземпляр и верните speed
3. В чем разница между методами экземпляра, класса и статическими методами в Python?
4. Переменные в Python передаются по ссылке или по значению?
5. Как объединить списки в Python?

**Контрольные вопросы по теме семинарского занятия УК-4(1,2,3,4),
УК-10(1,2,3,4,5)**

Устройство модулей и пакетов, инструкции import и from.

Создание собственных модулей и пакетов

Практическое задание 1. УК-4(1,2,3,4), УК-10(1,2,3,4,5)

Напишите программу, которая для введенного n вычисляет значение n! (не использовать функции Python).

Практическое задание 2. УК-4(1,2,3,4), УК-10(1,2,3,4,5)

Реализовать декоратор с именем not_none, который генерирует исключительную ситуацию если декорируемая функция вернула значения None

Тема 7. Введение в объектно-ориентированное программирование

**Тематика текущего опроса, презентаций, дискуссий УК-4(1,2,3,4),
УК-10(1,2,3,4,5)**

1. Объявление класса, конструктор, создание объектов и одиночное наследование в Python. Управление доступом к атрибутам класса.
2. Полиморфизм и утиная типизация и проверка принадлежности объекта к классу в языке Python.
3. Методы классов и статические переменные и методы.
4. Специальные методы для использования пользовательских классов со стандартными операторами и функциями.

5. Объявление класса, конструктор, создание объектов, наследование
6. Управление доступом к атрибутам класса в Python.
7. Полиморфизм и утиная типизация и проверка принадлежности объекта к классу
8. Методы классов и статические переменные и методы.
9. Специальные методы для использования пользовательских классов со стандартными операторами и функциями.

Контрольные вопросы по теме семинарского занятия УК-4(1,2,3,4), УК-10(1,2,3,4,5)

Базовые возможности ООП в Python: создание классов и объектов; наследование и полиморфизм; функция `super()`.

Практическое задание 1. УК-4(1,2,3,4), УК-10(1,2,3,4,5)

Напишите скрипт на Python обрабатывающий текстовый файл так, что в нем все заглавные буквы преобразуются в строчные.

Практическое задание 2. УК-4(1,2,3,4), УК-10(1,2,3,4,5)

В магазине была объявлена скидка размером 60 % на ряд товаров, и для того чтобы покупатели лучше ориентировались, владелец торговой точки решил вывесить отдельную таблицу со скидками с указанием уцененных товаров и их оригинальных цен. Используйте цикл для создания подобной таблицы, в которой будут исходные цены, суммы скидок и новые цены для покупок на сумму \$4,95, \$9,95, \$14,95, \$19,95 и \$24,95. Убедитесь в том, что суммы скидки и новые цены отображаются с двумя знаками после запятой..

Тема 8. Объектно-ориентированное программирование в Python

Тематика текущего опроса, презентаций, дискуссий УК-4(1,2,3,4), УК-10(1,2,3,4,5)

1. Объектно-ориентированное программирование
2. Что такое декоратор?
3. Каковы преимущества использования Python?
4. Итератор, генератор, декораторы

5. Лямбда-выражения
6. Регулярное выражение

**Контрольные вопросы по теме семинарского занятия УК-4(1,2,3,4),
УК-10(1,2,3,4,5)**

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

Практическое задание 1. УК-4(1,2,3,4), УК-10(1,2,3,4,5)

Выберите библиотеку Python для хэширования данных в целях быстрого поиска текстовых строк

Тема 9. Введение в функциональное программирование

**Тематика текущего опроса, презентаций, дискуссий УК-4(1,2,3,4),
УК-10(1,2,3,4,5)**

1. Основные возможности, поддерживаемые функциональными языками программирования. Поддержка элементов функционального программирования в Python.
2. Глобальные и локальные переменные в функциях на примере Python. Побочные эффекты вызова функций и их последствия.
3. Вложенные функции и замыкания, специфика реализации в Python.
4. Как выполняется обработка исключений в Python?
5. Каковы преимущества использования Python?
6. Как писать комментарии на Python?

**Контрольные вопросы по теме семинарского занятия УК-4(1,2,3,4),
УК-10(1,2,3,4,5)**

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

Практическое задание 1. УК-4(1,2,3,4), УК-10(1,2,3,4,5)

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

Практическое задание 2. УК-4(1,2,3,4), УК-10(1,2,3,4,5)

1. Создайте класс Заказ (Order), у которого есть свойства код товара (code), цена (price), количество (count) и методы `__init__` и `__str__`.

Тема 10. Функциональное программирование в Python

Тематика текущего опроса, презентаций, дискуссий УК-4(1,2,3,4), УК-10(1,2,3,4,5)

1. Назовите изменяемые и неизменяемые объекты
2. Какая разница между словарями и JSON?
3. В чем разница между `pass`, `continue` и `break`?
4. Преобразуйте следующий цикл `for` в генератор списков (list comprehension)
5. Как перевести строку в верхний/нижний регистр?

Контрольные вопросы по теме семинарского занятия УК-4(1,2,3,4), УК-10(1,2,3,4,5)

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

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

Практическое задание 1. УК-4(1,2,3,4), УК-10(1,2,3,4,5)

Опишите структуру CSV файла фиксирующего факты продаж, необходимого для подсчета суммарных продаж по различным географическим подразделениям фирмы

Практическое задание 2. УК-4(1,2,3,4), УК-10(1,2,3,4,5)

Напишите функцию, которая для заданного радиуса r вычисляет площадь круга и длину окружности. Функция возвращает кортеж из 2 значений.

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

Тематика текущего опроса, презентаций, дискуссий УК-4(1,2,3,4), УК-10(1,2,3,4,5)

1. Как объединить списки в Python?
2. В чем разница между списками и массивами?
3. Как объединить два массива?
4. *Как прокомментировать несколько строк в Python?*
5. Связанные списки: однонаправленные и двунаправленные – принцип реализации.
6. Абстрактная структура данных стек и очередь: базовые и расширенные операции, их сложность.

Контрольные вопросы по теме семинарского занятия УК-4(1,2,3,4), УК-10(1,2,3,4,5)

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.

Практическое задание 1. УК-4(1,2,3,4), УК-10(1,2,3,4,5)

Опишите закономерности/вариабельность в синтаксических конструкциях Python для работы со списками и словарями.

Тема 12. Алгоритмы поиска и сортировки

Тематика текущего опроса, презентаций, дискуссий УК-4(1,2,3,4), УК-10(1,2,3,4,5)

1. Форматирование даты и времени.
2. Форматирование строк.
3. Форматы чисел.
4. Функции и методы для работы со строками.
5. Числовые литералы.
6. Алгоритм обменной сортировки, сложность сортировки и возможности по ее улучшению.
7. Алгоритм сортировки выбором, сложность сортировки и возможности по ее улучшению.
8. Алгоритм сортировки вставками, его сложность. Алгоритм быстрого поиска в отсортированном массиве. Сложность поиска в отсортированном и не отсортированном массиве.
9. Алгоритм сортировки Шелла, сложность сортировки и возможности по ее улучшению.

Контрольные вопросы по теме семинарского занятия УК-4(1,2,3,4), УК-10(1,2,3,4,5)

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.

Практическое задание 1. УК-4(1,2,3,4), УК-10(1,2,3,4,5)

Опишите подходы используемых в Python для группировки функций ориентированных на обработку однотипных объектов и решение близких задач

Перечень заданий для выполнения контрольной работы

1. Составить в интерпретаторе Python программу которая:
 - просит пользователя ввести целое число и присваивает его переменной num

- просит пользователя ввести множитель для возведения в степень и присваивает его переменной `step`
- проверяет истинность условия, что введенное пользователем целое число `num` меньше 100
- если это условие ИСТИННО, то необходимо возвести число `num` в степень `step` (напоминаю, возведение в степень осуществляется оператором `**`) и присвоить результат переменной `result`.

Результат вывести на печать, если результат проверки «ложь», то вывести на печать сообщение: "Введенное вами число > 100"

2. Напишите программу, находящую коэффициенты многочлена n -й степени, проходящего через $n+1$ точку:

Программа должна принимать на вход в первой строке натуральное число M ($M=n+1$) после этого в цикле M строк. Каждая из M строк содержит действительных числа: x_i, y_i - координаты M точек, через которые проходит график многочлена.

На выходе программа должна выдавать через пробел $n+1$ ($n+1=M$) коэффициент многочлена: $c_0, c_1, c_2 \dots c_n$.

3. Напишите программу, которая будет решать систему линейных уравнений вида:

$$\begin{cases} a_{11}x + a_{12}y = b_1 \\ a_{21}x + a_{22}y = b_2 \end{cases}$$

На вход программа должна принимать 2 строки (используйте стандартную функцию `input()` Python). Каждая строка содержит 3 действительных числа через пробел: a_{i1}, a_{i2}, b_i , где i - номер строки.

На выходе программа должна выдавать значения x и y в одну строку через пробел.

4. Напишите функцию `f2`, которая имеет 2 параметра: первый (`p1`) – ссылка на функцию с 1 аргументом, которая возвращает число, второй (`p2`) – число, которое по умолчанию равно 1. Функция `f2` выводит на экран значение `p1` в точке `p2`.

5. Создайте класс `Товар`, в котором имеются атрибуты:

- `name` – название товара (закрытый);
- `count` – количество;
- `price` – цена (закрытый).

Создайте конструктор класса.

Создайте метод `discount`, который уменьшает цену на количество процентов, заданных в качестве параметра метода.

Создайте экземпляр класса `Товар` (`name` – Тетради, `count` – 100, `price` – 85).

Уменьшите количество товара на 25 (используем атрибут), а цену – на 10% (вызовите метод).

6. 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

7. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)

Напишите программу, которая будет решать систему линейных уравнений вида:

$$\begin{cases} a_{11}x + a_{12}y = b_1 \\ a_{21}x + a_{22}y = b_2 \end{cases}$$

На вход программа должна принимать 2 строки (используйте стандартную функцию input() Python). Каждая строка содержит 3 действительных числа через пробел: a_{i1}, a_{i2}, b_i , где i - номер строки.

На выходе программа должна выдавать значения x и y в одну строку через пробел.

8. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)

Напишите функцию f2, которая имеет 2 параметра: первый (p1) – ссылка на функцию с 1 аргументом, которая возвращает число, второй (p2) – число, которое по умолчанию равно 1. Функция f2 выводит на экран значение p1 в точке p2.

9. Создайте класс Товар, в котором имеются атрибуты:

- name – название товара (закрытый);
- count – количество;
- price – цена (закрытый).

Создайте конструктор класса.

Создайте метод discount, который уменьшает цену на количество процентов, заданных в качестве параметра метода.

Создайте экземпляр класса Товар (name – Тетради, count – 100, price – 85).

Уменьшите количество товара на 25 (используем атрибут), а цену – на 10% (вызовите метод).

Промежуточный контроль

Вопросы для подготовки к зачету

1. Язык программирования Python и его место среди других языков программирования. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
2. Работа с числами в Python. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
3. Базовые числовые типы int и float. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
4. Оператор присваивания. Комбинированные операторы присваивания. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
5. Операторы для работы с числовыми объектами. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)

6. Встроенные функции и модули для работы с числами. (УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5))
7. Использование модулей. Инструкции `import` и `from`. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
8. Встроенные функции: `abs`, `round`. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
9. Операции над строками: конкатенация, повторение, доступ по индексу. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
10. Операции над строками: получение подстроки, проверка вхождения. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
11. Параметры функции `print`: `sep`, `end`. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
12. Преобразование строки в число и числа в строку. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
13. Операции сравнения. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
14. Логические операторы. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
15. Инструкция `if...else`. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
16. Инструкция цикла `while`. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
17. Инструкция цикла `for`. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
18. Функции и методы для работы со строками. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
19. Строки. Операции над строками. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
20. Функция `len`. Методы строк: `strip`, `find`, `count`, `replace`. (УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5))
21. Списки в Python. Создание: создание пустого списка. (УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5))
22. Генераторы списков. Создание копии списка. (УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5))
23. Основные операции над списками. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
24. Многомерные списки. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
25. Базовые алгоритмы: вычисление количества элементов, удовлетворяющих условию. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
26. Базовые алгоритмы: нахождение суммы, произведения, максимума, минимума. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
27. Добавление и удаление элементов списка. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
28. Сортировка списков по возрастанию и убыванию. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
29. Преобразование списка в строку. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
30. Кортежи. Создание. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
31. Операции над кортежами. Методы кортежей. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)

32. Словари. Создание словаря. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
33. Функции dict. zip. Генераторы словарей. (УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5))
34. Операции над словарями. Перебор элементов словаря. (УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5))
35. Множества. Создание. Генераторы множеств. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
36. Проверка наличия элемента во множестве. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
37. Встроенные функции и модули для работы с числами. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)

Вопросы для подготовки к Экзамену

1. Классы в языке Python. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
2. Атрибуты класса и экземпляра класса. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
3. Закрытые атрибуты. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
4. Глобальные и локальные переменные. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
5. Работа с текстовыми файлами. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
6. Методы класса. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
7. Перебор строк файла в цикле for. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
8. Запись в текстовый файл. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
9. Множественное наследование. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
10. Базовые принципы объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
11. Понятие класса и объекта. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
12. Определение класса и создание экземпляра класса. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
13. Методы класса. Параметр self. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
14. Статические методы. Закрытые методы. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
15. Атрибуты класса и экземпляра класса. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
16. Доступ к атрибуту. Закрытые атрибуты. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
17. Свойства. Создание и использование свойства. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)

18. Наследование. Базовый и производный классы. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
19. Событие. Обработчик события. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
20. Цикл обработки событий. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
21. Менеджеры размещения pack и grid. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
22. Использование элементов Button (Кнопка). УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
23. Использование элементов Label (Надпись). УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
24. Использование элементов Entry (Поле ввода). УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
25. Использование элементов Checkbutton (Флажок). УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
26. Использование элементов Radiobutton (Переключатель). УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
27. Создание и конфигурирование виджета. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)
28. Использование метода bind для обработки событий. УК-4 (1, 2, 3, 4), УК-10 (1, 2, 3, 4, 5)