

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--



**УТВЕРЖДЕНО**  
решением Ученого совета  
факультета культуры и искусства  
от «20» мая 2024 г., протокол №12/272

Председатель  /Н.С. Сафронов/  
Зав.кафедрой дизайна и  
интерьера факультета  
культуры и искусства  
Е.Л.Силантьева (по доверенности  
№ 218/08 от 29.01.2024г.)

## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Дисциплина	<b>Основы программирования на Python</b>
Факультет	Факультет математики, информационных и авиационных технологий
Кафедра	Кафедра информационных технологий
Курс	2 - очно-заочная форма обучения

Направление (специальность): 53.03.02 Музыкально-инструментальное искусство

Направленность (профиль/специализация): Оркестровые духовые и ударные инструменты

Форма обучения: очно-заочная

Дата введения в учебный процесс УлГУ: 01.09.2024 г.

Программа актуализирована на заседании кафедры: протокол № \_\_\_\_\_ от \_\_\_\_\_ 20\_\_ г.

Программа актуализирована на заседании кафедры: протокол № \_\_\_\_\_ от \_\_\_\_\_ 20\_\_ г.

Программа актуализирована на заседании кафедры: протокол № \_\_\_\_\_ от \_\_\_\_\_ 20\_\_ г.

Сведения о разработчиках:

ФИО	КАФЕДРА	Должность, ученая степень, звание
Санкин Николай Юрьевич	Кафедра информационных технологий	Доцент, Кандидат технических наук

## 1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

### Цели освоения дисциплины:

сформировать знания и умения, включающие базовые навыки программирования;

сформировать у студентов современные представления о возможностях и преимуществах информатизации в профессиональной деятельности.

### Задачи освоения дисциплины:

научиться проектировать и разрабатывать приложения, используя базовые возможности языка программирования Python.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Основы программирования на Python» относится к числу дисциплин блока Б1.О, предназначенного для студентов, обучающихся по направлению: 53.03.02 Музыкально-инструментальное искусство.

В процессе изучения дисциплины формируются компетенции: ЦК-1, ЦК-2.

Основные положения дисциплины используются в дальнейшем при изучении таких дисциплин как: Технологии и продукты цифровой экономики, Подготовка к сдаче и сдача государственного экзамена.

## 3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Код и наименование реализуемой компетенции	Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций
ЦК-1 Способен использовать инновационные продукты и технологии, анализировать данные и применять методы искусственного интеллекта	<p><b>знать:</b> методы применения сквозных цифровых технологий, методы и технологии сбора, структурирования, анализа данных для построения новых организационных и управленческих моделей, продуктов и сервисов</p> <p><b>уметь:</b> в случае выбора между перспективными инновационными и устаревшими подходами легко выбирать новые идеи и методы и предпринимать конкретные действия для генерации и реализации инновационных идей и подходов, уметь анализировать, синтезировать и оценивать информацию для принятия решений и реализации своих действий</p> <p><b>владеть:</b> навыками изменения решений при наличии новых аргументов или произошедших изменений, владеть технологиями управления полным жизненным циклом</p>

Код и наименование реализуемой компетенции	Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций
	данных
ЦК-2 Способен разрабатывать программы на языке Python для использования в сфере своей профессиональной деятельности	<p><b>знать:</b> основные понятия языка программирования Python, методы описания структур данных и классы задач, формулируемых и решаемых на Python</p> <p><b>уметь:</b> разрабатывать программы на языке Python, применять изученные методы и структуры данных в соответствии с технологией разработки программ</p> <p><b>владеть:</b> навыками разработки, отладки и тестирования программ на языке Python для использования в сфере своей профессиональной деятельности</p>

#### 4. ОБЩАЯ ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ

4.1. Объем дисциплины в зачетных единицах (всего): 2 ЗЕТ

4.2. Объем дисциплины по видам учебной работы (в часах): 72 часа

Форма обучения: очно-заочная

Вид учебной работы	Количество часов (форма обучения <u>очно-заочная</u> )	
	Всего по плану	В т.ч. по семестрам
1	2	3
Контактная работа обучающихся с преподавателем в соответствии с УП	12	12
Аудиторные занятия:	12	12
Лекции	4	4
Семинары и практические занятия	-	-
Лабораторные работы, практикумы	8	8
Самостоятельная работа	60	60
Форма текущего контроля знаний и контроля самостоятельной работы: тестирование, контр. работа, коллоквиум, реферат и др. (не менее 2 видов)	Тестирование	Тестирование
Курсовая работа	-	-
Виды промежуточной аттестации (экзамен, зачет)	Зачёт	Зачёт
Всего часов по дисциплине	72	72

### 4.3. Содержание дисциплины. Распределение часов по темам и видам учебной работы

Форма обучения: очно-заочная

Название разделов и тем	Всего	Виды учебных занятий					Форма текущего контроля знаний
		Аудиторные занятия			Занятия в интерактивной форме	Самостоятельная работа	
		Лекции	Практические занятия, семинары	Лабораторные работы, практикумы			
1	2	3	4	5	6	7	8
<b>Раздел 1. Среда разработки Python</b>							
Тема 1.1. Язык программирования Python. Области применения	3.5	0.5	0	0	0	3	Тестирование
Тема 1.2. Основные типы данных	3.5	0.5	0	0	0	3	Тестирование
Тема 1.3. Встроенные функции и методы для работы с числами	3.5	0.5	0	0	0	3	Тестирование
Тема 1.4. Строковый тип данных	5	1	0	1	0	3	Тестирование
<b>Раздел 2. Алгоритм. Типы алгоритмов</b>							
Тема 2.1. Понятие алгоритма. Типы алгоритмов	5.5	0.5	0	1	0	4	Тестирование
Тема 2.2. Циклический алгоритм	5.5	0.5	0	1	0	4	Тестирование
<b>Раздел 3. Отладка программы. Обработка исключений в Python</b>							
Тема 3.1. Отладка программы	5	0	0	1	0	4	Тестирование

Название разделов и тем	Всего	Виды учебных занятий					Форма текущего контроля знаний
		Аудиторные занятия			Занятия в интерактивной форме	Самостоятельная работа	
		Лекции	Практические занятия, семинары	Лабораторные работы, практикумы			
1	2	3	4	5	6	7	8
Тема 3.2. Разработка алгоритма с учетом ошибок выполнения	5	0	0	1	0	4	Тестирование
<b>Раздел 4. Функции. Модули. Создание собственных функций</b>							
Тема 4.1. Понятие функции	4.5	0.5	0	0	0	4	Тестирование
Тема 4.2. Создание собственных функций	4	0	0	0	0	4	Тестирование
<b>Раздел 5. Строки. Списки. Словари. Кортежи</b>							
Тема 5.1. Строки. Списки	5	0	0	1	0	4	Тестирование
Тема 5.2. Словари. Кортежи	5	0	0	1	0	4	Тестирование
<b>Раздел 6. Файловый ввод-вывод</b>							
Тема 6.1. Работа с файлами	5	0	0	1	0	4	Тестирование
<b>Раздел 7. Основы объектно-ориентированного программирования</b>							
Тема 7.1. Основы объектно-ориентированного подхода	4	0	0	0	0	4	Тестирование
<b>Раздел 8. Применение Python в искусственном интеллекте</b>							
Тема 8.1. Введение в машинное обучение с Python	4	0	0	0	0	4	Тестирование

Название разделов и тем	Всего	Виды учебных занятий					Форма текущего контроля знаний
		Аудиторные занятия			Занятия в интерактивной форме	Самостоятельная работа	
		Лекции	Практические занятия, семинары	Лабораторные работы, практикумы			
1	2	3	4	5	6	7	8
Тема 8.2. Обработка естественного языка (NLP) с Python	4	0	0	0	0	4	Тестирование
<b>Итого подлежит изучению</b>	72	4	0	8	0	60	

## 5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### Раздел 1. Среда разработки Python

#### Тема 1.1. Язык программирования Python. Области применения

Язык программирования Python. Области применения. Достоинства и недостатки. Области применения. Интерактивный режим. Официальный сайт языка Python. Работа в IDLE Python. Запуск/остановка скриптов на языке Python. Стиль написания кода: отступы, базовый синтаксис. Имена переменных, длина строки. Рекомендации по созданию имен переменных. Ключевые (служебные) слова. Первая программа. Сообщения интерпретатора.

#### Тема 1.2. Основные типы данных

Основные типы данных. Особенность оперирования данными в языке Python. Переменные, соотношение имени переменной со значением в памяти компьютера. Числовые типы данных. Преобразования числовых типов. Основные операции с числовыми данными.

#### Тема 1.3. Встроенные функции и методы для работы с числами

Встроенные функции языка Python для работы с числовыми данными. Ввод/вывод числовых данных. Ограничение точности вычислений при работе с вещественными (float) типом данных.

#### Тема 1.4. Строковый тип данных

Строки. Склеивание строк. Логический тип данных. Принцип высказываний. Базовые операции И, ИЛИ, НЕ. Примеры использования логических операций. Взаимодействие с числовыми и строковыми типами данных. Битовые операции.

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

## **Раздел 2. Алгоритм. Типы алгоритмов**

### **Тема 2.1. Понятие алгоритма. Типы алгоритмов**

Описание задачи в виде алгоритма. Линейный алгоритм. Блок-схемы. Операторы для работы с последовательностью. Приоритет выполнения операторов. Программирование разветвляющихся процессов. Алгоритмы с ветвлением. Условный оператор. Синтаксис условного оператора. Вложенные условия. Множественные условия. Сложные условия. Формулировка условий.

### **Тема 2.2. Циклический алгоритм**

Цикл for, синтаксис. Функция range(). Команды break и continue. Вложенные циклы for. Решение задач с помощью цикла for. Цикл while Циклы с предусловием и постусловием. Синтаксис. Счетчик итераций. Команды break и continue. Вложенные циклы. Бесконечный цикл. Формулировка условия. Решение задач с использованием циклов while.

## **Раздел 3. Отладка программы. Обработка исключений в Python**

### **Тема 3.1. Отладка программы**

Отладка программ. Типы ошибок: синтаксические, ошибки выполнения, семантические. Использование исключений для обработки ошибок выполнения без аварийного завершения программы.

### **Тема 3.2. Разработка алгоритма с учетом ошибок выполнения**

Сообщения интерпретатора об ошибках. Поиск документации по ошибкам. Отладка программы с семантическими ошибками: Экспериментальная отладка, создание контрольных меток.

## **Раздел 4. Функции. Модули. Создание собственных функций**

### **Тема 4.1. Понятие функции**

Назначение функции. Встроенные функции. Вызов справки по функциям. Встроенные модули, назначение. Вызов справки по модулям. Математические функции. Подключение модулей. Модуль Math. Использование функций модуля Math для решения математических задач.

### **Тема 4.2. Создание собственных функций**

Использование собственных функций. Параметры и аргументы функций. Локальные параметры и переменные. Возвращаемое значение. Область видимости переменных. Использование функций в программах. Создание собственных модулей. Модули NumPy, matplotlib, tkinter.

## **Раздел 5. Строки. Списки. Словари. Кортежи**

### **Тема 5.1. Строки. Списки**

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

Строки. Функция len. Перебор элементов строки. Срезы строк. Поиск. Подсчет. Методы строк. Сравнение строк. Работа с текстовыми файлами. Списки. Перебор элементов списка. Операции со списками. Срезы списков. Методы списков. Отображение, фильтрация и сокращение.

## **Тема 5.2. Словари. Кортежи**

Словари как счетчики. Циклы и словари. Обратный поиск. Словари и списки. Глобальные переменные. Кортежи и операции присваивания. Кортежи в качестве возвращаемого значения. Кортежи с переменным числом аргументов. Списки и кортежи. Словари и кортежи. Сравнение кортежей. Последовательности последовательностей.

## **Раздел 6. Файловый ввод-вывод**

### **Тема 6.1. Работа с файлами**

Открытие файла. Чтение файла. Текстовые и бинарные файлы. Запись в файл. Базы данных. Сериализация. Десериализация. Конвейеры

## **Раздел 7. Основы объектно-ориентированного программирования**

### **Тема 7.1. Основы объектно-ориентированного подхода**

Объекты в реальном мире, объекты в Python. Объект, атрибуты, методы. Создание объектов. Примеры класса. Полиморфизм и наследование. Экземпляры класса в качестве возвращаемого значения. Использование классов в программах.

## **Раздел 8. Применение Python в искусственном интеллекте**

### **Тема 8.1. Введение в машинное обучение с Python**

Типы задач машинного обучения: классификация, регрессия, кластеризация. Основные алгоритмы машинного обучения: линейная регрессия, логистическая регрессия, метод k-ближайших соседей, деревья решений, метод опорных векторов. Библиотеки Python для машинного обучения: Scikit-learn, TensorFlow, PyTorch.

### **Тема 8.2. Обработка естественного языка (NLP) с Python**

Введение в NLP: Основные задачи NLP: обработка текста, перевод, извлечение информации, чат-боты. Предварительная обработка текста: Токенизация, лемматизация, стеминг, удаление стоп-слов. Векторное представление слов: Word2Vec, GloVe, FastText.

## **6. ТЕМЫ ПРАКТИЧЕСКИХ И СЕМИНАРСКИХ ЗАНЯТИЙ**

### **7. ЛАБОРАТОРНЫЕ РАБОТЫ, ПРАКТИКУМЫ**



Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

Линейные алгоритмы. Особенности подключения и использования модулей.

Цели: получить навыки вводить и выводить данные, создавать переменные и выполнять арифметические операции.

Содержание: Написать программу для расчета по заданным формулам. Предварительно подготовить тестовые примеры в таблице excel.

Результаты: Написать программу для расчета по формуле. Подготовить не менее трёх тестовых примеров.

Ссылка: <http://lib.ulsu.ru/MegaPro/Download/MObject/15009>

Разветвляющиеся процессы

Цели: получить навык работы с условными операторами на языке Python.

Содержание: Написать программу для расчета по заданным формулам. Предварительно подготовить тестовые примеры в таблице excel.

Результаты: Написать программу для расчета по формуле. Подготовить не менее трёх тестовых примеров.

Ссылка: <http://lib.ulsu.ru/MegaPro/Download/MObject/15009>

Организация циклов

Цели: навык использования операторов цикла на языке Python

Содержание: Написать программу для расчета по заданным формулам. Предварительно подготовить тестовые примеры в таблице excel.

Результаты: Написать программу для расчета по формуле. Подготовить не менее трёх тестовых примеров.

Ссылка: <http://lib.ulsu.ru/MegaPro/Download/MObject/15009>

Работа со строками

Цели: изучить возможности языка Python для работы со строками

Содержание: Написать программу для расчета по заданным формулам. Предварительно подготовить тестовые примеры в таблице excel.

Результаты: Написать программу для расчета по формуле. Подготовить не менее трёх тестовых примеров.

Ссылка: <http://lib.ulsu.ru/MegaPro/Download/MObject/15009>

Одномерные массивы

Цели: Написать программу для расчета по заданным формулам. Предварительно подготовить тестовые примеры в таблице excel.

Содержание: Написать программу для расчета по заданным формулам. Предварительно подготовить тестовые примеры в таблице excel.

Результаты: Написать программу для расчета по формуле. Подготовить не менее трёх тестовых примеров.

Ссылка: <http://lib.ulsu.ru/MegaPro/Download/MObject/15009>

Двумерные массивы и функции

Цели: дать студентам практический навык в написании программ обработки двумерных массивов с использованием функций.

Содержание: Написать программу для расчета по заданным формулам. Предварительно подготовить тестовые примеры в таблице excel.

Результаты: Написать программу для расчета по формуле. Подготовить не менее трёх тестовых примеров.

Ссылка: <http://lib.ulsu.ru/MegaPro/Download/MObject/15009>

Файлы

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

Цели: дать студентам практический навык в написании программ, в которых выполняются операции с текстовыми файлами – чтение, запись

Содержание: Написать программу для расчета по заданным формулам. Предварительно подготовить тестовые примеры в таблице excel.

Результаты: Написать программу для расчета по формуле. Подготовить не менее трёх тестовых примеров.

Ссылка: <http://lib.ulsu.ru/MegaPro/Download/MObject/15009>

## 8. ТЕМАТИКА КУРСОВЫХ, КОНТРОЛЬНЫХ РАБОТ, РЕФЕРАТОВ

Данный вид работы не предусмотрен УП.

## 9. ПЕРЕЧЕНЬ ВОПРОСОВ К ЗАЧЕТУ

1. Какие правила форматирования кода в Python используются для обеспечения его читаемости (отступы, длина строки, комментарии)?

2. Как документировать свой код, чтобы он был понятен не только вам, но и другим разработчикам?

3. Какие ключевые слова в Python вы используете в своей функции?

4. Объяснить, как работает механизм присвоения значения переменной в Python. Что происходит в памяти компьютера, когда мы присваиваем значение переменной а числу 10? Как Python различает имена переменных и фактические значения в памяти?

5. В чем разница между целыми числами (int), числами с плавающей точкой (float) и комплексными числами (complex) в Python? Приведите примеры кода, демонстрирующие преобразование одного числового типа в другой.

6. Написать программу на Python, которая запрашивает у пользователя два целых числа, а затем вычисляет их сумму, разность, произведение и частное. Оформить код с помощью комментариев, объясняющих, что делает каждая строка программы.

7. Перечислить 5 встроенных функций Python, которые полезны для работы с числовыми данными. Для каждой функции приведите пример ее использования. В чем преимущества использования встроенных функций по сравнению с написанием собственного кода для выполнения этих операций?

8. Написать программу на Python, которая запрашивает у пользователя два вещественных числа (float) с помощью функции input(), затем вычисляет их сумму и выводит результат на экран с помощью функции print(). Убедитесь, что введенные пользователем значения преобразуются в вещественный тип данных перед вычислениями. Как проверить, что результат вычислений отображается с достаточной точностью?

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

9. Объясните, в чем заключается проблема точности вычислений с вещественными числами (float) в Python? Привести пример, который иллюстрирует эту проблему. Как можно минимизировать её влияние на результаты вычислений.

10. Объяснить, как работают операции склеивания строк в Python с использованием оператора + и метода join(). Привести пример, где для склеивания используется список строк, и результат получается разный при применении + и join().

11. Написать программу, которая запрашивает у пользователя его имя и возраст, а затем выводит на экран сообщение "Ваше имя: [имя], ваш возраст: [возраст]". В программе используйте логическое выражение, которое проверяет, является ли возраст пользователя большим 18. Если да, добавьте в сообщение фразу "Вы совершеннолетний".

12. Описать принцип работы битовых операций AND, OR, XOR, NOT в Python. Приведите пример кода, где битовые операции применяются для проверки наличия конкретных битов в двоичном представлении числа. Каковы преимущества использования битовых операций в программировании?

13. Привести примеры разных типов алгоритмов (линейный, ветвящийся, циклический). Как можно описать задачу в виде алгоритма?

14. Написать программу на Python, которая принимает на вход два числа и выводит на экран сообщение, какое из них больше. Используйте условный оператор if-else и добавьте в программу обработку случая, когда числа равны. Создайте блок-схему, иллюстрирующую логику работы этой программы.

15. Вложенные условия в Python. Приведите пример, где используются вложенные условия для проверки нескольких условий одновременно. Как можно использовать множественные условия и сложные условия в программах с ветвлением?

16. Написать программу на Python, которая выводит на экран все чётные числа в диапазоне от 1 до 20. Используйте цикл for и функцию range() для генерации чисел в диапазоне. Объясните, как работают команды break и continue в цикле for.

17. Написать программу на Python, которая спрашивает у пользователя число и выводит на экран все делители этого числа. Используйте цикл while для проверки всех чисел до введенного пользователем. Объясните, как вы сформировали условие цикла while. Что произойдёт, если пользователь введет нулевое число? Как можно изменить код, чтобы избежать ошибки в этом случае?

18. Объяснить, чем отличаются циклы for и while. В каких случаях лучше использовать цикл for, а в каких цикл while? Приведите примеры задач, которые можно решить как с помощью цикла for, так и с помощью цикла while.

19. Объяснить разницу между синтаксическими, ошибками выполнения и семантическими ошибками в программах на Python. Примеры каждого типа ошибок. Как можно использовать

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

инструменты отладки Python (например, `print` или `pdb`) для поиска и исправления ошибок?

20. Написать программу на Python, которая спрашивает у пользователя два числа и вычисляет их частное. Добавить обработку исключения `ZeroDivisionError`, чтобы программа не завершилась с ошибкой, если пользователь введет нулевое число в качестве делителя. Выведите сообщение об ошибке и попросите пользователя ввести другое число. Как можно использовать блок `try-except` для обработки ошибок?

21. Какие преимущества дает использование исключений для обработки ошибок выполнения по сравнению с простым проверяющим кодом. Приведите пример, где использование исключений делает код более читаемым и удобным для поддержания.

22. Как интерпретатор Python выдает сообщения об ошибках. Какие типичные элементы содержатся в сообщениях об ошибках? Как можно использовать эти элементы для поиска информации о причине ошибки в документации Python?

23. Написать программу на Python, которая вычисляет факториал числа. Внесите в код семантическую ошибку (например, неправильно используйте индекс в списке или ошибочно задайте условие в цикле). Опишите шаги, которые вы будете предпринимать для отладки программы с помощью методов экспериментальной отладки и создания контрольных меток.

24. Какие дополнительные инструменты (кроме `print` и `pdb`) можно использовать для отладки программ на Python? В чем их преимущества и недостатки? Как выбрать подходящий инструмент отладки для конкретной задачи?

25. В чем преимущества использования функций в программах? Привести пример встроенной функции Python. Показать, как можно получить справку по ней с помощью функции `help()`.

26. Модуль в Python. Какова роль модулей в организации кода? Привести пример встроенного модуля Python.

27. Написать программу на Python, которая спрашивает у пользователя радиус круга и вычисляет площадь и периметр этого круга. Используйте функции из модуля `math` для вычисления числа  $\pi$  и возведения в степень. Объясните, как вы подключаете модуль `math` к своей программе. Какие ещё математические функции может предоставлять модуль `math`?

28. Написать функцию на Python, которая принимает два числа в качестве аргументов и возвращает их сумму. Как использовать эту функцию в другой программе для вычисления суммы двух чисел, введенных пользователем? Объясните, как работают параметры и аргументы функции, а также понятие "возвращаемое значение".

29. Создайте собственный модуль на Python, который содержит функцию для вычисления факториала числа. Импортируйте этот модуль в другую программу и используйте функцию для вычисления факториала числа, введенного пользователем. Объясните, как работает импорт модулей в Python. Каковы преимущества создания собственных модулей?

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

30. Опишите, как можно использовать модули NumPy, matplotlib и tkinter в программах на Python. Приведите примеры применения каждого модуля. Какие задачи можно решать с помощью этих модулей?

31. Написать программу, которая запрашивает у пользователя текст и выводит на экран количество слов в этом тексте. Используйте функцию len() для определения длины строки и метод split() для разбиения строки на слова. Объясните, как работает метод split().

32. Создайте список из 10 случайных чисел. Напишите программу, которая сортирует этот список по возрастанию и выводит отсортированный список на экран. Используйте метод sort() для сортировки списка. Создайте новый список из первых 5 элементов отсортированного списка, используя срез списка. Объясните, как работать со срезом списка.

33. Написать программу, которая считывает текстовый файл и выводит на экран количество строк в этом файле. Используйте цикл for для перебора строк в файле. Измените программу так, чтобы она выводила на экран только те строки, которые начинаются с буквы "А". Используйте условный оператор if для проверки первого символа строки. Как можно использовать метод readlines() для считывания всех строк из файла одновременно?

34. Написать программу, которая считывает текст и выводит на экран количество вхождений каждого слова в этом тексте. Используйте словарь в качестве счетчика для подсчета вхождений слов. Объясните, как использовать цикл for для перебора слов в тексте и обновления значений в словаре. Как можно вывести результаты подсчета в формате "слово: количество"?

35. Создайте список из 5 имен. Напишите программу, которая запрашивает у пользователя имя и выводит на экран индекс этого имени в списке (если имя есть в списке). Используйте цикл for для перебора элементов списка и условный оператор if для сравнения имен. Как можно использовать словарь для более эффективного поиска имени в списке?

36. Напишите функцию, которая принимает в качестве аргумента список и возвращает кортеж, содержащий два элемента: первый элемент - это среднее значение чисел в списке, а второй элемент - это количество элементов в списке. Объясните, как работают кортежи в Python. Какие особенности имеет работа с кортежами по сравнению со списками?

37. Написать программу, которая запрашивает у пользователя имя файла и строку текста, затем записывает эту строку в указанный файл. Если файл уже существует, программа должна добавить строку в конец файла. Объясните, какие режимы открытия файла используются в Python для записи данных в файл?

38. Создать текстовый файл с несколькими строками текста. Напишите программу, которая считывает этот файл строка за строкой и выводит на экран только те строки, которые начинаются с буквы "А". Используйте цикл for для перебора строк в файле и условный оператор if для проверки первого символа строки. Как можно использовать метод readlines() для считывания всех строк из файла одновременно?

39. Написать программу, которая запрашивает у пользователя имя файла и выводит на экран

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

количество слов в этом файле. Используйте функцию `len()` для определения длины строки и метод `split()` для разбиения строки на слова. Как можно обработать случай, когда файл не существует?

40. Объяснить основные принципы объектно-ориентированного программирования (ООП): инкапсуляция, наследование и полиморфизм. Привести примеры реализации этих принципов в коде на Python, используя классы и объекты.

41. Создать класс Dog в Python, который будет представлять собаку. Класс должен иметь атрибуты `name` (имя) и `breed` (порода) и методы `bark()` (лай) и `run()` (бежать). Создайте объект класса Dog и продемонстрируйте использование его атрибутов и методов.

42. Написать программу на Python, которая создает список из двух объектов класса Dog, затем перебирает элементы списка и вызывает метод `bark()` для каждой собаки. Использовать цикл `for` для перебора элементов списка объектов.

43. Объясните разницу между задачами классификации, регрессии и кластеризации в машинном обучении. Приведите примеры реальных задач, которые можно решить с помощью каждого из этих типов машинного обучения.

44. Опишите алгоритм k-ближайших соседей (kNN). Как он работает? Какие параметры нужно установить для алгоритма kNN? Приведите пример использования алгоритма kNN с помощью библиотеки Scikit-learn для классификации данных.

45. Что такое глубокое обучение? Как отличается глубокое обучение от традиционного машинного обучения? Какие библиотеки Python часто используются для глубокого обучения? Приведите пример использования TensorFlow или PyTorch для решения задачи классификации изображений.

46. Объяснить разницу между токенизацией, лемматизацией и стемингом в предварительной обработке текста. Почему важно проводить предварительную обработку текста перед применением моделей NLP?

47. Векторное представление слов. Какие методы используются для создания векторного представления слов? Какие преимущества дает векторное представление слов для моделей NLP?

48. Описать работу рекуррентных нейронных сетей (RNN) и их применение в NLP. Как RNN могут использоваться для обработки последовательных данных, таких как текст?

## 10. САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩИХСЯ

*Содержание, требования, условия и порядок организации самостоятельной работы обучающихся с учетом формы обучения определяются в соответствии с «Положением об организации самостоятельной работы обучающихся», утвержденным Ученым советом УлГУ (протокол №8/268 от 26.03.2019г.).*

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

По каждой форме обучения: очная/заочная/очно-заочная заполняется отдельная таблица

Форма обучения: очно-заочная

Название разделов и тем	Вид самостоятельной работы (проработка учебного материала, решение задач, реферат, доклад, контрольная работа, подготовка к сдаче зачета, экзамена и др).	Объем в часах	Форма контроля (проверка решения задач, реферата и др.)
<b>Раздел 1. Среда разработки Python</b>			
Тема 1.1. Язык программирования Python. Области применения	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	3	Тестирование
Тема 1.2. Основные типы данных	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	3	Тестирование
Тема 1.3. Встроенные функции и методы для работы с числами	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	3	Тестирование
Тема 1.4. Строковый тип данных	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	3	Тестирование
<b>Раздел 2. Алгоритм. Типы алгоритмов</b>			
Тема 2.1. Понятие алгоритма. Типы алгоритмов	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	4	Тестирование
Тема 2.2. Циклический алгоритм	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	4	Тестирование
<b>Раздел 3. Отладка программы. Обработка исключений в Python</b>			
Тема 3.1. Отладка программы	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	4	Тестирование
Тема 3.2. Разработка алгоритма	Проработка учебного материала с	4	Тестирование



<b>Название разделов и тем</b>	<b>Вид самостоятельной работы (проработка учебного материала, решение задач, реферат, доклад, контрольная работа, подготовка к сдаче зачета, экзамена и др).</b>	<b>Объем в часах</b>	<b>Форма контроля (проверка решения задач, реферата и др.)</b>
с учетом ошибок выполнения	использованием ресурсов учебно-методического и информационного обеспечения дисциплины.		
<b>Раздел 4. Функции. Модули. Создание собственных функций</b>			
Тема 4.1. Понятие функции	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	4	Тестирование
Тема 4.2. Создание собственных функций	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	4	Тестирование
<b>Раздел 5. Строки. Списки. Словари. Кортежи</b>			
Тема 5.1. Строки. Списки	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	4	Тестирование
Тема 5.2. Словари. Кортежи	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	4	Тестирование
<b>Раздел 6. Файловый ввод-вывод</b>			
Тема 6.1. Работа с файлами	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	4	Тестирование
<b>Раздел 7. Основы объектно-ориентированного программирования</b>			
Тема 7.1. Основы объектно-ориентированного подхода	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	4	Тестирование
<b>Раздел 8. Применение Python в искусственном интеллекте</b>			
Тема 8.1. Введение в машинное обучение с Python	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	4	Тестирование



Название разделов и тем	Вид самостоятельной работы (проработка учебного материала, решение задач, реферат, доклад, контрольная работа, подготовка к сдаче зачета, экзамена и др).	Объем в часах	Форма контроля (проверка решения задач, реферата и др.)
	дисциплины.		
Тема 8.2. Обработка естественного языка (NLP) с Python	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	4	Тестирование

## 11. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### а) Список рекомендуемой литературы основная

1. Златопольский Д.М. Основы программирования на языке Python : учебник / Д.М. Златопольский ; Златопольский Д.М. - Москва : ДМК-пресс, 2017. - 284 с. - URL: <https://www.studentlibrary.ru/book/ISBN9785970605523.html>. - Режим доступа: ЭБС "Консультант студента"; по подписке. - ISBN 978-5-97060-552-3. / .— ISBN 0\_253809

2. Чернышев Станислав Андреевич. Основы программирования на Python : Учебное пособие для вузов / С.А. Чернышев ; Чернышев С. А. - Москва : Юрайт, 2022. - 286 с. - (Высшее образование). - URL: <https://urait.ru/bcode/496893> . - Режим доступа: Электронно-библиотечная система Юрайт, для авториз. пользователей. - Электрон. дан. - ISBN 978-5-534-14350-8 : 909.00. / .— ISBN 0\_312177

### дополнительная

1. Федоров Дмитрий Юрьевич. Программирование на языке высокого уровня Python : Учебное пособие для вузов / Д.Ю. Федоров ; Федоров Д. Ю. - 3-е изд. ; пер. и доп. - Москва : Юрайт, 2022. - 210 с. - (Высшее образование). - URL: <https://urait.ru/bcode/492920> . - Режим доступа: Электронно-библиотечная система Юрайт, для авториз. пользователей. - Электрон. дан. - ISBN 978-5-534-14638-7 : 569.00. / .— ISBN 0\_313662

2. Гуриков Сергей Ростиславович. Основы алгоритмизации и программирования на Python : Учебное пособие / С.Р. Гуриков ; Московский технический университет связи и информатики. - 1. - Москва : ООО "Научно-издательский центр ИНФРА-М", 2022. - 343 с. - ВО - Бакалавриат. - <http://znanium.com/catalog/document?id=379975>. - <https://znanium.com/cover/1356/1356003.jpg>. - Режим доступа: ЭБС Znanium; по подписке. - ISBN 978-5-16-017142-5. - ISBN 978-5-16-102278-8. / .— ISBN 0\_462127

3. Тарланов А. Т. Основы языка программирования Python : учебно-методическое пособие / А. Т. Тарланов, Ш. Г. Магомедов ; Тарланов А. Т., Магомедов Ш. Г. - Москва : РТУ МИРЭА, 2019. - 107 с. - Библиогр.: доступна в карточке книги, на сайте ЭБС Лань. - Книга из коллекции РТУ МИРЭА -

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

Информатика. - <https://e.lanbook.com/book/171465>. -  
<https://e.lanbook.com/img/cover/book/171465.jpg>. - Режим доступа: ЭБС "Лань"; для авторизир.  
пользователей. / .— ISBN 0\_392929

### **учебно-методическая**

1. Перцева И. А. Методические указания для самостоятельной работы студентов по дисциплине «Основы программирования на языке Python» для студентов всех направлений и специальностей не ИТ профиля / И. А. Перцева, Ю. Г. Савинов, И. А. Санников. - 2022. - 69 с. - Неопубликованный ресурс. - URL: <http://lib.ulsu.ru/MegaPro/Download/MObject/13479>. - Режим доступа: ЭБС УлГУ. - Текст : электронный. / .— ISBN 0\_476097.

### **б) Программное обеспечение**

- Операционная система "Альт образование"
- Офисный пакет "Мой офис"
- Python IDLE
- Visual studio code- Visual Studio Pro

### **в) Профессиональные базы данных, информационно-справочные системы**

#### **1. Электронно-библиотечные системы:**

1.1. Цифровой образовательный ресурс IPRsmart : электронно-библиотечная система : сайт / ООО Компания «Ай Пи Ар Медиа». - Саратов, [2024]. – URL: <http://www.iprbookshop.ru>. – Режим доступа: для зарегистрир. пользователей. - Текст : электронный.

1.2. Образовательная платформа ЮРАЙТ : образовательный ресурс, электронная библиотека : сайт / ООО Электронное издательство ЮРАЙТ. – Москва, [2024]. - URL: <https://urait.ru>. – Режим доступа: для зарегистрир. пользователей. - Текст : электронный.

1.3. База данных «Электронная библиотека технического ВУЗа (ЭБС «Консультант студента») : электронно-библиотечная система : сайт / ООО Политехресурс. – Москва, [2024]. – URL: <https://www.studentlibrary.ru/cgi-bin/mb4x>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.4. Консультант врача. Электронная медицинская библиотека : база данных : сайт / ООО Высшая школа организации и управления здравоохранением-Комплексный медицинский консалтинг. – Москва, [2024]. – URL: <https://www.gosmedlib.ru>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.5. Большая медицинская библиотека : электронно-библиотечная система : сайт / ООО Букап. – Томск, [2024]. – URL: <https://www.books-up.ru/ru/library/> . – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.6. ЭБС Лань : электронно-библиотечная система : сайт / ООО ЭБС Лань. – Санкт-Петербург, [2024]. – URL: <https://e.lanbook.com>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

1.7. ЭБС **Znanium.com** : электронно-библиотечная система : сайт / ООО Знаниум. - Москва, [2024]. - URL: <http://znanium.com> . – Режим доступа : для зарегистрир. пользователей. - Текст : электронный.

2. **КонсультантПлюс** [Электронный ресурс]: справочная правовая система. /ООО «Консультант Плюс» - Электрон. дан. - Москва : КонсультантПлюс, [2024].

3. **eLIBRARY.RU**: научная электронная библиотека : сайт / ООО «Научная Электронная Библиотека». – Москва, [2024]. – URL: <http://elibrary.ru>. – Режим доступа : для авториз. пользователей. – Текст : электронный

4. **Федеральная государственная информационная система «Национальная электронная библиотека»** : электронная библиотека : сайт / ФГБУ РГБ. – Москва, [2024]. – URL: <https://нэб.рф>. – Режим доступа : для пользователей научной библиотеки. – Текст : электронный.

5. **Российское образование** : федеральный портал / учредитель ФГАУ «ФИЦТО». – URL: <http://www.edu.ru>. – Текст : электронный.

6. **Электронная библиотечная система УлГУ** : модуль «Электронная библиотека» АБИС Мега-ПРО / ООО «Дата Экспресс». – URL: <http://lib.ulsu.ru/MegaPro/Web>. – Режим доступа : для пользователей научной библиотеки. – Текст : электронный.

## 12. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ:

Аудитории для проведения лекций, семинарских занятий, для выполнения лабораторных работ и практикумов, для проведения текущего контроля и промежуточной аттестации, курсового проектирования, групповых и индивидуальных консультаций (*выбрать необходимое*)

Аудитории укомплектованы специализированной мебелью, учебной доской. Аудитории для проведения лекций оборудованы мультимедийным оборудованием для представления информации большой аудитории. Помещения для самостоятельной работы оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа к электронной информационно-образовательной среде, электронно-библиотечной системе. Перечень оборудования, используемого в учебном процессе:

- Мультимедийное оборудование: компьютер/ноутбук, экран, проектор/телевизор
- Компьютерная техника

## 13. СПЕЦИАЛЬНЫЕ УСЛОВИЯ ДЛЯ ОБУЧАЮЩИХСЯ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) могут предлагаться одни из следующих вариантов восприятия

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

информации с учетом их индивидуальных психофизических особенностей:

- для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); в печатной форме на языке Брайля; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации;

- для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации;

- для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.

В случае необходимости использования в учебном процессе частично/исключительно дистанционных образовательных технологий, организация работы ППС с обучающимися с ОВЗ и инвалидами предусматривается в электронной информационно-образовательной среде с учетом их индивидуальных психофизических особенностей.

Разработчик	Доцент Кандидат технических наук	Санкин Николай Юрьевич
	Должность, ученая степень, звание	ФИО