

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

**УТВЕРЖДЕНО**  
решением Ученого совета факультета математики,  
информационных и авиационных технологий  
от «01» мая 2024 г., протокол № 5/24

Председатель \_\_\_\_\_ / М.А. Волков  
«21» мая 2024 г.



## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Дисциплина	<b>Программирование на языке Python</b>
Факультет	Факультет математики, информационных и авиационных технологий
Кафедра	Кафедра математического моделирования технических систем
Курс	1 - очная форма обучения

Направление (специальность): 15.03.04 Автоматизация технологических процессов и производств

Направленность (профиль/специализация): Цифровой инжиниринг в медицинском приборостроении

Форма обучения: очная

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
Санников Игорь Алексеевич	Кафедра математического моделирования технических систем	Заведующий кафедрой, Кандидат физико-математических наук, Доцент

## 1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

### Цели освоения дисциплины:

формирование у студентов навыков, соответствующих видам профессиональной деятельности, необходимых для решения профессиональных задач.

### Задачи освоения дисциплины:

приобретение студентами необходимых знаний о базовых концепциях программирования на Python, областях его применимости, конструкциях языка Python и технологии разработки программ на Python.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Программирование на языке Python» относится к числу дисциплин блока Б1.О, предназначенного для студентов, обучающихся по направлению: 15.03.04 Автоматизация технологических процессов и производств.

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

Основные положения дисциплины используются в дальнейшем при изучении таких дисциплин как: Статистика для анализа данных, Выполнение, подготовка к процедуре защиты и защита выпускной квалификационной работы.

## 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> Знает методы применения сквозных цифровых технологий, методы и технологии сбора, структурирования, анализа данных для построения новых организационных и управленческих моделей, продуктов и сервисов. Знает основные сквозные технологии (новые производственные технологии; нейротехнологии и искусственный интеллект; технологии беспроводной связи; компоненты робототехники и сенсорики; квантовые технологии; системы распределенного реестра; технологии виртуальной и дополненной реальности).</p> <p><b>уметь:</b> Умеет находить креативные способы решения проблемы, анализировать их плюсы и минусы, риски, выбирать оптимальное решение. Умеет распознавать непродуктивные ментальные модели и стереотипы и отказываться от них.</p> <p><b>владеть:</b></p>

Код и наименование реализуемой компетенции	Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций
	Владеет навыками изменения решений при наличии новых аргументов или произошедших изменений, владеть технологиями управления полным жизненным циклом данных. Владеет технологиями принятия решений, основанных на данных (культура и этика принятия решений на основе данных; встраивание процесса принятия решений на основе данных в бизнес-процессы организации; системы автоматического принятия решений, включая системы искусственного интеллекта). Владеет методиками обеспечения безопасности данных.
ЦК-2 Способен разрабатывать программы на языке Python для использования в сфере своей профессиональной деятельности	<p><b>знать:</b> Знает основные понятия языка программирования Python, методы описания структур данных и классы задач, формулируемых и решаемых на Python.</p> <p><b>уметь:</b> Умеет разрабатывать программы на языке Python, применять изученные методы и структуры данных в соответствии с технологией разработки программ.</p> <p><b>владеть:</b> Владеет 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: очная

Вид учебной работы	Количество часов (форма обучения <u>очная</u> )	
	Всего по плану	В т.ч. по семестрам
		1
1	2	3
Контактная работа обучающихся с преподавателем в соответствии с УП	36	36
Аудиторные занятия:	36	36
Лекции	-	-
Семинары и практические занятия	-	-
Лабораторные работы, практикумы	36	36
Самостоятельная работа	36	36
Форма текущего контроля знаний и контроля самостоятельной работы: тестирование, контр. работа, коллоквиум, реферат и др. (не менее 2 видов)	Оценивание выполнения задания, Тестирование	Оценивание выполнения задания, Тестирование

Вид учебной работы	Количество часов (форма обучения <u>очная</u> )	
	Всего по плану	В т.ч. по семестрам
		1
1	2	3
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: очная

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. Основы программирования на Python.</b>							
Тема 1.1. Язык Python. Типы данных. Особенности и ввода/вывода.	8	0	0	4	0	4	Тестирование
Тема 1.2. Линейные алгоритмы. Особенности и подключения и использования модулей.	8	0	0	4	0	4	Тестирование
Тема 1.3. Разветвляющиеся процессы. Условный оператор.	12	0	0	6	0	6	Оценивание выполнения задания, 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 1.4. Циклические алгоритмы. Операторы цикла..	8	0	0	4	0	4	Оценивание выполнения задания, Тестирование
Тема 1.5. Работа со строками.	8	0	0	4	0	4	Оценивание выполнения задания, Тестирование
Тема 1.6. Кортежи, списки словари, множества. Одномерные массивы.	12	0	0	6	0	6	Оценивание выполнения задания, Тестирование
Тема 1.7. Двумерные массивы.	8	0	0	4	0	4	Оценивание выполнения задания, Тестирование
Тема 1.8. Работа с файлами.	8	0	0	4	0	4	Тестирование
<b>Итого подлежит изучению</b>	72	0	0	36	0	36	

## 5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### Раздел 1. Основы программирования на Python.

#### Тема 1.1. Язык Python. Типы данных. Особенности ввода/вывода.

Python как объектно-ориентированный язык. Типы данных. Особенности. Ввод-вывод величин разных типов. Преобразование типов. Форматный вывод.

#### Тема 1.2. Линейные алгоритмы. Особенности подключения и использования модулей.

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

Модули в языке Python. Особенности подключения и использования. Знакомство с модулем math. Составление линейных алгоритмов.

### **Тема 1.3. Разветвляющиеся процессы. Условный оператор.**

Разветвляющиеся процессы. Программная реализация средствами языка Python.

### **Тема 1.4. Циклические алгоритмы. Операторы цикла..**

Циклические алгоритмы. Реализация циклов с условием, с повторением. Создание и использование пользовательских функций.

### **Тема 1.5. Работа со строками.**

Строковые величины. Особенности реализации и работы с ними. Основные методы работы со строками.

### **Тема 1.6. Кортежи, списки словари, множества. Одномерные массивы.**

Коллекции языка Python. Кортежи, списки, словари, множества. Особенности и реализация. Особенности работы с одномерными массивами средствами языка Python.

### **Тема 1.7. Двумерные массивы.**

Двумерные массивы. Особенности и реализация обработки.

### **Тема 1.8. Работа с файлами.**

Текстовые файлы в языке Python. Примеры работы с файлами.

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

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

Линейные программы

Цели: Напишите программу для расчета по двум формулам. Подготовьте не менее пяти тестовых примеров. Предварительно выполните вычисления с использованием калькулятора или офисного приложения, например Excel или Calc. Результаты вычисления по обеим формулам должны совпадать. Отсутствующие в языке функции выразите через имеющиеся.

Содержание: Для математических вычислений в Python имеются как встроенные, так и дополнительные функции и методы. Применить дополнительные математические функций можно после подключения модуля math: `import math` либо `from math import *` В первом случае функции рассматриваются как методы объекта `math` и должны записываться так: `import math`

`print(math.sin(math.pi/4)) print(math.sqrt(2)/2)` Во втором случае вызов функции может быть сделан в более привычной для нас форме: `from math import * print(sin(pi/4)) print(sqrt(2)/2)` Вместе с тем, такой способ импорта может нарушить пространство имен программы, поскольку может возникнуть конфликт между именами переменных, которые использует программист и именами импортируемых функций. При импорте можно ограничиться только необходимыми функциями, например: `from math import (pi, sin, cos,tan, log)` Ввод данных Ввод данных можно выполнить с клавиатуры функцией `input(): m = input([str])` При этом на экран будет выведена строка `str`, а переменная `m` получит значение строкового типа, введённое пользователем. Строковый тип может быть преобразован, например, к типу `int` или `float`, если введённое значение – число. Вывод данных Вывод данных на экран монитора может быть выполнен функцией `print()`. Эта функция позволяет выполнять форматированный вывод, как с использованием Си-подобного форматирования, так и с использованием форматной строки Python. Следующие строки демонстрируют, как можно форматировать вывод. `for x in range(1,11): print('%2d %3d %7.2f' % (x, x*x, x*x*x))` или `for x in range(1,11): print("{0:.2f} {1:.2f} {2:.4f}".format(x, x**2, x**3))` Буква в формате числа определяет тип выводимого числа. Так, `d` – это целый тип, `f` – вещественное число. Число в формате означает то число позиций, которое будет использовано для вывода числа. Для вещественного числа указывается, после точки, количество выводимых десятичных знаков. Во второй строке использован Си-подобный формат, в котором формат числа начинается с процента `"%"`. В этом формате аргументы отделяются от форматной части строки так же символом `%` – процент. В третьей строке используется форматная строка Python, в которой в форматной строке позиции для значений аргументов выделяются фигурными скобками. Обратите внимание на то, что сами форматные строки начинаются и завершаются одиночной или двойной кавычкой. В Python допускаются оба вида кавычек для выделения строки. Важно только что бы начало и конец были одинаковыми. Так же следует понимать, что в промежутках между символами форматирования могут находиться и другие символы или слова: `for x in range(1,11): print('x=%2d x^2=%3d x^3=%7.2f%' % (x, x*x, x*x*x))` или `for x in range(1,11): print("a={0:.2f} x={1:.2f} y={2:.4f}".format(x, x**2, x**3))` Использование форматных строк делает вывод данных более внятным. Решение задания Вернемся к нашим примерам и запишем их, используя правила языка Python. Первое выражение примет вид: `1. y = tan(x**2/2-1)**2+2*cos(x-pi/6)/(1/2+sin(a)**2)` Второе выражение представим в виде двух: `2. tmp=log(3-cos(pi/4+2*x),3+sin(x))/(1+tan(2*x/pi)**2) y = pow(2, tmp)` Описание алгоритма Для вычислений необходимо обеспечить ввод двух переменных `x` и `a`. Поскольку по условиям задачи их тип и точность представления не заданы, выберем для них вещественный тип (`float`). Для оптимизации записи выражения используем промежуточную переменную `tmp`. 1. Ввести значения `a` и `x`, преобразовать к типу `float`. 2. Вычислить выражение 1. 3. Вывести результат вычисления. 4. Вычислить значение переменной `tmp`; 5. Вычислить выражение 2. 6. Вывести результат вычисления. Листинг программы `from math import * a = float(input('Введите параметр a: ')) x = float(input('Введите значение x: ')) y=tan(x**2/2-1)**2+(2*cos(x-pi/6))/(1/2+sin(a)**2) print("{0:.2f} {1:.2f} {2:.4f}".format(a, x, y)) tmp=log(3-cos(pi/4+2*x),3+sin(x))/(1+tan(2*x/pi)**2) y=pow(2,tmp) print("{0:.2f} {1:.4f}".format(x, y))`

Результаты: Вывод в консоль результатов высисления

Ссылка: <http://lib.ulsu.ru/MegaPro/Download/MObject/13200>

Разветвляющиеся вычислительные процессы

Цели: Написать программу, которая по введённому значению аргумента вычисляет значение функции, заданной в виде графика.

Содержание: Для решения задачи использован оператор ветвления, который в языке Python имеет следующий вид: `if <Логическое выражение>: <Блок – выполняется, если условие истинно> [elif`

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

<Логическое выражение>: <Блок – выполняется, если условие истинно> ] [else: <Блок – выполняется, если все условия ложны> ] <Блок> – это набор вложенных инструкций, которые выделяются одинаковым количеством пробелов (обычно четырьмя). Для ввода данных используется инструкция input(), которая возвращает строку. Введённые значения, перед использованием в арифметических выражениях, должны быть преобразованы к числовому формату. Вывод данных выполняется инструкцией print(), в которой использован форматированный вывод данных. График функции представлен фрагментами прямых линий, описываемых уравнением  $y=kx+b$  и дугами кругов. В общем случае уравнение круга может быть представлено так:  $(x-a)^2+(y-b)^2=R^2$ . Известные параметры, угол наклона и смещение прямой, а так же координаты центра дуг, определим, используя данные из графика. Для прямой на интервале  $(-5, 0)$  можем записать следующую систему уравнений: Из второго уравнения следует, что  $b = -2$ , а из первого –  $k = -3/5$ . Для полукруга с центром  $(6, 2)$  уравнение круга примет вид:  $(x-6)^2+(y-2)^2=2^2$ . Перепишем уравнение так:  $(x-6)^2=4-(y-2)^2$ . Отсюда следует: Знак перед корнем выбран для случая, когда рассматривается верхняя часть полукруга. Выполнив необходимые вычисления для всех фрагментов функции, мы получим систему уравнений, которую запишем в следующем виде: Функция определена на всём диапазоне. При этом, особых точек у неё нет. Описание алгоритма 1. Ввести значение аргумента  $x$  и преобразовать его к типу float. 2. Определить, к какому интервалу из области определения функции оно принадлежит, и вычислить значение функции  $y$  по соответствующей формуле. 3. Вывести значение  $x$  и  $y$ . Описание входных и выходных данных Входные данные поступают с клавиатуры, а выходные - выводятся на монитор для просмотра. Входные и выходные данные имеют тип float. Листинг программы (вариант 1) 

```
from math import * # теперь можно так: # print(sin(pi/4)) x = float(input('Введите значение x=')) if x < -5: y = 1 if x >=-5 and x<0: y = -(3/5)*x-2 if x >= 0 and x<2: y = -sqrt(4-x**2) if x >= 2 and x<4: y = x-2 if x >= 4 and x<8: y = 2+sqrt(4-(x-6)**2) if x >= 8: y = 2 print("X={0:.2f} Y={1:.2f}".format(x, y))
```

 Следует отметить, что в такой записи алгоритма проверка выполняется для всех условных операторов, в том числе и тех, которые следуют за вычисленным. Так, например, если  $x$  равно  $-3$ , то выполнится второй оператор, но и во всех последующих операторах операция сравнения будет проведена. Число проверок можно сократить, если написать программу с использованием вложенных условных операторов. Листинг программы (вариант 2) 

```
from math import * # теперь можно так: # print sin(pi/4) x = float(input('Введите значение x=')) if x < -5: y = 1 elif x >=-5 and x<0: y = -(3/5)*x-2 elif x >= 0 and x<2: y = -sqrt(4-x**2) elif x >= 2 and x<4: y = x-2 elif x >= 4 and x<8: y = 2+sqrt(4-(x-6)**2) else: y = 2 print("X={0:.2f} Y={1:.2f}".format(x, y))
```

 Результаты: Введите значение аргумента:  $-6$   $X= -6.00$   $Y= 1$  Введите значение аргумента:  $-3.33$   $X= -3.33$   $Y= -0.00$  Введите значение аргумента:  $6$   $X= 6.00$   $Y= 4.00$

Ссылка: <http://lib.ulsu.ru/MegaPro/Download/MObject/13200>

Организация циклов

Цели: Вычислить и вывести на экран в виде таблицы значения функции, заданной графически (см. лабораторная работа № 2, задание 1), на интервале от  $X_{нач}$  до  $X_{кон}$  с шагом  $dx$ . Интервал и шаг задать таким образом, чтобы проверить все ветви программы. Таблицу снабдить заголовком и шапкой.

Содержание: Для решения задачи использована программа, подготовленная в лабораторной работе №2, задание 1 и оператор цикла с предусловием: <Начальное значение> while <Условие>: <Инструкции> <Приращение> [else: <Блок, выполняемый, если не использовался break> ] Для обмена с консолью (вывод сообщений и ввод начальных данных) использованы стандартные процедуры print() и input(). Результаты работы программы записываются в текстовый файл. Описание алгоритма 1. Ввести значения переменных  $X_{beg}$ ,  $X_{end}$ ,  $dx$ . 2. Присвоить текущему значению  $X_t$  начальное значение:  $X_t = X_{нач}$ . 3. Вычислить значение функции и вывести в виде



строки таблицы. 4. Вычислить новое значение аргумента  $X_t = X_t + D_x$ . 5. Если значение аргумента меньше  $X_{end}$ , то перейти к пункту 3. 6. Завершить рисование таблицы и работу программы. Описание входных и выходных данных В предшествующей работе был принят вещественный тип данных (real). В этой работе тип данных сохранён. Для упрощения последующего контроля работы программы в выходной текстовый файл записываются и начальные данные. Листинг программы

```
from math import *
print('Введите Xbeg, Xend и Dx')
xb = float(input('Xbeg='))
xe = float(input('Xend='))
dx = float(input('Dx='))
print("Xbeg={0: 7.2f} Xend={1: 7.2f}".format(xb, xe))
print(" Dx={0: 7.2f}".format(dx))
xt = xb
print("+-----+-----+")
print("I X I Y I")
print("+-----+-----+")
while xt <= xe:
    if xt < -5: y = 1
    elif xt >=-5 and xt<0: y = -(3/5)*xt-2
    elif xt >= 0 and xt<2: y = -sqrt(4-xt**2)
    elif xt >= 2 and xt<4: y = xt-2
    elif xt >= 4 and xt<8: y = 2+sqrt(4-(xt-6)**2)
    else: y = 2
    print("I{0: 7.2f} I{1: 7.2f} I".format(xt, y))
    xt += dx
print("+-----+-----+")
```

Результаты: Xbeg= -10.00 Xend= 10.00 Dx= 1.00 +-----+-----+ I X I Y I +-----+-----+ I -10.00 I 1.00 I I -9.00 I 1.00 I I -8.00 I 1.00 I I -7.00 I 1.00 I I -6.00 I 1.00 I I -5.00 I 1.00 I I -4.00 I 0.40 I I -3.00 I -0.20 I I -2.00 I -0.80 I I -1.00 I -1.40 I I 0.00 I -2.00 I I 1.00 I -1.73 I I 2.00 I 0.00 I I 3.00 I 1.00 I I 4.00 I 2.00 I I 5.00 I 3.73 I I 6.00 I 4.00 I I 7.00 I 3.73 I I 8.00 I 2.00 I I 9.00 I 2.00 I I 10.00 I 2.00 I +-----+-----+

Ссылка: <http://lib.ulsu.ru/MegaPro/Download/MObject/13200>

## 8. ТЕМАТИКА КУРСОВЫХ, КОНТРОЛЬНЫХ РАБОТ, РЕФЕРАТОВ

Данный вид работы не предусмотрен УП.

## 9. ПЕРЕЧЕНЬ ВОПРОСОВ К ЗАЧЕТУ

1. Переменные и базовые типы данных языка Python. Арифметические операции. Оператор присваивания.
2. Ввод/вывод. Особенности реализации. Форматный вывод.
3. Особенности подключения и использования модулей в Python.
4. Операции отношения и логические операции. Условный оператор. Особенности использования
5. Циклы с условием и заданным числом повторений. Реализация циклов в Python.
6. Функции в языке Python. Создание пользовательских функций и модулей.
7. Строки и символы в языке Python, способы задания и вывода строки. Основные функции для работы со строками.
8. Коллекции языка Python. Списки. Особенности работы со списками.
9. Коллекции языка Python. Множества. Примеры работы с множествами.

10. Коллекции языка Python. Словари. Примеры.
11. Коллекции языка Python. Кортежи. Примеры использования.
12. Массивы. Способы задания и обработки массивов в Python.
13. Текстовые файлы. Основные функции.

## 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.).*

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

Форма обучения: очная

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. Основы программирования на Python.</b>			
Тема 1.1. Язык Python. Типы данных. Особенности ввода/вывода.	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	4	Тестирование
Тема 1.2. Линейные алгоритмы. Особенности подключения и использования модулей.	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	4	Тестирование
Тема 1.3. Разветвляющиеся процессы. Условный оператор.	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	6	Оценивание выполнения задания, Тестирование
Тема 1.4. Циклические алгоритмы. Операторы цикла..	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	4	Оценивание выполнения задания, Тестирование
Тема 1.5. Работа со строками.	Проработка учебного материала с	4	Оценивание выполнения задания,

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.)
	использованием ресурсов учебно-методического и информационного обеспечения дисциплины.		Тестирование
Тема 1.6. Кортежи, списки словари, множества. Одномерные массивы.	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	6	Оценивание выполнения задания, Тестирование
Тема 1.7. Двумерные массивы.	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	4	Оценивание выполнения задания, Тестирование
Тема 1.8. Работа с файлами.	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. Федоров Д. Ю. Программирование на python : учебное пособие / Д. Ю. Федоров. - 6-е изд. ; пер. и доп. - Москва : Юрайт, 2025. - 187 с. - (Высшее образование). - URL: <https://urait.ru/bcode/556864> . - Режим доступа: Электронно-библиотечная система Юрайт, для авториз. пользователей. - ISBN 978-5-534-19666-5. / .— ISBN 0\_545942

### дополнительная

1. Лучано Рамальо. Python. К вершинам мастерства : практическое руководство / Р. Лучано ; Лучано Рамальо. - Москва : ДМК-пресс, 2016. - 768 с. - URL: <https://www.studentlibrary.ru/book/ISBN9785970603840.html>. - Режим доступа: ЭБС "Консультант студента"; по подписке. - ISBN 978-5-97060-384-0. / .— ISBN 0\_253816

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

2. Маккинли, Уэс. Python и анализ данных / У. Маккинли, А. Слинкина ; Уэс Маккинли; перевод А. Слинкина. - Саратов : Профобразование, 2019. - 482 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - Текст. - Лицензия до 28.10.2024. - электронный. - Электрон. дан. (1 файл). - URL: <http://www.iprbookshop.ru/88752.html>. - Режим доступа: ЭБС IPR BOOKS; для авторизир. пользователей. - ISBN 978-5-4488-0046-7. / .— ISBN 0\_150368

### **учебно-методическая**

1. Рацеев Сергей Михайлович. Задачи по программированию и основные алгоритмы : учеб.-метод. пособие по курсу "Информатика". Ч. 2 / С.М. Рацеев. - Ульяновск : УлГУ, 2009. - Загл. с экрана. - Имеется печ. аналог. - Электрон. текстовые дан. (1 файл : 364 Кб). - URL: <http://lib.ulsu.ru/MegaPro/Download/MObject/776>. - Режим доступа: ЭБС УлГУ. - Текст : электронный. / .— ISBN 0\_1369.

2. Угаров В. В. Элементы программирования на языке Python : учебно-методическое пособие / В. В. Угаров, Ю. В. Цыганова ; УлГУ, ФМИАТ. - Ульяновск : УлГУ, 2019. - Загл. с экрана. - Электрон. текстовые дан. (1 файл : 2,02 МБ). - Режим доступа: ЭБС УлГУ. - Текст : электронный. / .— ISBN 0\_36128.

3. Перцева И. А. Методические указания для самостоятельной работы студентов по дисциплине «Основы программирования на Python» для студентов всех направлений и специальностей не ИТ профиля / И. А. Перцева, Ю. Г. Савинов, И. А. Санников ; УлГУ, ФМИиАТ. - 2023. - 69 с. - Неопубликованный ресурс. - URL: <http://lib.ulsu.ru/MegaPro/Download/MObject/15009>. - Режим доступа: ЭБС УлГУ. - Текст : электронный. / .— ISBN 0\_511237.

4. Перцева И. А. Методические указания для самостоятельной работы студентов по дисциплине «Программирование на языке Python» для студентов всех направлений и специальностей ФМИАТ / И. А. Перцева, Ю. Г. Савинов, И. А. Санников. - 2022. - 87 с. - Неопубликованный ресурс. - URL: <http://lib.ulsu.ru/MegaPro/Download/MObject/13200>. - Режим доступа: ЭБС УлГУ. - Текст : электронный. / .— ISBN 0\_468729.

### **б) Программное обеспечение**

- Операционная система "Альт образование"
- Офисный пакет "Мой офис"

### **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.

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

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.rosmedlib.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.

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

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

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 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	Кандидат физико-математических наук, Доцент	Санников Игорь Алексеевич
	Должность, ученая степень, звание	ФИО