

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизация производственных процессов»

Язык программирования Python как основа обработки больших данных

Методические указания  
с заданиями по контрольной работе  
для студентов заочной формы обучения

Ростов-на-Дону  
ДГТУ  
2022

УДК 681.5

Составитель: Быкадор В.С.

Методические указания. – Ростов-на-Дону : Донской гос. техн. ун-т, 2022. – 42 с.

Методические указания с заданиями по контрольной работе по дисциплине «Язык программирования Python как основа обработки больших данных. Задания для контрольной работы» предназначены для студентов заочной формы обучения по направлению подготовки 15.04.04 «Автоматизация технологических процессов и производств» профиль «Интеллектуальные системы сбора и анализа больших данных»

УДК 681.5

Печатается по решению редакционно-издательского совета  
Донского государственного технического университета

---

В печать \_\_\_\_\_.\_\_\_\_.20\_\_ г.  
Формат 60x84/16. Объем \_\_\_\_\_ усл. п. л.  
Тираж \_\_\_\_\_ экз. Заказ № \_\_\_\_\_.

---

Издательский центр ДГТУ  
Адрес университета и полиграфического предприятия:  
344000, г. Ростов-на-Дону, пл. Гагарина, 1

© Донской государственный  
технический университет, 2022

## Содержание

<b>ОБЩИЕ СВЕДЕНИЯ И ВЫБОР ВАРИАНТА .....</b>	<b>4</b>
<b>ЗАДАНИЕ № 1. Работа с основными стандартными типами и операциями языка программирования Python .....</b>	<b>5</b>
<b>ЗАДАНИЕ № 2. Отработка основных элементов инструкций языка программирования Python.....</b>	<b>8</b>
<b>ЗАДАНИЕ № 3. Применение парадигмы структурного программирования языка программирования Python.....</b>	<b>10</b>
<b>ЗАДАНИЕ № 4. Разработка элементарного класса на языке программирования Python.....</b>	<b>12</b>
<b>ЗАДАНИЕ № 5. Расширение функциональности пользовательского класса через механизм наследования .....</b>	<b>19</b>
<b>ЗАДАНИЕ № 6. Разработка системы классов, реализующих механизм полиморфизма, перегрузку операторов и функции обратного вызова в языке программирования Python .....</b>	<b>21</b>
<b>ЗАДАНИЕ № 7. Проработка форматирования строк и вывода текстовой информации в консоль на языке программирования Python .....</b>	<b>23</b>
<b>ЗАДАНИЕ № 8. Работа со средствами стандартной библиотеки языка программирования Python для обработки дат, времени и файлов.....</b>	<b>25</b>
<b>ЗАДАНИЕ № 9. Работа со средствами стандартной библиотеки языка программирования Python для взаимодействия с реляционной базой данных sqlite и XML-файлами .....</b>	<b>30</b>
<b>ЗАДАНИЕ № 10. Работа с основными классами и функциями математической библиотеки numpy языка программирования Python, используемых для обработки данных .....</b>	<b>35</b>
<b>ЗАДАНИЕ № 11. Работа с основными классами и функциями математической библиотеки scipy языка программирования Python, используемых для обработки данных .....</b>	<b>38</b>
<b>ЗАДАНИЕ № 12. Работа с основными классами и функциями библиотеки визуализации данных matplotlib языка программирования Python.....</b>	<b>40</b>
<b>Учебно-методическое и информационное обеспечение дисциплины .....</b>	<b>42</b>

## ОБЩИЕ СВЕДЕНИЯ И ВЫБОР ВАРИАНТА

Контрольная работа состоит из ряда заданий, выполнение которых необходимо для освоения дисциплины. Одни из заданий выполняются вне зависимости от номера варианта, а другие задания выполняются по номеру зачётной книжки. Задания, которые для выполнения требуют номера зачетной книжки отмечены следующей надписью

**ВНИМАНИЕ:**  
**Вариант задания выбирается по последней цифре номера зачётной книжки.**

## ЗАДАНИЕ № 1. Работа с основными стандартными типами и операциями языка программирования Python

Цель — изучить основные встроенные типы объектов и элементарные операции языка программирования Python.

Подключаемые библиотеки: отсутствуют.

### Описание задания

1) Выполните элементарные математические операции над следующими числами и заполните таблицу:

Операнды		Математические операции					
		+	—	*	/	**	//
45	12						
23.12	3.22						
3 + 4j	5 + 10j						

2) Выполните операции над строкой и заполните таблицы:

Строка S	Операции					
	len(S)	S[2]	S[-1]	S[1:5]	S[2:]	S + ' what'
'Example'						

Строка S	Операции		
	S*3	S.find('am')	S.replace('am', 'QWE')
'Example'			

3) Выполните операции над списком и заполните таблицы:

Список L	Операции		
	len(L)	L[2]	L[0][2]
['have', 678, 12.45]			

Список L	Операции		
	L + ['q', 1]	L.append(3)	L.pop(1)
['have', 678, 12.45]			

Список L	Операции		
	L.sort()	L.reverse()	L[100]
['c', 'a', 'f']			

Список L	Операции		
	L[2]	L[2][2]	
[[1, 5], ['aqw', 7], [2.67, 2+3j 'abc']]			

4) Выполните операции над словарём

**D = {'name': 'Пётр', 'age': 20, 'group': 'МЗИСБ11'}**

и заполните таблицу:

Метод	Результат выполнения
D.keys()	
D.values()	

<b>D['group']</b>	
<b>D['stature ']= 190</b>	
<b>del(D['age'])</b> <b>D['age']</b>	

5) Выполните операции над кортежем

**T = ('a', 'b', 'c', 'a')**

и заполните таблицу:

<b>Метод</b>	<b>Результат выполнения</b>
<b>len(T)</b>	
<b>T + (3, 12, 3)</b>	
<b>T.index('b')</b>	
<b>T.count('a')</b>	
<b>T.count(3)</b>	
<b>T[1] = 'group'</b>	

## ЗАДАНИЕ № 2. Отработка основных элементов инструкций языка программирования Python

Цель — изучить основные элементы инструкций языка программирования Python.

Подключаемые библиотеки: отсутствуют.

### Описание задания

- 1) Напишите программу, которая должна выполнить суммирование чётных чисел [2, 21, 43, 67, 24, 67, 68, 22, 32, 27, 89, 45, 46, 20, 88] используя оператор if-else и цикл for; оператор if-else и цикл while.
- 2) Напишите простейшую программу калькулятор, которая будет реализовывать четыре арифметических действия: +, -, \*, /

Порядок работы такой программы в консоли следующий:

*(цифры «2», «3» и символ «+» вводит пользователь)*

Новое вычисление:

a = 2

операция: +

b = 3

Результат: 2 + 3 = 5

Продолжить? [д/н]

Порядок работы Если пользователь ответ «д» или «Д», но начинается новый сеанс вычисления. Если пользователь ответит «н» или «Н», то программа завершает свою работу.

Пример консольных сообщений программы калькулятора показан на рисунке ниже:



Новое вычисление:

a = 2

операция: +

b = 3

Результат:  $2.0 + 3.0 = 5.0$

Продолжить? [д/н]д

Новое вычисление:

a = 3

операция: /

b = 2

Результат:  $3.0 / 2.0 = 1.5$

Продолжить? [д/н]н

Для реализации программы калькулятор полезны следующие функции:

input(...) - считывание данных с консоли.

float(...) - приведение строки к вещественному числу.

print(...) - вывод данных в консоль.

### ЗАДАНИЕ № 3. Применение парадигмы структурного программирования языка программирования Python

Цель — изучить основы реализации парадигмы структурного программирования на языке программирования Python.

Подключаемые библиотеки: `math`.

#### Описание задания

Усовершенствуйте программу калькулятора, добавив в программу вычисление суммы последовательности чисел, среднего арифметического и вычисление корней квадратного уравнения.

Программа должна предоставлять интерактивный интерфейс работы через консоль, который работает циклически и позволяет пользователю находится в постоянном сеансе работы с программой.

Организовать программу необходимо в виде функций:

Основная программа реализует интерактивный сеанс с пользователем позволяя ему выбирать то, что пользователь желает вычислить. В зависимости от того что пользователь выбрал вызывается программная функция (подпрограмма) для реализации соответствующих вычислений.

Функции в программе, то есть то, что должно быть реализованно как **def**:

- Обычные вычисления — теперь калькулятор, который реализовал обычные вычисления становится отдельной функцией.
- Вычисление суммы последовательности чисел.
- Вычисление среднего арифметического последовательности чисел.
- Вычисление корней квадратного уравнения.

Ограничение: из модуля `math` можно использовать только функцию `sqrt()`.

Примеры консольных сообщений программы калькулятора показаны на рисунках ниже:

Новое вычисление:

Обычный калькулятор: КАЛК или CALC  
Сумма значений: СУМ или SUM  
Среднее арифметическое: СРД или AVR  
Квадратное уравнение: КВУР или SQ  
Выйти из режима: введите любой другой символ

Ваш выбор: калк  
a = 2  
операция: \*  
b = 12  
Результат: 2.0 \* 12.0 = 24.0  
Продолжить? [д/н]

Новое вычисление:

Обычный калькулятор: КАЛК или CALC  
Сумма значений: СУМ или SUM  
Среднее арифметическое: СРД или AVR  
Квадратное уравнение: КВУР или SQ  
Выйти из режима: введите любой другой символ

Ваш выбор: сум  
Введите числа через пробел: 1 2 3.5 4.1  
Результат: СУМ = 10.6  
Продолжить? [д/н]

Продолжить? [д/н]

Новое вычисление:

Обычный калькулятор: КАЛК или CALC  
Сумма значений: СУМ или SUM  
Среднее арифметическое: СРД или AVR  
Квадратное уравнение: КВУР или SQ  
Выйти из режима: введите любой другой символ

Ваш выбор: срд  
Введите числа через пробел: 3 4 5 10.24  
Результат: СРД = 5.5600000000000005  
Продолжить? [д/н]

Продолжить? [д/н]

Новое вычисление:

Обычный калькулятор: КАЛК или CALC  
Сумма значений: СУМ или SUM  
Среднее арифметическое: СРД или AVR  
Квадратное уравнение: КВУР или SQ  
Выйти из режима: введите любой другой символ

Ваш выбор: кwur

ВЫЧИСЛЕНИЕ КОРНЕЙ КВАДРАТНОГО УРАВНЕНИЯ

$$A * (x^2) + B * x + C = 0$$

Введите коэффициенты уравнения

$$A = 3$$

$$B = 2$$

$$C = 1$$

Корни:

$$-0.3333333333333333 + j0.47140452079103173$$

$$-0.3333333333333333 - j0.47140452079103173$$

Продолжить? [д/н]

## ЗАДАНИЕ № 4. Разработка элементарного класса на языке программирования Python

Цель — изучить основы реализации парадигмы объектно-ориентированного программирования (реализация собственных классов) на языке программирования Python.

Подключаемые библиотеки: возможно потребуется в некоторых вариантах math.

### Описание задания

Реализовать пользовательский класс в соответствии с вариантом задания.

**ВНИМАНИЕ:**  
**Вариант задания выбирается по последней цифре номера зачётной книжки.**

**Многие из представленных заданий имеют реализацию в стандартных библиотеках языка программирования Python, НО ЭТО УЧЕБНЫЕ ЗАДАЧИ И ПОЭТОМУ ТРЕБУЕТСЯ ПРЕДЛОЖИТЬ СВОЮ РЕАЛИЗАЦИЮ КЛАССОВ!**

### Вариант № 0:

Создать класс Employee - сотрудник. Класс должен включать:

- поле типа int для хранения номера сотрудника;
- поле типа int для хранения номера отдела ;
- поле типа string для хранения имени сотрудника;
- и поле типа float для хранения величины его оклада.

Данные класса должны инициализироваться через конструктор и через свойства, подумайте для каких полей свойства нужны только чтение, а для каких — для чтения и записи.

Методы класса должны позволять:

- пользователю отображать данные класса;
- переводить сотрудника в другой отдел.

Написать программу, которая запросит пользователя ввести данные для трех сотрудников и выведет полученную информацию на экран. По переводить сотрудника в разные отделы.

### **Вариант № 1:**

Создать класс `Int`, имитирующий стандартный тип `integer`. По умолчанию экземпляр класса должен инициализироваться нулем. Единственное поле этого класса должно иметь тип `Int`.

Создать свойство, которое будет устанавливать значение поля в требуемое значение и позволять получать его значение.

Определить методы, которые позволяют:

- выполнять основные арифметические действия (+, -, \*, /) над двумя объектами типа `Int` и как результат возвращать объект типа `Int`;
- преобразовать типа `Int` в строку;
- заменять числа, из которых состоит значение типа `Int`, на соответствующий список названий чисел. Например, пусть тип `Int` содержит значение **536**, тогда метод должен выдавать список ['пять', 'три', 'шесть'].

Написать программу, позволяющую продемонстрировать работу свойств методов экземпляра класса `Int`.

### **Вариант № 2:**

Реализовать класс `Complex` типа комплексное число. Поля: вещественная составляющая, мнимая составляющая.

Реализовать свойства возвращающие:

- вещественную часть комплексного числа;
- мнимую часть комплексного числа;
- модуль комплексного числа и его угла;
- угол комплексного числа.

Реализовать методы позволяющие:

- осуществлять умножение комплексных чисел;
- осуществлять сложение/вычитание комплексных чисел;
- возвращение значение комплексного числа в виде строки в алгебраической форме;
- возвращение значение комплексного числа в виде строки в показательной форме.

Написать программу, позволяющую продемонстрировать работу свойств методов экземпляра класса `Complex`.

### **Вариант № 3:**

Создать класс типа `Date` с полями и свойствами (помните, что в свойствах должна быть реализована проверка корректности получаемых данных):

- день (1–31);
- месяц (1–12);
- год (целое число).

Класс должен имеет конструктор.

Методы:

- печать по шаблону «5 января 1997 года»;
- печать по шаблону «05.01.1997»;
- вычислять интервал между двумя датами и возвращать разницу в виде списка в формате [лет, месяцев, дней].

Написать программу, позволяющую продемонстрировать работу свойств методов экземпляра класса `Date`.

#### Вариант № 4:

Создать класс типа Rectangle. Инициализация значений полей как через конструктор, так и свойства.

Поля и свойства: высота и ширина (свойства должны проверять корректность введенных данных - значения полей не равны нулю и не отрицательные).

Ещё свойства вычисления: площади и периметра.

Методы:

- печати параметров прямоугольника;
- рисования прямоугольника в консоли, где одна единица длины равна символу '\*', например прямоугольник Ш x Д = 3 x 5, будет выглядеть так

```
*****
*      *
*****
```

Написать программу, позволяющую продемонстрировать работу свойств методов экземпляра класса Rectangle.

#### Вариант № 5:

Создать класс типа Circle. Инициализация значений полей как через конструктор, так и свойства.

Поля и свойства: радиус, координаты центра (свойства должны проверять корректность введенных данных - значения полей не равны нулю и не отрицательные).

Ещё свойства вычисления: площади, длины окружности.

Метод печати параметров окружности.

Написать программу, позволяющую продемонстрировать работу свойств методов экземпляра класса Circle.

## Вариант № 6:

Создать класс множество целых чисел `Set`.

Методы реализуют:

- добавление и удаление элемента из множества (заметьте, что во множестве находится только уникальные элементы!);
- объединение множеств;
- пересечение множеств;
- разность множеств.

Написать программу, позволяющую продемонстрировать работу свойств методов экземпляра класса `Set`.

## Вариант № 7:

Создать класс типа `Square`. Инициализация значений полей как через конструктор, так и свойства.

Поля и свойства:

- длина стороны (свойства должны проверять корректность введенных данных - значение поля не равно нулю и не отрицательное).
- координата левого верхнего угла  $X$  и  $Y$  (свойства должны проверять корректность введенных данных - значение поля не отрицательное).

Ещё свойства вычисления: площади и периметра.

Методы:

- печати параметров квадрата ;
- рисования квадрата в консоли, где одна единица длины равна символу `'*'`, например прямоугольник  $Ш \times Д = 3 \times 3$ , будет выглядеть так

```
***
*  *
***
```

- осуществлять сдвиг квадрата по координатам  $X$  и  $Y$ .

Написать программу, позволяющую продемонстрировать работу свойств методов экземпляра класса `Square`.



### **Вариант № 8:**

Создать класс типа Time с полями:

- час (0–23);
- минуты (0–59);
- секунды (0–59).

Свойства установки:

- времени (проверка корректности получаемых данных);
- получения часа;
- получения минут;
- получения секунд.

Инициализация экземпляра класса также должна осуществляться через конструктор.

Методы печати:

- печать по шаблону «16 часов 18 минут 3 секунды»;
- «4 p.m. 18 минут 3 секунды».

Написать программу, позволяющую продемонстрировать работу свойств методов экземпляра класса Time.

### **Вариант № 9:**

Создать класс Vector.

Методы:

- добавления нового элемента в вектор;
- обращение к отдельному элементу по заданному индексу;
- сложения и вычитания со скаляром;
- сложения и вычитания с другим объектом Vector такой же длины и возвращение нового объекта типа Vector.

Возвращаемое свойство – количество фактических элементов массива (длина массива).

Написать программу, позволяющую продемонстрировать работу свойств методов экземпляра класса `Vector`.

## **ЗАДАНИЕ № 5. Расширение функциональности пользовательского класса через механизм наследования**

Цель — изучить основы реализации парадигмы объектно-ориентированного программирования (наследование) на языке программирования Python.

Подключаемые библиотеки: отсутствуют.

### **Описание задания**

Реализовать набор пользовательских классов в соответствии с вариантом задания.

Требуется:

- Определить иерархию классов (в соответствии с вариантом – выделить базовый и производные классы).
- Реализовать классы (самостоятельно задать поля, свойства и методы класса). Написать демонстрационную программу, в которой создаются объекты различных классов и демонстрируется их работа.

**ВНИМАНИЕ:**  
**Вариант задания выбирается по последней цифре номера зачётной книжки.**

### **Вариант № 0:**

Классы – человек (имя, дата рождения), абитуриент (количество баллов), студент (курс, группа, факультет), преподаватель (должность, кафедра).

### **Вариант № 1:**

Классы – растение (название, вид), дерево (возраст), цветок (длина стебля), роза (цвет).

### **Вариант № 2:**

Классы – кадры (имя), рабочий (специальность, цех), инженер (квалификация, подразделение), администрация (должность).

### **Вариант № 3:**

Классы – БГТУ (адрес), факультет (название), группа (номер, староста, курс), подгруппа (номер, количество студентов).

### **Вариант № 4:**

Классы – печатное издание (издательство, год, название), журнал (номер, месяц), книга (тематика, автор, количество страниц), учебник (назначение).

### **Вариант № 5:**

Классы – млекопитающие (род), парнокопытные (среда обитания), птицы (хищники), животное (вид, род, вес).

### **Вариант № 6:**

Классы – место (площадь, название), область (количество населенных пунктов, руководство), город (область, количество жителей, мэр), деревня (район).

### **Вариант № 7:**

Классы – товар (название), радиотовары (назначение), пищевой продукт (отдел), молочный продукт (разновидность, дата изготовления), транзистор (тип, номер).

### **Вариант № 8:**

Классы – автомобиль (марка, номер), поезд (номер, количество вагонов, количество пассажиров в вагоне), транспортное средство (средняя скорость, вид топлива, год выпуска).

### **Вариант № 9:**

Классы – область (место расположения), населённый пункт (вид, количество жителей), королевство (король), государство (название, денежная единица, символика, руководитель).

## ЗАДАНИЕ № 6. Разработка системы классов, реализующих механизм полиморфизма, перегрузку операторов и функции обратного вызова в языке программирования Python

Цель — изучить основы реализации парадигмы объектно-ориентированного программирования (полиморфизм) на языке программирования Python.

Подключаемые библиотеки: отсутствуют.

**Классы, которые предлагается реализовать в задании, имеют реализацию в стандартных библиотеках языка программирования Python, НО ЭТО УЧЕБНЫЕ ЗАДАЧИ И ПОЭТОМУ ТРЕБУЕТСЯ ПРЕДЛОЖИТЬ СВОЮ РЕАЛИЗАЦИЮ КЛАССОВ!**

### Описание задания

Реализовать три класса:

- UserComplex - комплексное число;
- UserVector – вектор;
- UserMatrix – матрица.

Все эти пользовательские классы должны реализовывать следующие действия:

- инициализация начальными значениями через конструктор;
- суммирование двух экземпляров одного и того же класса, например, одно комплексное число складывается с другим комплексным числом и получается третье комплексное число, как результат операции суммирования;
- вычитание двух экземпляров одного и того же класса;
- суммирование с присваиванием;
- вычитание с присваиванием;
- равенство значений двух объектов одного класса;
- вывод на консоль значений объектов через стандартную функцию print().

То есть необходимо реализовать следующие операции: +, -, +=, -=; ==, print(объект\_пользовательского\_типа).

Рекомендация. В языке Python уже есть заготовленные методы, которые позволяют реализовать вышеуказанную функциональность пользовательских классов и добиться их полиморфного поведения. Это так называемые специальные методы, ещё их называют «магические» методы или dunder-методы.

Так, например, для определения операции суммирования есть метод `__add__(...)`.

## ЗАДАНИЕ № 7. Проработка форматирования строк и вывода текстовой информации в консоль на языке программирования Python

Цель — изучить основы форматированного вывода строк в консоль на языке программирования Python.

Подключаемые библиотеки: отсутствуют.

Используя строковое форматирование `.format()` можно добиться вывода строки в консоль в определенном виде.

Например, пусть есть строка «студент» и её необходимо вывести с начала строки вывода в консоли:

```
s = 'студент'
print(s + '|')
```

-----  
студент|

Теперь выведем строку «студент» так, чтобы после неё оставалось ещё 5-ть пустых символов («студент» = 7 символов + 5 пустых символов = 12 символов):

```
s = 'студент'
f = '{:12}'.format(s)
print(f + '|')
```

-----  
студент |

Далее заполним пустые символы - «\_»:

```
s = 'студент'
f = '{:_<12}'.format(s)
print(f + '|')
```

-----  
студент\_\_\_\_\_|

## Описание задания

- 1) '{??.format('студент') – разместить слово посередине 20-ти символьной строки.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
						с	т	у	д	е	н	т							

- 2) '{??.format(56)

		5	6
--	--	---	---

- 3) '{??.format(-23)

-			2	3
---	--	--	---	---

- 4) Пусть есть список lst = [1, 'a', 'hello', 'all', 4] необходимо вывести  
'{??.format(??)

a	1	1			4
---	---	---	--	--	---



## ЗАДАНИЕ № 8. Работа со средствами стандартной библиотеки языка программирования Python для обработки дат, времени и файлов

Цель — изучить основы преобразований информации о времени и датами, а также получить первоначальные навыки работы с файлами на языке программирования Python.

Подключаемые библиотеки: `datetime`, `calendar`, `os`.

### Описание задания

#### 1) Арифметика дат (`import datetime`).

- a. Создайте объект текущей даты, используя класс `datetime.now()`.
- b. Из созданного объекта текущей даты получите дату вчерашнего и завтрашнего дней используя объект `timedelta()`.
- c. Выведи значения полученных дат в консоль.
- d. Выполните вычитание из текущей даты вчерашней и завтрашней, какие получились результаты и что они означают?
- e. Сравните все даты между собой, выведите полученные результаты в консоль.
- f. Создайте дату 20.12.2022 15:20.
- g. Добавьте к созданной дате: 3 часа, 5 дней и 2 месяца.
- h. Выведите полученную дату в следующем формате: «12 апреля 2018 года 11:23».

#### 2) Работа с классом календаря (`import calendar`).

- a. Класс `calendar.TextCalendar()` позволяет создать календарь, а метод `formatyear(2023)` класса `TextCalendar()` позволяет вывести в консоль календарь в виде:

2023													
January							February						
Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su
						1			1	2	3	4	5
2	3	4	5	6	7	8	6	7	8	9	10	11	12
9	10	11	12	13	14	15	13	14	15	16	17	18	19
16	17	18	19	20	21	22	20	21	22	23	24	25	26
23	24	25	26	27	28	29	27	28					
30	31												

- b. Требуется перевести календарь на русский язык

2023													
Январь							Февраль						
Пн	Вт	Ср	Чт	Пт	Сб	Вс	Пн	Вт	Ср	Чт	Пт	Сб	Вс
						1			1	2	3	4	5
2	3	4	5	6	7	8	6	7	8	9	10	11	12
9	10	11	12	13	14	15	13	14	15	16	17	18	19
16	17	18	19	20	21	22	20	21	22	23	24	25	26
23	24	25	26	27	28	29	27	28					
30	31												

Если неделя начинается с воскресенья, то необходимо установить начало недели с понедельника.

- 3) Работа с методом `calendar.monthcalendar()` для создания собственного вывода в консоль календаря по месяцам (`import calendar` и `import os`).
- a. Выполните элементарный вывод данных сформированных методом `calendar.monthcalendar(2022, 3)` и изучите структуру данных.
- b. Выполните следующий код для того, чтобы изучить как выполняется раскраска текста при выводе в консоль:

```
print('\033[31m' + 'Привет' + '\033[32m' + ' Мир!' + '\033[0m' + ' Обычный текст')
```

```
Привет Мир! Обычный текст
```

- c. Необходимо разработать и реализовать пользовательский класс для `UserCalendar` для собственного вывода в консоль календаря по месяцу на заданный год, то есть вызов пользовательского класса должен выглядеть так:

```
from clsUserCalendar import UserCalendar
```

```
user_calendar = UserCalendar(2022, 3)
print(user_calendar)
```

- d. Вывод должен быть следующего вида для марта 2022 года:

Март 2022 года						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Обратите внимание, что:

- вывод должен быть на русском языке;
- выходные дни должны быть обозначены синим цветом;
- праздничные дни должны быть выделены красным цветом и записаны жирным цветом.

Справочные данные для выше приведённого задания.

В качестве праздничных дней примите следующие дни:

Январь – 01, 02, 03, 04, 05, 06, 07, 08, 09, 10;

Февраль – 23;

Март – 08;

Май – 01, 09;

Июнь – 12;

Ноябрь – 04.

Коды цветов для текста:

обычный текст и рабочие дни –	\033[0m
выходные дни (но не праздники) –	\033[34m
праздничные дни –	\033[1;31m

Ещё несколько скриншотов для лучшего понимания работы класса UserCalendar:

Январь 2022 года						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Май 2022 года						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Июль 2022 года						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

По желанию можете реализовать дополнительно раскраску следующего рабочего дня, в том случае, если праздничный день приходится на выходной день.

- 4) Работа с файлами. Необходимо разработать программу, которая будет считывать данные из файла data.txt, которые представлены в определенном формате. Далее необходимо считанные из файла данные преобразовать к виду таблицы и записать их в другой файл out.txt.

```

≡ data.txt
1 Петров;Математика:3;Физика:3;Информатика:3
2 Сидорова;Математика:4;ТАУ:3;Информатика:5
3 Иванов;Математика:5;Физика:5;СУБД:4;Информатика:5

```

программа

out.txt						
1	Фамилия	Математика	Информатика	ТАУ	Физика	СУБД
2						
3	Петров	3	3	–	3	–
4	Сидорова	4	5	3	–	–
5	Иванов	5	5	–	5	4
6						

При этом следует обратить внимание на то, что программа должна быть адаптивной, то есть при внесении изменений в файл data.txt (естественно в рамках принятого формата) программа должна выдавать (без её изменения) в выходном файле out.txt, также таблицу. Например, добавим в файл data.txt ещё одну строку, в которой будет новая дисциплина «Теор. мех.», программа без какой-либо доработки должна выполнить преобразование исходного файла в выходной файл:

≡ data.txt

```
1 Петров;Математика:3;Физика:3;Информатика:3
2 Сидорова;Математика:4;ТАУ:3;Информатика:5
3 Иванов;Математика:5;Физика:5;СУБД:4;Информатика:5
4 Анатолий;Математика:2;Физика:3;Информатика:4;Теор. мех.:5
```

≡ out.txt

1	Фамилия	ТАУ	СУБД	Математика	Информатика	Теор. мех.	Физика
2							
3	Петров	–	–	3	3	–	3
4	Сидорова	3	–	4	5	–	–
5	Иванов	–	4	5	5	–	5
6	Анатолий	–	–	2	4	5	3

## ЗАДАНИЕ № 9. Работа со средствами стандартной библиотеки языка программирования Python для взаимодействия с реляционной базой данных sqlite и XML-файлами

Цель — изучить основы получения информации из реляционных баз данных и xml-файлов на языке программирования Python.

Подключаемые библиотеки: sqlite3, xml.etree, os.

### Описание задания

1) Дана база данных ToolMachines.db. Необходимо реализовать программу, которая будет реализовывать интерактивный режим обращения к базе данных.

а. Получить список цехов и их идентификаторы используя следующий SQL-запрос:

```
SELECT * FROM dicShops
```

б. Выполните вывод полученных результатов в консоль в виде таблицы:

№ цеха	Название цеха
1	Механообрабатывающий цех
2	Механосборочный цех
3	Кузнечно-прессовый цех
4	Подготовительный цех
5	Опытно-экспериментальный цех

Выберите № цеха:

с. Реализовать интерактивный режим, в котором пользователь должен ввести номер цеха, который является его идентификатором:

№ цеха	Название цеха
1	Механообрабатывающий цех
2	Механосборочный цех
3	Кузнечно-прессовый цех
4	Подготовительный цех
5	Опытно-экспериментальный цех

Выберите № цеха:

d. Введенный пользователем номер цеха передать в следующий SQL-запрос:

```
SELECT
    dicGroupToolMachines.Name,
    dicModelToolMachine.Name,
    dicInvNumToolMachines.InvNumber
FROM
    dicShops INNER JOIN dicToolMachinesShops
        ON dicShops.Id = dicToolMachinesShops.IdShop
        INNER JOIN dicInvNumToolMachines
            ON dicToolMachinesShops.InvNumber =
dicInvNumToolMachines.InvNumber
            INNER JOIN dicGroupToolMachines
                ON dicInvNumToolMachines.IdGroup = dicGroupToolMachines.Id
                INNER JOIN dicModelToolMachine
                    ON dicInvNumToolMachines.IdModel = dicModelToolMachine.Id
WHERE
    dicShops.Id = сюда необходимо передать номер цеха, выбранный
пользователем
```

e. Выполните вывод полученных результатов в консоль в виде таблицы:

Группа	Модель	Инв. №
Фрезерный	BM127	961816
Токарный	1516Ф3	936405
Зубообрабатывающий	5M14	893737
Отрезной	ОН 401	845585
Сверлильный	2A106П	791280
Фрезерный	BM127	745243
Сверлильный	ГС2112	604579
Сверлильный	ГС2112	599851
Токарный	ДИП 300	590298
Зубообрабатывающий	5310	565334
Протяжной	7A523	526503
Зубообрабатывающий	5E32	524341
Фрезерный	BM127	438417
Сверлильный	2A106П	367330
Фрезерный	ГФ2171С5	361384
Сверлильный	ГС2112	355019
Протяжной	7A523	351434
Зубообрабатывающий	5B12	202210
Сверлильный	НС-12А	153248
Сверлильный	2Е470	112104
Сверлильный	2Н115ПМ	50050

Желаете продолжить? [Да/Нет]: ☐

- f. Не забудьте выполнить проверку ошибок, например, если пользователь введёт в место номера цеха буквенный символ:

№ цеха	Название цеха
1	Механообрабатывающий цех
2	Механосборочный цех
3	Кузнечно-прессовый цех
4	Подготовительный цех
5	Опытно-экспериментальный цех

Выберите № цеха: а  
 № цеха = а не является номером.  
 Желаете продолжить? [Да/Нет]: ☐

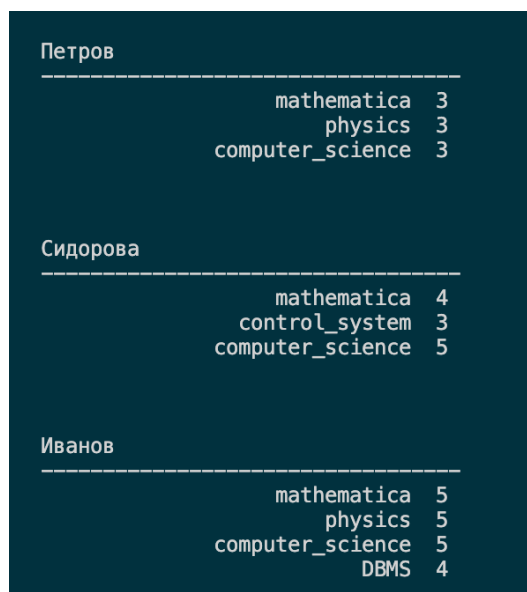


- 2) Дан xml-файл task\_xml.xml. Необходимо реализовать программу, которая будет выполнять элементарный вывод данных в консоль, считанных с xml-файла.

Исходный xml-файл:

```
<?xml version="1.0" encoding="UTF-8"?>
<group>
  <student name="Петров">
    <mathematica>3</mathematica>
    <physics>3</physics>
    <computer_science>3</computer_science>
  </student>
  <student name="Сидорова">
    <mathematica>4</mathematica>
    <control_system>3</control_system>
    <computer_science>5</computer_science>
  </student>
  <student name="Иванов">
    <mathematica>5</mathematica>
    <physics>5</physics>
    <computer_science>5</computer_science>
    <DBMS>4</DBMS>
  </student>
</group>
```

Вывод в консоль:



```
Петров
-----
      mathematica  3
        physics   3
    computer_science  3

Сидорова
-----
      mathematica  4
    control_system  3
    computer_science  5

Иванов
-----
      mathematica  5
        physics   5
    computer_science  5
                DBMS  4
```

При добавлении элемента в структуру xml-файла, например:

```
.....  
.....  
.....  
<student name="Анатолий">  
  <mathematica>2</mathematica>  
  <physics>3</physics>  
  <computer_science>4</computer_science>  
  <mechanica>4</mechanica>  
</student>
```

Программа должна правильно обработать добавленные элементы в xml-файл без внесения в программу каких-либо изменений:

Петров	
mathematica	3
physics	3
computer_science	3
Сидорова	
mathematica	4
control_system	3
computer_science	5
Иванов	
mathematica	5
physics	5
computer_science	5
DBMS	4
Анатолий	
mathematica	2
physics	3
computer_science	4
mechanica	4

## ЗАДАНИЕ № 10. Работа с основными классами и функциями математической библиотеки numpy языка программирования Python, используемых для обработки данных

Цель — изучить некоторые элементы математической библиотеки numpy, широко используемой для математических расчётов с использованием языка программирования Python.

Подключаемые библиотеки: numpy, matplotlib. (библиотеки numpy и matplotlib требуют дополнительной установки, так как данные библиотеки не входят в состав стандартных библиотек).

### Описание задания

1) Вычислите и постройте графики базовых функций:

- a. синусоидальная функция на отрезке  $[0...2\pi]$ ;
- b. тангенс на отрезке  $[0...2\pi]$ ;
- c.  $f(x) = \begin{cases} x + 1 & x < 5 \\ x^2 & x \geq 5 \end{cases}$ ;

2) Выполнить действия с матрицами:

- a. Задать две матрицы размером 3x3:

$$\mathbf{A} = \begin{pmatrix} 2 & 10 & 4 \\ -3 & 4 & 22 \\ 22 & -8 & 10 \end{pmatrix} \text{ и } \mathbf{B} = \begin{pmatrix} -2 & 5 & -6 \\ 4 & 10 & 2 \\ 6 & 3 & 3 \end{pmatrix}$$

- b.  $\mathbf{C} = \mathbf{A} + \mathbf{B}$
- c.  $\mathbf{C} = \mathbf{A} - \mathbf{B}$
- d.  $\mathbf{C} = \mathbf{A} * \mathbf{B}$
- e.  $\mathbf{A}^T$
- f.  $\mathbf{B}^{-1}$
- g.  $\text{determinant}(\mathbf{A})$

3) Дана функция:

$$y(x) = x^2$$

Необходимо найти определенный интеграл функции  $y(x)$  на отрезке  $[2; 8]$  используя функцию `trapz()` и проверить полученный результат аналитически, то есть по общему выражению:

$$F = \int_2^8 x^2 dx = \dots$$

5) Дана система линейных уравнений:

$$\begin{cases} x_1 + 2x_2 + 3x_3 + 1 = 0 \\ 2x_1 + 2x_2 - 2x_3 - 4 = 0 \\ 3x_1 + x_2 + x_3 - 1 = 0 \end{cases}$$

Требуется найти решение данной системы линейных уравнений, то есть отыскать чему равны  $x_i$ , используя функцию `solve()`. После нахождения равны  $x_i$  требуется выполнить проверку, чтобы убедиться в правильности решения.

6) Даны следующие экспериментальный данные:

$X_i$	0	1	2	3	4	5	6	7	8	9	10
$Y_i$	1	-8	-11	-2	25	76	157	274	433	640	901

- Постройте график зависимости  $Y_i(X_i)$  в виде точек.
- Используя метод наименьших квадратов, который реализован двумя функциями `polyfit()` и `polyval()`, найдите коэффициенты полиномов 2-ой и 3-ей степеней и постройте графики этих полиномов в виде линий на том же графике на котором построена зависимости  $Y_i(X_i)$  и сравните полученные результаты.

7) Пусть задана гармоническая функция:

$$y(t) = 5 + 2 \sin(2\pi \cdot f \cdot t) + 3 \cos(2\pi \cdot 3 \cdot f \cdot t) + 0.5 \sin(2\pi \cdot 10 \cdot f \cdot t)$$

- Необходимо построить спектр гармонического сигнала по следующему алгоритму

## Алгоритм построения спектра виброакустического сигнала

1. если в дискретизированном сигнале содержится постоянная составляющая, то она может замаскировать слабые компоненты спектра. В этом случае необходимо вычислить среднее значение исходного сигнала (сигнала во временной области) и вычесть его из исходного сигнала. Или приравнять нулю составляющую спектра на нулевой частоте.

2. Выполнить расчет частот составляющих по выражению

$$f_m = \frac{m \cdot f_s}{N}$$

где  $m = 0, 1, \dots, N - 1$ ;

$N$  – количество отсчетов;

$f_s = \frac{1}{\Delta t}$  – дискрета по частоте;

$\Delta t$  – дискрета по времени.

3. Вычислить значение амплитуды каждой составляющей спектра

$$A(m) = \left| \frac{2 \cdot X(m)}{N} \right|$$

где  $X(m)$  – дискретный спектр, полученный операцией БПФ.

4. Удалить симметрию спектра, только  $0 \leq m \leq \frac{N}{2}$  – требуемые отсчеты;

- b. В пункте №3 говорится об операции БПФ – быстрое преобразование Фурье, которое реализовано в виде функции `fft()`.
- c. После вычисления дискретных частот и амплитуд требуется построить график дискретного амплитудно-частотного спектра  $A_m(f_m)$ .
- d. Проверьте на спектре соответствие амплитуд и частот исходному сигналу.

## ЗАДАНИЕ № 11. Работа с основными классами и функциями математической библиотеки `scipy` языка программирования Python, используемых для обработки данных

Цель — изучить некоторые элементы математической библиотеки `scipy`, широко используемой для математических расчётов с использованием языка программирования Python.

Подключаемые библиотеки: `numpy`, `scipy`, `matplotlib`. (библиотеки `scipy` и `matplotlib` требуют дополнительной установки, так как данные библиотеки не входят в состав стандартных библиотек).

### Описание задания

- 1) Найдите решение следующего нелинейного уравнения:

$$3 \cdot x^2 - 100 \cdot e^{-0,1 \cdot x} = 0$$

- a. найдите решение графическим способом – постройте графики решения;
- b. найдите решение используя специальную функцию

`scipy.optimize.fsolve()`

- 2) Задайте передаточную функцию линейной стационарной динамической системы:

$$W(s) = \frac{2s + 1}{2s^2 + 0,1s + 1}$$

Используя класс передаточной функции `scipy.signal.TransferFunction()`.

- a. найдите полюсы системы, используя свойство `TransferFunction.poles` и покажите их на комплексной плоскости;
- b. найдите переходной процесс системы, используя функцию `step()` и постройте график переходной характеристики;

с. найдите амплитудно-частотную характеристику системы, используя функцию `bode()` и постройте её график.

**3)** Найдите численное решение нелинейного обыкновенного дифференциального уравнения, используя `scipy.integrate.solve_ivp()`:

$$\ddot{x}(t) + x(t) \cdot \dot{x}(t) + x(t) = 0$$

## ЗАДАНИЕ № 12. Работа с основными классами и функциями библиотеки визуализации данных matplotlib языка программирования Python

Цель — изучить некоторые элементы библиотеки matplotlib, широко используемой для визуализации данных расчётов с использованием языка программирования Python.

Подключаемые библиотеки: numpy, scipy, matplotlib.

### Описание задания

#### 1) Работа с функцией plot():

- a. Выведите графики двух любых функций в одной системе координат.
- b. Выведите графики двух любых функций в одном окне но в разных системах координат.
- c. Сделайте «подписи» осей графиков и установите координатную решётку.

#### 2) Работа с функцией bar():

- a. Задайте два массива (5...6 элементов) целочисленных данных и массив названий этих данных, например, `label = ['A1', 'A2', 'A3', 'A4', 'A5']`.
- b. Используйте функцию `bar()` для вывода данных.
- c. Добавьте «легенду» к выведенным данным.

#### 3) Работа с функцией quiver():

Постройте фазовый портрет системы:

$$\ddot{x}(t) + x(t) \cdot \dot{x}(t) + x(t) = 0$$

Используя функцию `quiver()`. Для вычислений потребуется использовать функцию `numpy.meshgrid()`.



4) Работа с функцией `contour()`:

Постройте проекции плоскостей среза трёхмерной поверхности:

$$z(x, y) = -\sin(x^2 + y^2) + 1$$

Используя функцию `contour()`. Для вычислений потребуется использовать функцию `numpy.meshgrid()`.

5) Работа с функцией `plot_surface()`:

Постройте проекции плоскостей среза трёхмерной поверхности:

$$z(x, y) = -\sin(x^2 + y^2) + 1$$

Используя функцию `plot_surface()`. Для вычислений потребуется использовать функцию `numpy.meshgrid()`.

## **Учебно-методическое и информационное обеспечение дисциплины**

1. Сузи, Р.А. Язык программирования Python: учеб. пособие, М.: Интернет-Ун-т Информ. Технологий: Бином. Лаборатория знаний, 2006.
2. Буйначев, С.К., Боклаг, Н.Ю. Основы программирования на языке Python: учебное пособие, Екатеринбург: Издательство Уральского университета, 2014.
3. Сузи, Р.А. Язык программирования Python: учебное пособие, М.: Интернет- Университет Информационных Технологий (ИНТУИТ), 2016
4. Уэс, Маккинли Python и анализ данных: практическое пособие, Саратов: Профобразование, 2017.
5. Балджы, А.С., Хрипунова, М.Б. Математика на Python: учебно-методическое пособие, М.: Прометей, 2018.
6. Гуриков С. Р. Основы алгоритмизации и программирования на Python: учебное пособие, М.: Издательство "ФОРУМ", 2017.
7. Жуков Р. А. Язык программирования Python: практикум: Учебное пособие, М.: ООО "Научно- издательский центр ИНФРА-М", 2020.