

# Python. Вложенные списки

ноябрь, 2024

Иногда нужно хранить данные в виде прямоугольных таблиц. В Python мы используем вложенные списки для представления таких таблиц. Например, чтобы создать таблицу с двумя строками и тремя столбцами, мы можем сделать следующее:

```
A = [[1, 2, 3], [4, 5, 6]]
```

Здесь первая строка списка `A[0]` является списком из чисел `[1, 2, 3]`. То есть `A[0][0] == 1`, значение `A[0][1] == 2`, `A[0][2] == 3`, `A[1][0] == 4`, `A[1][1] == 5`, `A[1][2] == 6`.

Для обработки и вывода списка, как правило, используется два вложенных цикла. Первый цикл по номеру строки, второй цикл по элементам внутри строки. Например, вывести двумерный числовой список на экран построчно, разделяя числа пробелами внутри одной строки, можно так:

```
for i in range(len(A)):
    for j in range(len(A[i])):
        print(A[i][j], end=' ')
    print()
```

## Создание вложенных списков

Пусть даны два числа: количество строк `n` и количество столбцов `m`. Необходимо создать список размером `n × m`, заполненный нулями.

Первый способ: сначала создадим список из `n` элементов (для начала просто из `n` нулей). Затем сделаем каждый элемент списка ссылкой на другой одномерный список из `m` элементов:

```
A = [0] * n
for i in range(n):
    A[i] = [0] * m
```

Другой (но похожий) способ: создать пустой список, потом `n` раз добавить в него новый элемент, являющийся списком-строкой:

```
A = []
for i in range(n):
    A.append([0] * m)
```

**Теорию для данной лабораторной вы можете найти в учебном пособии**

# Аудиторный практикум

## Задание 1

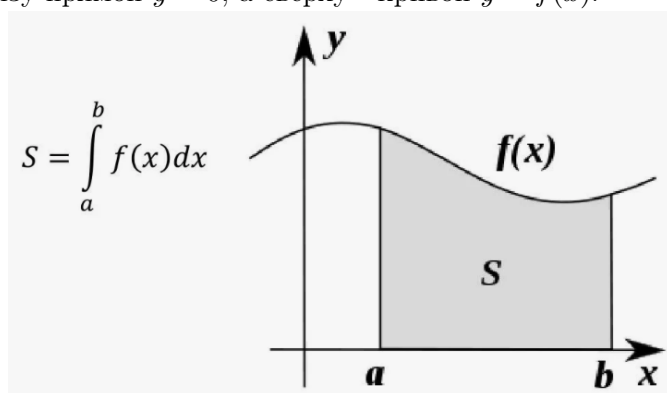
Приобретение навыков вычисления определённых интегралов с использованием численного метода интегрирования (метод прямоугольников)

Пусть задана функция  $f(x)$ , определённым интегралом функции  $f(x)$  от  $a$  до  $b$  называется конечное число, равное пределу

$$\lim_{\Delta x \rightarrow 0} \sum_{i=1}^{\infty} f(x_i) \cdot \Delta x_i$$

где  $x_i, x_{i-1}$  - значения переменной  $x$ , лежащие в интервале от  $[a, b]$ ,  $\Delta x_i = x_i - x_{i-1}$ . Это число обозначается  $\int_a^b f(x) dx$ .

Вспомним из курса школьной математики геометрическое определение интеграла - он равен площади криволинейной трапеции, ограниченной слева и справа вертикальными прямыми  $x = a, x = b$ , снизу прямой  $y = 0$ , а сверху - кривой  $y = f(x)$ .



Эта геометрическая интерпретация лежит в основе численных методов вычисления определённых интегралов. Из них, наиболее распространенными на практике являются метод прямоугольников, метод трапеций и метод Симпсона (парабол).

Суть всех перечисленных методов состоит в нахождении площади не исходной криволинейной трапеции, а площади некоторой более простой геометрической фигуры.

Естественно, что найденная площадь не будет совпадать с площадью криволинейной трапеции – все перечисленные выше методы являются приближенными. Погрешность вычисления интеграла зависит от того, насколько точно контур предложенной фигуры повторяет контур исходной криволинейной трапеции. Рассмотрим методы численного интегрирования подробнее. Для простоты будем иметь дело с равномерными сетками переменной  $x$ .

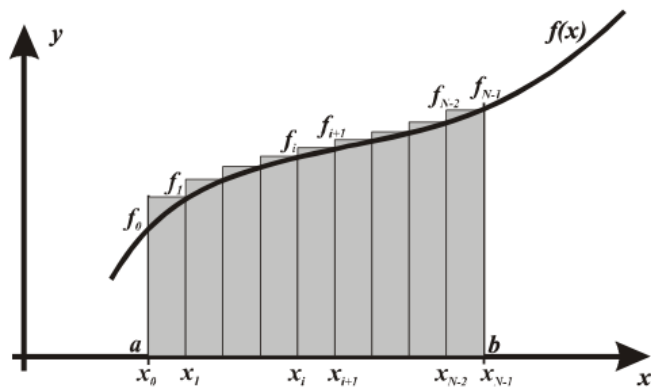
Равномерная сетка  $\{x_i\}$  характеризуется числом точек  $N$  и значениями  $a$  и  $b$  - крайней левой и правой границей интервала переменной  $x$ , и задается следующим образом:

$$x_i = a + i \cdot h$$

где индекс  $i$  пробегает значения от 0 до  $N - 1$ , шаг сетки  $h = (b - a)/N$ .

### Метод прямоугольников

Заменяем фигуру «под графиком функции» на множество прямоугольников, как показано



$$S = \sum_{i=1}^{N-1} f_i * h$$

Давайте реализуем данный метод:

В качестве примера, возьмем  $f(x) = x^2/(10 - x^3)$ ;  $a = -2$ ;  $b = 5$ . Разобьем данный интервал на 1000

```
def func(x):
    return x**2 / (10 + x**3)
```

```
a = -2
b = 5
n = 10000
h = (b - a) / n
```

```
f = []
```

```
i = a
while i < b:
    f.append(func(i))
    i += h
```

```
sum = 0
```

```
for i in f:
    sum += h * i
```

```
print(f"{sum:.2f}")
```

## Задание 2

Дана матрица. Вывести на экран первый и последний столбцы.

Входные данные	Выходные данные
1 2 3 4	1 4
5 6 1 1	5 1
2 5 1 0	2 0
4 3 2 1	4 1

### Задание 3

Дан вложенный список 3×5 элементов. Определить, сколько раз встречается число 7 среди элементов массива.

Входные данные	Выходные данные
1 2 3 5 6 2 4 5 7 7 9 8 7 6 5	3

### Задание 4

Дана матрица выведите номер строки и столбца ячейки, в которой находится максимальный элемент.

Входные данные	Выходные данные
1 2 3 5 6 2 4 5 7 5 9 8 0 6 5	3 1

### Задание 5

Получены значения температуры воздуха за 4 дня с трех метеостанций, расположенных в разных регионах страны:

№ станции	1 день	2 день	3 день	4 день
1	-8	14	-19	-18
2	25	28	26	20
3	11	18	20	25

Т.е. запись показаний в двумерном массиве выглядела бы так:

```
t[0][0]=-8 t[0][1]=-14 t[0][2]=-19 t[0][3]=-18  
t[1][0]=25 t[1][1]=28 t[1][2]=26 t[1][3]=20  
t[2][0]=11 t[2][1]=18 t[2][2]=20 t[2][3]=25
```

1. Распечатать температуру на 2-й метеостанции за 4-й день и на 3-й метеостанции за 1-й день.
2. Распечатать показания термометров всех метеостанций за 2-й день.
3. Определить среднюю температуру на 3-й метеостанции.

## Лабораторная работа 9

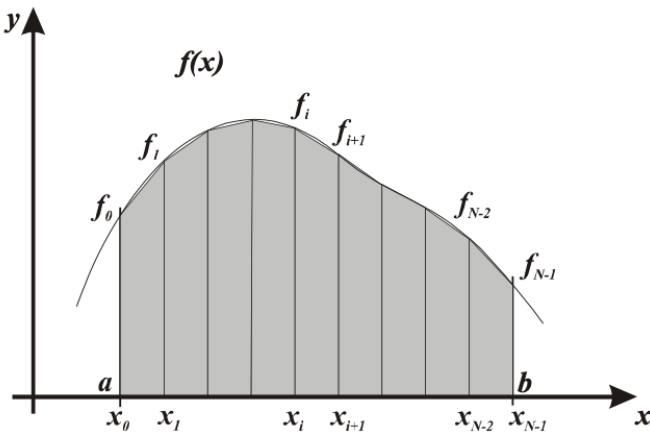
В данной лабораторной работе присутствуют задания повышенной сложности, поэтому оценивание выполненной лабораторной работы будет происходить следующим образом:

Критерии оценивания лабораторной:

- Чтобы получить оценку 4 (75%), необходимо сделать 7 заданий.
- Чтобы получить оценку 5 (90%), необходимо сделать 8 заданий.
- Чтобы получить оценку 5 (100%), необходимо сделать 9 заданий.

### Задание 1

Необходимо реализовать метод трапеций для вычисления приближенного значения интеграла. Формула находится под рисунком. Шаг  $h$  вычисляется точно также как и в методе прямоугольников. Выведите значения интегралов для  $N = 10, 100, 1000$ .



$$\int_a^b f(x)dx \approx \frac{h}{2}(f_0 + f_{N-1}) + \sum_{i=1}^{n-2} f_i$$

### Задание 2

Реализуйте на языке Python программу случайной генерации магического квадрата. (Что такое магический квадрат? Читайте тут -> [https://ru.wikipedia.org/wiki/Магический\\_квадрат](https://ru.wikipedia.org/wiki/Магический_квадрат))

Входные данные	Выходные данные
	2 7 6 9 5 1 4 3 8

### Задание 3

Мореплаватель Александр нашел карту с описанием местоположения пиратского клада. Карта представлена в виде вложенного списка, где каждый элемент - это список с координатами сокровища на острове.

```
treasure_map = [  
    [1, 2],  
    [3, 5],  
    [5, 6],
```

[7, 8]

]

Напишите программу, которая:

Принимает на вход  $n$  - количество сокровищ, карту (координаты сокровищ) и координаты точки, где стоит Саша. Определяет, на каком расстоянии от Александра находится ближайшее сокровище. Выводит координаты ближайшего сокровища.

Входные данные	Выходные данные
Количество сокровищ: 4 Координаты сокровищ: 1 2 3 5 5 6 7 8 Координаты Александра: 2 3	1 2

Подсказка. URL: <https://nauka.club/wp-content/uploads/902786/fullsize.jpg>

#### Задание 4

Вы работали шеф-поваром в ресторане “Вкусно и быстро”. У вас есть меню ресторана в виде вложенного списка: название блюда, ингредиенты (список), цена.

```
menu = [  
    ["Пицца Маргарита", ["мука", "томаты", "сыр", "базилик"], 10],  
    ["Салат Цезарь", ["салат", "курица", "сыр", "сухарики"], 8],  
    ["Суп Томатный", ["томаты", "лук", "морковь", "картофель"], 6]  
]
```

Напишите программу, которая позволит вам:

1. Отобразить меню ресторана.
2. Найти блюдо по названию и отобразить его ингредиенты и цену. (Вы вводите название блюда).
3. Добавить новое блюдо в меню.
4. Изменить цену блюда (Вы вводите название и новую цену).

#### Задание 5

Дана матрица размером  $n \times m$ . Симметричная ей относительно главной диагонали матрица называется транспонированной к данной. Она имеет размеры  $m \times n$ : строки исходной матрицы становятся столбцами транспонированной, столбцы исходной матрицы становятся строками транспонированной.

Для данной матрицы постройте транспонированную и выведите ее на экран.

Входные данные	Выходные данные
3 4 11 12 13 14 21 22 23 24 31 32 33 34	11 21 31 12 22 32 13 23 33 14 24 34

## Задание 6

Дана квадратная матрица  $n \times n$ . Поменяйте местами элементы, стоящие на главной и побочной диагонали, при этом каждый элемент должен остаться в том же столбце (то есть в каждом столбце нужно поменять местами элемент на главной диагонали и на побочной диагонали).

Входные данные:

$n$  - количество строк и столбцов

и сама матрица вводится с клавиатуры.

Входные данные	Выходные данные
3	7 2 9
1 2 3	4 5 6
4 5 6	1 8 3
7 8 9	

## Задание 7

Кинотеатр “Звездный” готовится к премьере долгожданного блокбастера. В зале  $n$  рядов по  $m$  мест в каждом. Заказ билетов поступает онлайн, а информация о свободных местах хранится в виде двумерного списка, где 1 - проданный билет, а 0 - свободное место.

Поступила группа из  $k$  зрителей, которые хотят сесть рядом в одном ряду.

**Ваша задача:**

Напишите программу, которая поможет кассиру “Звездного” определить, можно ли посадить эту группу.

**Ввод:**

$n$  - количество рядов в кинотеатре  $m$  - количество мест в ряду  $n$  строк с  $m$  числами (0 или 1), разделенных пробелами - информация о занятости мест в каждом ряду  $k$  - количество билетов в заказе

**Вывод:**

Номер ряда, в котором есть  $k$  подряд идущих свободных мест (номер ряда начинается с 1). Если таких рядов несколько, то выведите номер наименьшего подходящего ряда. Если подходящего ряда нет, выведите число 0.

Входные данные	Выходные данные
5 5	1
0 1 0 0 0	
1 0 0 1 0	
0 0 0 0 1	
1 1 1 0 0	
0 0 0 0 0	
3	

**Объяснение:**

В первом ряду есть 3 свободных места подряд, поэтому программа выводит 1 (номер ряда).

## Задание 8

Даны два числа  $n$  и  $m$ . Создайте матрицу  $n \times m$  и заполните ее по следующим правилам:

Числа, стоящие в строке 0 или в столбце 0 равны 1 ( $A[0][j] = 1$ ,  $A[i][0] = 1$ ). Для всех остальных элементов матрицы  $A[i][j] = A[i-1][j] + A[i][j-1]$ , то есть каждый элемент равен сумме двух элементов, стоящих слева и сверху от него.

Выведите данную матрицу на экран, отводя на вывод каждого элемента матрицы ровно 6 символов (см. пример).

Входные данные	Выходные данные					
4 6	1	1	1	1	1	1
	1	2	3	4	5	6
	1	3	6	10	15	21
	1	4	10	20	35	56

## Задание 9

В данном задании вам предлагается реализовать простейший вариант игры "Морской бой". Дано поле 4x4 и 4 однопалубных корабля, которые расставлены рандомно. Пользователю каждый раз показывается поле:

```
. . . .
. . . .
. . . .
. . . .
```

и предлагается сделать выстрел (например, кордината (1, 2))

Если он попал в корабль, то данная ячейка становится равной X

```
. X . .
. . . .
. . . .
. . . .
```

иначе поле остается без изменений.

Игра будет идти до тех пор, пока пользователь не нашел все корабли.

Но на поле присутствует одна бомба (B), при попадании на которую пользователь проигрывает. Положение бобмы определяется рандомом.

При положительном исходе игры, попимо вывода поздравления, выведите за сколько шагов пользователь нашел все 4 корабля.



## Решения аудиторного практикума

### Задание 2

```
matrix = []
n = 0
m = 0
while n <= 0 or m <= 0:
    try:
        n = int(input("Введите количество строк: "))
        m = int(input("Введите количество столбцов: "))
    except ValueError:
        print("При создании матрицы произошла ошибка. Введите ее размер снова")

print("Давайте заполним матрицу (вводите по одному элементу): ")
for i in range(n):
    line = []
    for j in range(m):
        line.append(input("> "))
    matrix.append(line)

print("Исходная матрица:")
for i in range(n):
    for j in range(m):
        print(matrix[i][j], end=" ")
    print()

print("Результат:")

for i in range(n):
    print(matrix[i][0], matrix[i][-1])
```

### Задание 3

```
m = [
    [1, 2, 3, 5, 6],
    [2, 4, 5, 7, 7],
    [9, 8, 7, 6, 5]
]

count_7 = 0

for i in range(len(m)):
    for j in range(len(m[i])):
        if m[i][j] == 7:
            count_7 += 1

print(count_7)
```

### Задание 4

```
m = [
    [1, 2, 3, 5, 6],
    [2, 4, 5, 7, 5],
    [9, 8, 0, 6, 5]
]

max = m[0][0]
row = 0
col = 0

for i in range(len(m)):
    for j in range(len(m[i])):
        if m[i][j] > max:
            max = m[i][j]
            row = i
            col = j

print(row + 1, col + 1)
```

### Задание 5

```
t = [[-8, 14, -19, -18],
     [25, 28, 26, 20],
     [11, 18, 20, 25]]

print("Температура на 2 метеостанции в 4 день:", t[1][-1])
print("Температура на 3 метеостанции в 1 день:", t[2][0])
```

```
print("-" * 100)

print("Температура со всех метеостанций во 2 день:")
for i in range(3):
    print(i + 1, "метеостанция: ", t[i][1])

print("-" * 100)

avg = 0
for i in range(len(t[2])):
    avg += t[2][i]

avg /= len(t[2])
print("Средняя температура на 3-й метеостанции:", avg)
```