

МИНОБРНАУКИ РОССИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Самарский национальный исследовательский университет имени  
академика С.П. Королева»  
Институт экономики и управления  
Кафедра математических методов в экономике

**ОТЧЕТ О ЛАБОРАТОРНОМ ПРАКТИКУМЕ**  
по дисциплине «Объектно-ориентированное программирование и  
информационная безопасность»

Выполнил      И. И. Иванов  
                         группа 7350-380305D  
Проверил  
  
оценка    дата

**Самара 2024**

## ОГЛАВЛЕНИЕ

Основные цели и задачи .....	3
Лабораторная работа №1 .....	4
Лабораторная работа №2 .....	5
Лабораторная работа №3 .....	6
Лабораторная работа №4 .....	7
Лабораторная работа №5 .....	9
Лабораторная работа №6 .....	11
Лабораторная работа №7 .....	13
Список использованных источников.....	14

## Основные цели и задачи

- Познакомиться со средой разработки Jupyter Notebook.
- Изучить основные типы данных, команды ввода и вывода данных.
- Освоить математические операции, операции сравнения, операции преобразования типов и логические операции.
- Рассмотреть функции и методы для работы с последовательностями, списками, строками и словарями.
- Ознакомиться с составными инструкциями, инструкциями ветвления, циклами и генераторами списков.
- Научиться создавать и применять собственные функции.
- Изучить библиотеку NumPy и освоить работу с массивами.

# Лабораторная работа №1

1. Сколько секунд в часе? Используйте Python как калькулятор и умножьте количество секунд в минуте (60) на количество минут в часе (60).

Ввод [1]: `60 * 60`

Out[1]: 3600

2. Присвойте результат вычисления предыдущего упражнения переменной, которая называется `seconds_per_hour`.

Ввод [2]: `seconds_per_hour = 60 * 60`  
`print(seconds_per_hour)`  
`seconds_per_hour`

3600

Out[2]: 3600

3. Сколько секунд в сутках? Используйте для вычисления переменную `seconds_per_hour`. Сохраните результат вычисления в переменной `seconds_per_day`.

Ввод [3]: `seconds_per_day = seconds_per_hour * 24`  
`print(seconds_per_day)`

86400

4. Разделите значение переменной `seconds_per_day` на значение переменной `seconds_per_hour`.

Ввод [4]: `print(seconds_per_day / seconds_per_hour)`

24.0

5. Разделите значение переменной `seconds_per_day` на значение переменной `seconds_per_hour` используя целочисленное деление. Совпадает ли полученный результат с ответом на предыдущее упражнение, если не учитывать символы `.0` в конце?

Ввод [5]: `print(seconds_per_day // seconds_per_hour)`

24

Совпадает.

## Лабораторная работа №2

1. Напишите программный код, который запрашивает у пользователя строку (строка может быть произвольной длины) и выводит пользователю на экран самый первый и самый последний символы строки.

Ввод [4]: 

```
str = input("Введите строку текста: ")
print("Первый символ: ", str[0])
print("Последний символ: ", str[-1])
```

Введите строку текста: Абырвалк  
Первый символ: А  
Последний символ: к

2. Напишите программный код, который запрашивает у пользователя два числа (числа не обязательно целые), записывает их в переменные x и y и выводит с помощью функции print() на экран произведение этих чисел.

Ввод [5]: 

```
x = float(input("Введите число X: "))
y = float(input("Введите число Y: "))
print("Произведение чисел равно = ", x * y)
```

Введите число X: 44  
Введите число Y: 55  
Произведение чисел равно = 2420.0

3. Напишите программный код, который позволяет получить строку из двух первых и двух последних символов строки, введенной с клавиатуры.

Ввод [2]: 

```
s = input("Введите строку текста: ")
n = len(s)
print("Новая строка: ", s[:2] + s[-2:])
```

Введите строку текста: 1234567890  
Новая строка: 1290

4. Создайте список years\_list, содержащий год, в который вы родились, и каждый последующий год вплоть до вашего пятого дня рождения. Например, если вы родились в 1980 году, список будет выглядеть так: years\_list = [1980, 1981, 1982, 1983, 1984, 1985].

Ввод [8]: 

```
years = [1957, 1958, 1959, 1960, 1961, 1962]
print(years)
```

[1957, 1958, 1959, 1960, 1961, 1962]

5. С помощью операции взятия элемента по индексу выведите год, в который был ваш третий день рождения. Помните, в первый год вам было 0 лет.

Ввод [9]: 

```
print(years[3])
```

1960

## Лабораторная работа №3

1. Создайте список, содержащий несколько строковых элементов.

Введите номер элемента списка с клавиатуры.

Преобразуйте с помощью операции взятия элемента по индексу и метода .upper() элемент с заданным номером к верхнему регистру, а затем выведите список.

Изменился ли список?

```
Ввод [1]: cheese = ['mozzarella', 'brynzа', 'адыгейский', 'маскарпоне', 'фета']
n = int(input("Введите номер (0-4)"))
cheese[n] = cheese[n].upper()
print(cheese)
```

Введите номер (0-4):

['моцарелла', 'брынза', 'АДЫГЕЙСКИЙ', 'маскарпоне', 'фета']

Список изменился.

2. Удалите преобразованный элемент из списка и выведите список на экран.

```
Ввод [2]: cheese.remove(cheese[n])
print(cheese)

['моцарелла', 'брынза', 'маскарпоне', 'фета']
```

3. Создайте англо-русский (или любой другой) словарь (несколько пар слов) и выведите его на экран.

Выведите пару вариантов перевода.

```
Ввод [4]: dictionary = {'dog':'собака', 'cat':'кот', 'walrus':'морж', 'sheep':'овца', 'kitten':'котенок'}
print(dictionary['walrus'])
print(dictionary['dog'])

морж
собака
```

4. Создайте многоуровневый словарь (например, life). Используйте несколько строк для ключей верхнего уровня (например, 'animals', 'plants' и 'other'). Сделайте так, чтобы один ваш ключ (например, 'animals') ссылался на другой словарь, имеющий несколько ключей (например, 'cats', 'octopi' и 'emus'). Сделайте так, чтобы один ключ (например, 'cats') ссылался на список строк со значениями (например, 'Henri', 'Grumpy' и 'Lucy'). Остальные ключи должны ссылаться на пустые словари.

```
Ввод [5]: life = {'animals': {'cats': ['Henri', 'Grumpy', 'Lucy'], 'octopi': {}, 'emus': {}}, 'plants': {}, 'other': {}}
print(life)

{'animals': {'cats': ['Henri', 'Grumpy', 'Lucy'], 'octopi': {}, 'emus': {}}, 'plants': {}, 'other': {}}
```

5. Выведите на экран высокоуровневые ключи словаря. Выведите на экран ключи вложенного словаря (например, life['animals']). Выведите значения вложенного словаря.

```
Ввод [6]: print(life.keys())
print(life['animals'].keys())
print(life['animals'].values())

dict_keys(['animals', 'plants', 'other'])
dict_keys(['cats', 'octopi', 'emus'])
dict_values([[ 'Henri', 'Grumpy', 'Lucy'], {}, {}])
```

## Лабораторная работа №4

1. Введите какую-либо вещественную переменную. Далее напишите условные проверки (if, else и elif), чтобы вывести строку 'too low', если значение переменной меньше 7, 'too high', если оно больше 7, и 'just right', если равно 7.

```
Ввод [ ]: while True: #Итерационный цикл
    a1fa = float(input("Введите число = "))
    if a1fa < 7:
        print("too low")
    elif a1fa > 7:
        print("too high")
    else:
        print("just right")
    print()
```

2. Введите две переменные.

Напишите цикл while, внутри которого будет происходить сравнение переменных.

Выведите значения переменных и строку 'too low', если значение первой переменной меньше значения второй переменной. Увеличьте значение ПЕРВОЙ переменной и продолжите цикл.

Если значения переменных равны, выведите строку 'found it!' и выйдите из цикла.

Если значение первой переменной больше значения второй переменной, выведите строку 'oops' и выйдите из цикла.

Попробуйте использовать форматный вывод, чтобы избежать сдвига при выводе двухзначных чисел.

```
Ввод [ ]: while True:
    a1 = float(input("Введите первое число = "))
    a2 = float(input("Введите второе число = "))

    while True:
        if a1 < a2:
            print("a1 = %d, a2 = %d, too low" % (a1, a2))
            a1 = a1 + 1
        elif a1 == a2:
            print("found it!")
            break
        else:
            print("oops")
            break
    a1 = 1
    print()
```

3. Введите с клавиатуры количество элементов списка слов, затем организуйте ввод списка с подсказкой.

Выведите список на экран с номерами элементов.

Введите произвольное слово и определите, есть ли это слово в списке.

Если есть, то выведите его номер.

Если нет, то выведите сообщение.

```
Ввод [ ]: n = int(input("Введите количество элементов списка = "))
spisok = []
for i in range(n):
    text = input("Введите слово = ")
    spisok.append(text)

for i in range(n):
    print("Слово [%d, %d] = %s, spisok[%d]" % (i, i, spisok[i]))

while text != "stop": #Итерационный цикл
    text = input("Введите слово для проверки = ")
    if text in spisok:
        print("Есть такое слово под номером = %d, spisok.index(text)" % spisok.index(text))
    else:
        print("Нет такого слова!!!")
    print()
text = ""
```

Создать англо-русский (или любой другой) словарь (несколько пар слов).

1. Организовать ввод слова на исходном языке и вывод перевода. В случае отсутствия слова вывести сообщение.

```
Ввод [ ]: word = ""
dictionary = { 'apple': 'яблоко',
              'banana': 'банан',
              'orange': 'апельсин',
              'pineapple': 'ананас',
              'grape': 'виноград' }

while word != "stop": ## бесконечный цикл
    word = input("Введите слово = ")

    if word not in dictionary.keys():
        print("Нет такого слова!")
    else:
        print("Перевод: ", dictionary[word])
    print()
word = ""
```

2. Сделать так, чтобы исходные слова можно было вводить в любом регистре.

```
Ввод [ ]: dictionary = { 'apple': 'яблоко',
                      'banana': 'банан',
                      'orange': 'апельсин',
                      'pineapple': 'ананас',
                      'grape': 'виноград' }

while word != "stop": ## бесконечный цикл
    word = input("Введите слово = ")
    word = word.lower() ## преобразуем в нижний регистр
    if word not in dictionary.keys():
        print("Нет такого слова!")
    else:
        print("Перевод: ", dictionary[word])
    print()
word = ""
```

3. Сделать так, чтобы слова можно было вводить на любом языке.

```
Ввод [ ]: dictionary = { 'apple': 'яблоко',
                      'banana': 'банан',
                      'orange': 'апельсин',
                      'pineapple': 'ананас',
                      'grape': 'виноград' }

word = ""
while word != "stop": ## бесконечный цикл
    word = input("Введите слово = ")
    word = word.lower() ## преобразуем в нижний регистр
    if word not in dictionary.keys() and word not in dictionary.values():
        print("Нет такого слова!")
    elif word in dictionary.keys():
        print("Перевод: ", dictionary[word])
    else:
        keys = [key for key in dictionary if dictionary[key] == word]
        print("Перевод: ", keys)
    print()
word = ""
```

Введите слово = ананас  
Перевод: ['pineapple']

## Лабораторная работа №5

**1. Введите с клавиатуры произвольную строку, которая состоит из двух слов разделенных пробелом. Затем вызовите функцию, которая возвращает строку, в которой слова поменялись местами. Результат выведите на экран.**

```
Ввод [7]: def change(s):  
    new_s = s[s.find(" ") + 1:] + " " + s[:s.find(" ")]  
    return new_s  
  
print("Введите строку, состоящую из двух слов, разделенных пробелом:")  
s = input()  
print(change(s))
```

Введите строку, состоящую из двух слов, разделенных пробелом:  
Самарский университет  
университет Самарский

```
Ввод [17]: def change(s):  
    words = s.split()  
    words[0], words[-1] = words[-1], words[0]  
    new_s = " ".join(words)  
    return new_s  
  
print("Введите строку, состоящую из двух слов, разделенных пробелом:")  
s = input()  
print(change(s))
```

Введите строку, состоящую из двух слов, разделенных пробелом:  
Самарский университет  
университет Самарский

**2. Введите количество элементов в списке, затем в цикле введите список целых чисел. Вызовите функцию, которая определяет количество различных чисел.**

```
Ввод [26]: def count(s):  
    return len(set(s)) # Считает из списка уникалы (так называется элемент) и подсчитывает его длину  
  
n = int(input("Введите количество чисел: "))  
  
s = []  
for i in range(n):  
    s.append(int(input("Введите целое число = ")))  
    s = s + [int(input("Введите целое число = "))]  
  
print("Количество различных чисел =", count(s))
```

Введите количество чисел: 6  
Введите целое число = 3  
Введите целое число = 15  
Введите целое число = 2  
Введите целое число = 15  
Введите целое число = 4  
Введите целое число = 2  
Количество различных чисел = 4

**3. Напишите функцию, которая генерирует пароль (случайным образом). Количество символов в пароле задает пользователь. Для решения данной задачи можно воспользоваться функцией `random.choice(sequence)` (из модуля `random`), которая возвращает случайный элемент непустой последовательности `sequence` (можно в качестве последовательности использовать список).**

```
Ввод [16]: import random # Подключаем модуль random  
  
def pass_gen(count):  
    # Список, из которого будет генерироваться пароль  
    arr = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '!', '@', '#', '$', '%', '&', '^', '*', '&#x2D;', '&#x26;', '&#x27;', '&#x28;', '&#x29;', '&#x3B;', '&#x3C;', '&#x3E;', '&#x3F;', '&#xA0;']  
    pass = "" # Пароль, собираемый пароль  
    for i in range(count):  
        pass += random.choice(arr) # Добавляем в пароль случайный символ  
    return pass  
  
n = int(input("Введите количество символов для пароля = "))  
print(pass_gen(n))
```

Введите количество символов для пароля = 10  
hbe4?aI!fe

4. Рассмотрим строки, состоящие только из заглавных английских букв. Например, рассмотрим строку AAAABSSSSCCDDDD. Длина этой строки равна 14. Поскольку строка состоит только из английских букв, повторяющиеся символы могут быть удалены и заменены числами, определяющими количество повторений. Таким образом, данная строка может быть представлена как 4A5B5C4D. Длина такой строки 7. Описанный метод мы назовем упаковкой строки. Напишите программу с функцией, которая берет упакованную строчку и восстанавливает по ней исходную строку. Каждый символ повторяется не более 9 раз. Максимальная длина строки не превышает 80 (проверку делать не нужно, данное ограничение просто нужно иметь в виду и не вводить строку более чем из 80 символов).

Ввод [24]:

```
def encode(s):
    st = "" # В эту строку будем записывать упакованную строку
    for i in range(len(s)): # Пока не встретим
        if s[i].isalpha(): # Если символ является числом
            st += s[i+1]*(int(s[i])-1) # В строку добавляем столько символов сколько переданной (число-1) раз
        else: st += s[i] # Если это не число и строка из символов
    return st

str = input("Введите упакованную строку = ") # Записали в переменную упакованную строку
print(encode(str))
```

Введите упакованную строку = 5Г2ёЦ7ж  
ГГГГёёЦжжжжжж

## Лабораторная работа №6

1. Создайте массив `arr_1` размером 3x4, все элементы которого равны 3.

```
Ввод [30]: import numpy as np
arr_1 = np.full((3, 4), 3)
arr_1
```

```
Out[30]: array([[3, 3, 3, 3],
               [3, 3, 3, 3],
               [3, 3, 3, 3]])
```

2. Создайте массив `arr_2`, заполненный случайными целыми числами от 0 до 9, размером 2x4.

```
Ввод [31]: arr_2 = np.random.randint(0, 10, size=(2,4))
arr_2
```

```
Out[31]: array([[5, 2, 7, 1],
               [2, 0, 7, 6]])
```

3. Выведите на экран количество элементов в созданных массивах `arr_1` и `arr_2`.

```
Ввод [33]: print("Количество элементов в массиве arr_1 =", arr_1.size)
print("Количество элементов в массиве arr_2 =", arr_2.size)
```

```
Количество элементов в массиве arr_1 = 12
Количество элементов в массиве arr_2 = 8
```

4. Соедините массивы `arr_1` и `arr_2` по нулевой оси.

```
Ввод [34]: z = np.concatenate((arr_1, arr_2), 0)
z
```

```
Out[34]: array([[3, 3, 3, 3],
               [3, 3, 3, 3],
               [3, 3, 3, 3],
               [5, 2, 7, 1],
               [2, 0, 7, 6]])
```

5. Создайте массив `arr_3` из двумерного кортежа произвольных целых чисел размером 4x3.

```
Ввод [35]: arr3 = ((1, 2, 3, 5, 7, 14), (1, 6, 0, 9, 4, -5))
arr_3 = np.array(arr3)
arr_3
```

```
Out[35]: array([[ -1,  2,  3,  5,  7, 14],
               [ 1,  6,  0,  9,  4, -5]])
```

6. Умножьте каждый элемент массива `arr_3` на случайное целое число (от 2 до 10), затем прибавьте к каждому элементу массива квадрат этого числа. Результат запишите в новый массив `arr_4`. Выведите число и полученный массив.

```
Ввод [45]: import random
chislo = random.randint(2, 10)
arr_4 = arr_3 * chislo + chislo**2
print("Случайное число:", chislo)
print("Массив arr_4: ", arr_4)
```

```
Случайное число: 6
Массив arr_4:
[[ 30  48  54  66  78 120]
 [ 42  72  36  90  60  6]]
```

**7. Преобразуйте массив arr\_3 к двумерному массиву 4x3. Результат запишите в массив arr\_5.**

```
Ввод [46]: arr_5 = arr_3.reshape(4, 3)
print("Массив arr_5: ", arr_5)
```

```
Массив arr_5:
[[ 30  48  54]
 [ 66  78 120]
 [ 42  72  36]
 [ 90  60   6]]
```

**8. Определите минимальные элементы массива arr\_5 вдоль первой оси.**

```
Ввод [47]: print("Минимальные элементы вдоль первой оси =", np.min(arr_5, axis=0))
```

```
Минимальные элементы вдоль первой оси = [30 66 36  6]
```

**9. Вычислите среднее арифметическое всех элементов массива arr\_5.**

```
Ввод [51]: print("Среднее арифметическое элементов массива arr_5 = ", np.mean(arr_5))
```

```
Среднее арифметическое элементов массива arr_5 = 58.5
```

## Лабораторная работа №7

1. Создайте массив `arr_6`, состоящий из квадратов чисел от 0 до 10.

```
Ввод [2]: import numpy as np  
arr_6 = np.arange(11)**2  
arr_6
```

```
Out[2]: array([ 0,  1,  4,  9, 16, 25, 36, 49, 64, 81, 100])
```

2. Выведите каждый второй элемент массива `arr_6`.

```
Ввод [116]: arr_6[arr_6::2]  
[ 1  9 25 49 81]
```

3. Выведите элементы массива `arr_6` в обратном порядке.

```
Ввод [52]: arr_6[arr_6::-1]  
[100 81 64 49 36 25 16  9  4  1  0]
```

4. Каждому второму элементу массива `arr_6` присвойте значение 2.

```
Ввод [117]: arr_6[1::2] = 2  
arr_6  
Out[117]: array([ 0,  2,  4,  2, 16,  2, 36,  2, 64,  2, 100], dtype=int32)
```

5. Проверьте, есть ли в массиве `arr_6` значение 49.

```
Ввод [3]: x = 49  
if x in arr_6:  
    print("Значение", x, "есть в массиве arr_6")  
else:  
    print("В массиве arr_6 нет значения", x)  
Значение 49 есть в массиве arr_6
```

6. Создайте двумерный массив `A`, состоящий из положительных и отрицательных чисел. Из массива `A` выберите все отрицательные элементы и запишите их в одномерный массив `B`.

```
Ввод [4]: A = np.array([[1, -2, 3, -4], [5, -6, 7, -8], [9, -10, 11, -12]])  
negative = A < 0  
B = A[negative]  
B  
Out[4]: array([-2, -4, -6, -8, -10, -12])
```

## Список использованных источников

### Основная литература:

1. Программирование на языке высокого уровня Python : учеб. пособие для прикладного бакалавриата / Д. Ю. Федоров. — 2-е изд., перераб. и доп. — М. : Издательство Юрайт, 2019. — 161 с. — (Серия : Бакалавр. Прикладной курс).
2. Python для детей. Самоучитель по программированию / Джейсон Бриггс ; пер. с англ. Станислава Ломакина ; [науч. ред. Д. Абрамова]. — М. : Манн, Иванов и Фербер, 2017. — 320 с.
3. Простой Python. Современный стиль программирования. — СПб.: Питер, 2016. — 480 с.: ил. — (Серия «Бестселлеры O'Reilly»).

### Интернет-ресурсы:

1. Бесплатный курс Python (питон), самостоятельное обучение программированию с нуля [Электронный ресурс] / <https://code-basics.com/ru>. — Режим доступа: <https://code-basics.com/ru/languages/python>, свободный.
2. Самоучитель Python [Электронный ресурс] / <https://pythonworld.ru>. Режим доступа: <https://pythonworld.ru/samouchitel-python>, свободный.
3. Уроки по языку программирования Python [Электронный ресурс] / <https://devpractice.ru>. Режим доступа: <https://devpractice.ru/python-lessons>, свободный.