

Департамент образования города Москвы

Государственное автономное образовательное учреждение высшего  
образования города Москвы «Московский Городской Педагогический  
Университет»

Институт цифрового образования  
Департамент информатики, управления и технологий

ЛАБОРАТОРНАЯ РАБОТА №1.1

по дисциплине «Практикум по Python»

Направление подготовки 38.03.05 – «бизнес-информатика»

Профиль подготовки «Аналитик бизнес-процессов: автоматизация и  
управление бизнесом» (очная форма обучения)

Тема: «Работа с переменными и типами данных.»

Выполнила: Губайдуллина А. И.

Студентка группы АБП – 231

Проверил: Босенко Т. М. доцент

Москва

2025

## **Цели:**

1. Ознакомиться с основными типами данных в языке Python: числа, строки и структуры данных.
2. Научиться работать с операциями для различных типов данных.
3. Закрепить навыки написания программ с использованием базовых конструкций языка.
4. Развить умение формализовать задачи, разрабатывать алгоритмы их решения и переносить их на программный код.

## **Задачи:**

1. Изучить основные арифметические и логические операции для работы с числами.
2. Научиться работать со строками, включая индексацию, срезы и основные операции
3. Изучить базовые структуры данных Python (списки и словари) и основные действия с ними.
4. Применить аналитический подход к решению задач: разбить решение на этапы, формализовать алгоритм, перенести его в программный код.

№ 1.1.1. Составьте программу, которая запрашивает у пользователя 2 целых числа и выполняет операции:

- арифметические: + , - , \* , / , // , % , \*\* ;
- сравнение: < , <= , > , >= , != , == , выводя на экран результат каждого действия. В случае получения вещественного результата, округлите его до 2-х знаков после запятой (используя функцию round() ).

Решение:



# Задание task\_01\_01

""" Составьте программу, которая запрашивает у пользователя 2 целых числа и выполняет операции:

- арифметические: +, -, \* , / , // , %, \*\*;
- сравнение: <, <=, >, >=, !=, ==,

выводя на экран результат каждого действия. В случае получения вещественного результата, округлите его до 2-х знаков после запятой (используя функцию round()).

"""

# Выполнила: Губайдуллина А.И.

# Группа: АБП-231

a = int(input("a="))

b = int(input("b="))

print(a + b)

print(a - b)

print(a \* b)

print(a / b)

print(a // b)

print(a % b)

print(a \*\* b)

print(a < b)

print(a <= b)

print(a > b)

print(a >= b)

print(a != b)

print(a == b)



a=3

b=7

10

-4

21

0.42857142857142855

0

3

2187

True

True

False

False

True

False

Рис. 1. Результат выполнения арифметических и сравнительных операций

№ 1.1.2. Вычислите значение следующего выражения (аргументы - целые числа и вводятся с клавиатуры):

$$f = \frac{\sqrt[3]{\frac{x^5 + 7}{|-6| \times y}}}{7 - z \bmod y}$$

Округлите результат до 3-х знаков после запятой, используя функцию round()

Решение:

```
[ ] # Задание task_01_02
    """ Вычислите значение следующего выражения (аргументы - целые числа и
        вводятся с клавиатуры):
        Округлите результат до 3-х знаков после запятой, используя функцию round() .
        """

    # Выполнила: Губайдуллина А.И.
    # Группа: АБП-231

    x = int(input("Введите целое число "))
    y = int(input("Введите целое число "))
    z = int(input("Введите целое число "))

    res = round((((x**5 + 7) / (abs(-6) * y)) ** (1 / 3) / 7 - z % y), 3)
    print(res)
```

```
➡ Введите целое число 0
   Введите целое число 1
   Введите целое число 2
   0.15
```

Рис. 2. Результат нахождения корней выражения

№ 1.1.3. Дана электрическая цепь, состоящая из 2-х последовательно соединенных проводников (сопротивление каждого известно). Найти общее сопротивление цепи (округление результата необходимо выполнить до 1-го знака после запятой). Решение (начало и общий ход мыслей): Аналитическое решение. Проводники соединены последовательно, поэтому общее сопротивление цепи будет равно сумме их сопротивлений. Сопротивление

может быть вещественным числом, поэтому для его хранения необходимо выбрать тип float.

Алгоритм.

- ввести 2 вещественных значения - сопротивления проводников;
- вычислить их сумму;
- вывести ответ на экран.

Программа.

Решение:

```
[ ] # Задание task_01_03
    """ Дана электрическая цепь, состоящая из 2-х последовательно соединенных
    проводников (сопротивление каждого известно). Найти общее сопротивление цепи
    (округление результата необходимо выполнить до 1-го знака после запятой).
    Решение (начало и общий ход мыслей):
    Аналитическое решение.
    Проводники соединены последовательно, поэтому общее сопротивление
    цепи будет равно сумме их сопротивлений. Сопротивление может быть
    вещественным числом, поэтому для его хранения необходимо выбрать тип float.
    Алгоритм.
    - ввести 2 вещественных значения - сопротивления проводников;
    - вычислить их сумму;
    - вывести ответ на экран.
    Программа.
    """

    # Выполнила: Губайдуллина А.И.
    # Группа: АБП-231

    # Сопротивление первого проводника
    r1 = float(12.1)
    # Сопротивление второго проводника
    r2 = float(16.1)

    # Общее сопротивление
    r = round(r1 + r2, 2)
    print(r)
```

⇒ 28.2

Рис. 3. Результат нахождения общего сопротивления цепи

№ 1.1.4. Дано двузначное и трехзначное число. Для каждого выведите на экран сумму и произведение цифр.

Решение:

```
[ ] # Задание task_01_04
    """ Дано двузначное и трехзначное число. Для каждого выведите на экран сумму
    и произведение цифр.
    """

    # Выполнила: Губайдуллина А.И.
    # Группа: АБП-231

    # Двузначное число
    num2 = int(input("Введите двузначное число "))

    # Трехзначное число
    num3 = int(input("Введите трехзначное число "))

    # 1-я цифра числа 'num2'
    num2_1 = num2 // 10
    # 2-я цифра числа 'num2'
    num2_2 = num2 % 10

    # Сумма цифр числа 'num2'
    num2_s = num2_1 + num2_2
    print("Сумма цифр двухзначного числа: ", num2_s)

    # Произведение цифр числа 'num2'
    num2_p = num2_1 * num2_2
    print("Произведение цифр двухзначного числа: ", num2_p)

    # 1-я цифра числа 'num3'
    num3_1 = num3 // 100
    # 2-я цифра числа 'num3'
    num3_2 = (num3 // 10) % 10
    # 3-я цифра числа 'num3'
    num3_3 = num3 % 10
```

```
# Сумма цифр числа 'num3'
num3_s = num3_1 + num3_2 + num3_3
print("Сумма цифр трехзначного числа: ", num3_s)

# Произведение цифр числа 'num3'
num3_p = num3_1 * num3_2 * num3_3
print("Произведение цифр трехзначного числа: ", num3_p)
```

➡ Введите двузначное число 66  
Введите трехзначное число 666  
Сумма цифр двухзначного числа: 12  
Произведение цифр двухзначного числа: 36  
Сумма цифр трехзначного числа: 18  
Произведение цифр трехзначного числа: 216

Рис. 4. Результат выполнения программы для двухзначного и трехзначного числа

№ 1.1.5. С начала суток прошло  $m$  минут ( $0 < m \leq 24 \cdot 60$ ). Определите:

- ☐ целое количество часов, прошедших с начала суток;
- ☐ количество минут, прошедших с момента начала последнего часа.

Решение:

```
[ ] # Задание task_01_05
    """ С начала суток прошло m минут (0<m≤24*60). Определите:
        целое количество часов, прошедших с начала суток;
        количество минут, прошедших с момента начала последнего часа.

        """

    # Выполнила: Губайдуллина А.И.
    # Группа: АБП-231

    # Количество минут
    m = int(input("Введите количество минут (0 < m < 1440): "))

    # Количество часов, прошедших с начала суток
    h = m // 60
    print("Целое количество часов: ", h)

    # Количество минут, прошедших с момента начала последнего часа
    m2 = m % 60
    print("Целое количество минут с начала последнего часа: ", m2)
```

```
⇒ Введите количество минут (0 < m < 1440): 123
   Целое количество часов: 2
   Целое количество минут с начала последнего часа: 3
```

Рис. 5. Результат выполнения программы

№ 1.1.6. Дано уравнение  $ax + b = 0$  и отрезок  $[m;n]$ . Ответьте на вопрос, попадает ли решение уравнения в указанный отрезок.

Решение (начало и общий ход мыслей):

1. Аналитическое решение. Коэффициенты уравнения и величина отрезка - произвольны, возьмем целые числа. Решение  $x$  при этом может быть дробным. После определения  $x$  составим логическое выражение попадания в отрезок, что и будет ответом.

2. Алгоритм.

3. - ввести все неизвестные величины;
  4. - определить корень уравнения;
  5. - составить логическое выражение попадания корня в область;
  6. - вывести результат попадания на экран.
7. Программа.

```
[ ] # Задание task_01_06
    """ Дано уравнение  $ax + b = 0$  и отрезок  $[m;n]$ . Ответьте на вопрос, попадает ли
    решение уравнения в указанный отрезок.
    Решение (начало и общий ход мыслей):
    1. Аналитическое решение.
    Коэффициенты уравнения и величина отрезка - произвольны,
    возьмем целые числа. Решение  $x$  при этом может быть дробным.
    После определения  $x$  составим логическое выражение попадания в
    отрезок, что и будет ответом.
    2. Алгоритм.
    3. - ввести все неизвестные величины;
    4. - определить корень уравнения;
    5. - составить логическое выражение попадания корня в область;
    6. - вывести результат попадания на экран.
    7. - Программа.

    """

    # Выполнила: Губайдуллина А.И.
    # Группа: АБП-231

    a = int(input("Введите число a: "))
    b = int(input("Введите число b: "))
    m = int(input("Введите число m: "))
    n = int(input("Введите число n: "))

    print("Попадает: ", m <= x <= n)
```

```
➡ Введите число a: 5
    Введите число b: 6
    Введите число m: 0
    Введите число n: 9
    Попадает: False
```

Рис. 6. Результат выполнения программы



№ 1.1.7. Составьте программу, которая запрашивает название футбольной команды и повторяет его на экране со словами  
... - чемпион!

После этого выполните:

- используя операцию дублирования, нарисуйте черту (набор "-" ), длиной, равной размеру названия команды;
- преобразуйте строку в нижний регистр и выведите на экран:
- длину наименования команды;
- есть ли в наименовании команды буква "п" (True/False)?
- сколько раз повторяется буква "а" ?

Решение:

```
[ ] # Задание task_01_07
    """ Составьте программу, которая запрашивает название футбольной команды и
    повторяет его на экране со словами
    ... - чемпион!
    После этого выполните:
    - используя операцию дублирования, нарисуйте черту (набор "-" ), длиной,
    равной размеру названия команды;
    - преобразуйте строку в нижний регистр и выведите на экран:
    4
    - длину наименования команды;
    - есть ли в наименовании команды буква "п" (True/False)?
    - сколько раз повторяется буква "а" ?

    """

    # Выполнила: Губайдуллина А.И.
    # Группа: АБП-231

    team = input("Введите название команды: ")
    print(team + " - чемпион!")

    print("-" * len(team))

    # Название команды в нижнем регистре
    team_lower = team.lower
    print(team_lower)
    print(len(team))

    print("п" in team)
    print(team.count("а"))
```



Введите название команды: цска  
цска - чемпион!

```

<built-in method lower of str object at 0x780174ce48a0>
4
False
1

```

Рис. 7. Результат выполнения программы

№ 1.1.8. Составьте программу, которая запрашивает название государства и его столицы, а затем выводит сообщение:

Государство - ..., столица - ...

На месте многоточий должны быть выведены соответствующие значения.

Решение:

```

[ ] # Задание task_01_08
    """ Составьте программу, которая запрашивает название государства и его столицы, а
    затем выводит сообщение:
    Государство - ..., столица - ...
    На месте многоточий должны быть выведены соответствующие значения

    """

    # Выполнила: Губайдуллина А.И.
    # Группа: АБП-231

    # Название государства
    country = input("Введите название государства: ")

    # Название столицы
    capital = input("Введите название столицы: ")

    print("Государство - {}, столица - {}".format(country, capital))

```

```

➡ Введите название государства: Италия
  Введите название столицы: Рим
  Государство - Италия, столица - Рим

```

Рис. 8. Результат выполнения программы

№ 1.1.9. Дано слово объектно-ориентированный . Используя индексацию и срезы составьте из него слова объект , ориентир , тир , кот , рента и выведите их на экран.

Решение:

```
[ ] # Задание task_01_09
    """ Дано слово объектно-ориентированный . Используя индексацию и срезы составьте
    из него слова объект , ориентир , тир , кот , рента и выведите их на экран.

    """

    # Выполнила: Губайдуллина А.И.
    # Группа: АБП-231

    word = "объективно-ориентированный"

    w1 = word[:6] # слово объект
    w2 = word[11:19] # слово ориентир
    w3 = word[16:19] # слово тир
    w4 = word[4:5] + word[9:10] + word[16:17] # слово кот
    w5 = word[12:13] + word[14:16] + word[16:17] + word[21:22] # слово рента

    print(w1, w2, w3, w4, w5, sep="\n")
```

⇒ объект  
ориентир  
тир  
кот  
рента

Рис. 9. Результат выполнения индексации и срезов

№ 1.1.10. Создайте 2 пустых списка и выполните операции, описанные в заготовке.

Решение:

```
[ ] # Задание task_01_10
    """ Создайте 2 пустых списка и выполните операции, описанные в заготовке.

    """

    # Выполнила: Губайдуллина А.И.
    # Группа: АБП-231

    a = []
    b = []

    # Добавить 2 вещественных числа 4.5 и 3.4 в список a с помощью append
    a.append(4.5)
    a.append(3.4)

    # Добавить 2 вещественных числа 8.7 и 1.3 в список 'a' с помощью extend
    a.extend([8.7, 1.3])

    # Добавить 2 вещественных числа (14.5, 3.4) в список 'b' с помощью append
    b.append(14.5)
    b.append(3.4)

    # Добавить 2 вещественных числа 8.7 и 11.3 в список 'b' с помощью extend
    b.extend([8.7, 11.3])

    # Вставить целое число 100 на 2-е и 4-е место списка 'a'
    a.insert(1, 100)
    a.insert(3, 100)

    # Вставить целое число 200 на 1-е и 3-е место списка 'b'
    b.insert(0, 200)
```

```
[ ] # Вставить целое число 100 на 2-е и 4-е место списка 'a'
    a.insert(1, 100)
    a.insert(3, 100)

    # Вставить целое число 200 на 1-е и 3-е место списка 'b'
    b.insert(0, 200)
    b.insert(2, 200)

    # Вывести списки на экран
    print("Исходные списки: ")
    print(a)
    print(b)

    # Удалить первые элементы из списков 'a' и 'b'
    del a[0]
    del b[0]

    # Удалить элемент 100 из списка 'a'
    a.remove(100)
    # Удалить элемент 200 из списка 'b'
    b.remove(200)

    # Вывести списки на экран
    print("\nПосле удалений:")
    print(a)
    print(b)

    # Создать множества из списков 'a' и 'b', а также их пересечение
    sa = set(a)
    sb = set(b)
    sa_and_sb = sa & sb
```

```
[ ] sa_and_sb = sa & sb

# Вывести экран уникальные элементы каждого списка и общие
print("\nУникальные элементы:")
print("только в списке a: ", sa - sb)
print("только в списке b: ", sb - sa)

print("общие:", sa_and_sb)

# Соединить списки 'a' и 'b'
c = a + b
print(c)

# Отсортировать список 'c' по возрастанию и убыванию
c_asc = sorted(c)
c_desc = sorted(c, reverse=True)

# Среднее арифметическое элементов списка 'c', расположенных на четных местах
even_elements = c[1::2]
sr_ar = sum(even_elements) / len(even_elements)
# Среднее геометрическое элементов списка 'c', расположенных на нечетных местах
from math import prod
odd_elements = c[::2]
sr_geom = prod(odd_elements) ** (1 / len(odd_elements))

# Максимальный и минимальный элементы
c_max = max(c)
c_min = min(c)

# Вывести результаты на экран
print("\nИтоговые:")

print("\n3-й:", c)
print("Среднее арифметическое (четные места): ", sr_ar)
print("Среднее геометрическое (нечетные места): ", sr_geom)
print("Максимальный элемент: ", c_max)
print("Минимальный элемент: ", c_min)
```



Исходные списки:

```
[4.5, 100, 3.4, 100, 8.7, 1.3]
[200, 14.5, 200, 3.4, 8.7, 11.3]
```

После удалений:

```
[3.4, 100, 8.7, 1.3]
[14.5, 3.4, 8.7, 11.3]
```

Уникальные элементы:

```
только в списке a: {1.3, 100}
только в списке b: {11.3, 14.5}
общие: {8.7, 3.4}
[3.4, 100, 8.7, 1.3, 14.5, 3.4, 8.7, 11.3]
```

Итоговые:

```
3-й: [3.4, 100, 8.7, 1.3, 14.5, 3.4, 8.7, 11.3]
Среднее арифметическое (четные места): 29.0
Среднее геометрическое (нечетные места): 7.815762540646051
Максимальный элемент: 100
Минимальный элемент: 1.3
```

Рис. 10. Результат выполнения операций над списками

№ 1.1.11. Создайте пустой словарь для хранения информации о себе и выполните операции, описанные в заготовке.

Решение:

```
[28] # Задание task_01_11
      """ Создайте пустой словарь для хранения информации о себе и выполните операции,
      описанные в заготовке.

      """

      # Выполнила: Губайдуллина А.И.
      # Группа: АБП-231

      # 1. Создание словаря

      info = {}

      # Добавить значения для ключей "фio", "дата_рождения", "место_рождения"
      info["фio"] = 'Алина'
      info["дата_рождения"] = '03.09.05'
      info["место_рождения"] = 'Татарстан'

      # Напечатать словарь
      print(info)

      # Создать ключ "хобби" со значением = список из строк -
      # наименований хобби (например "плавание" и т.д.)
      info["хобби"] = []
      info["хобби"].append('плавание')

      # Добавить "программирование" в список хобби
      info["хобби"].append('программирование')

      # Создать ключ "животные" со значением = кортеж из строк -
      # имен домашних животных (например "кошка Мурка" и т.д.)
      pet1 = 'кошка Мурка'
      pet2 = 'собака Пушка'
```

```
8] info["животные"] = (pet1, pet2)

# Создать ключ "ЕГЭ" и поместить в него пустой словарь
info["ЕГЭ"] = {}

# В словарь info["ЕГЭ"] добавить информацию о сданных экзаменах,
# где ключ - наименование предмета, а значение - количество баллов
info["ЕГЭ"]["математика"] = 70
info["ЕГЭ"]["обществознание"] = 81
info["ЕГЭ"]["русский"] = 85

# Добавить экзамен, который не был сдан, после чего удалить его
info["ЕГЭ"]["история"] = 0
del info["ЕГЭ"]["история"]

# Создать ключ "вузы" и поместить в него пустой словарь
info["вузы"] = {}

# В словарь info["вузы"] добавить информацию о вузах,
# где ключ - аббревиатура вуза, а значение -
# количество баллов для специальности, на которую хотели поступить
info["вузы"]["МГТУ им. Баумана"] = 245
info["вузы"]["МИРЭА"] = 264

# 2. Вывод на экран
print("Данные:")
# Получившийся словарь
print(info)

# Список экзаменов ЕГЭ в алфавитном порядке
# (используйте метод ``keys()``, преобразовав результат в кортеж)
exams = tuple(sorted(info["ЕГЭ"].keys()))
print("Предметы:", exams)
```



```
[28] # Список вузов в алфавитном порядке
uni = tuple(sorted(info['вузы'].keys()))
print("Вузы:", uni)

# 3. Ответы на вопросы
print("\nОтветы на вопросы:")

# Выделить имя из info["фио"]
name = info["фио"]
# Начинается на гласную? (True/False)
starts_with_vowel = name[0].lower() in 'ауёиоуыэюя'
print("* мое имя начинается на гласную букву:", starts_with_vowel)

# Выделить месяц из info["дата_рождения"]
month = int(info["дата_рождения"].split('.')[1])
# Родился зимой или летом? (True/False)
born_in_winter_or_summer = month in [12, 1, 2]
print("* родился летом или зимой:", born_in_winter_or_summer)

# Количество хобби и первое из них
hobbies_count = len(info["хобби"])
first_hobby = info["хобби"][0]
print("* у меня {} хобби, первое \ "{}\ {}".format(hobbies_count, first_hobby))

# Количество сданных экзаменов
print("* после окончания школы сдавал {} экз.".format(info["ЕГЭ"]))

# Сумма баллов по экзаменам
sum_mark = info["ЕГЭ"]["математика"] + info["ЕГЭ"]["обществознание"] + info["ЕГЭ"]["русский"]
print("* сумма баллов = {}".format(sum_mark))
```

```

8] # Сумма баллов по экзаменам
sum_mark = info["ЕГЭ"]["математика"] + info["ЕГЭ"]["обществознание"] + info["ЕГЭ"]["русский"]
print("** сумма баллов = {}".format(sum_mark))

# Максимальный балл среди сданных
max_mark = max(info["ЕГЭ"])
print("** макс. балл = {}".format(max_mark))

# Количество вузов, в которые Вы проходите по баллам
# Подсказка: определить, проходите Вы или нет, можно простым сравнением
# суммы баллов с проходным баллом вуза - ``True/False``.
total_score = sum(info["ЕГЭ"].values())

# Для того, чтобы определить количество таких вузов, преобразуйте
# сравнение в целое число (используя ``int()``) и сложите все сравниваемые вузы.
vuz_count = sum(map(lambda x: int(total_score >= x), info["вузы"].values()))
print("** кол-во вузов в которые прохожу: {}".format(vuz_count))

{ 'фio': 'Алина', 'дата_рождения': '03.09.05', 'место_рождения': 'Татарстан' }
Данные:
{ 'фio': 'Алина', 'дата_рождения': '03.09.05', 'место_рождения': 'Татарстан', 'хобби': ['плавание', 'прог]
Предметы: ('математика', 'обществознание', 'русский')
Вузы: ('МГТУ им. Баумана', 'МИРЭА')

Ответы на вопросы:
* мое имя начинается на гласную букву: True
* родился летом или зимой: False
* у меня 2 хобби, первое "плавание"
* после окончания школы сдавал {'математика': 70, 'обществознание': 81, 'русский': 85} экз.
* сумма баллов = 236
* макс. балл = русский
* кол-во вузов в которые прохожу: 0

```

Рис. 11. Результат выполнения операций над списками

**Вывод:** ознакомилась с основными типами данных в языке Python: числа, строки и структуры данных; научилась работать с операциями для различных типов данных; закрепила навыки написания программ с использованием базовых конструкций языка; развила умение формализовать задачи, разрабатывать алгоритмы их решения и переносить их на программный код.