

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВВГУ»)

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АНАЛИЗА ДАННЫХ  
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

ОТЧЕТ  
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1  
по дисциплине  
«Информатика и программирование»

Студент		
гр. GGG-NN-LL	_____	А.И. Студент
Ассистент		
преподавателя	_____	М.В. Водяницкий

## Задание

Выполнить задания на Python и оформить отчет по стандартам ВВГУ.

**Задание 1.** Объявить четыре переменные с любыми значениями для каждого из основных типов данных (int, float, str, bool).

**Задание 2.** Объявить две переменные: одну для вашего имени, другую для вашего возраста. Затем вывести значения переменных в консоль.

**Задание 3.** Объявить три переменные: одну со значением 342, вторую со значением 56.2 и третью со значением '43'. Найти сумму всех трех чисел и результат записать в новую переменную. Вывести значение новой переменной в консоль.

**Задание 4.** Объявить переменные a и b со значениями 3 и 8. Вычислить значение уравнения  $(a + 4b)(a - 3b) + a^2$  и вывести результат в консоль.

**Задание 5.** Написать программу для вычисления площади и периметра прямоугольника. Данные сторон прямоугольника должны запрашиваться у пользователя и вводиться с консоли. Результат работы программы вывести в консоль.

**Задание 6.** Вывести в консоль букву w, составленную из символов \*, расположенных в три строки.

Пример:

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

**Задание 7.** Создать две переменные с любыми именами и любыми числовыми значениями. Вывести в консоль результат выполнения всех арифметических операторов (7 операторов) и операторов сравнения (6 операторов).

**Задание 8.** Создать две переменные: одну для вашего имени, другую для вашего возраста. С помощью f-строки вывести в консоль строку вида: Меня зовут <ваше имя>, мне <ваш возраст> лет.

**Задание 9.** Разбить предложение Съешь еще этих мягких французских булок, да выпей чаю на несколько переменных, содержащих по одному-два слова. Затем с помощью конкатенации строк необходимо собрать предложение вновь и вывести в консоль (не забывайте про пробелы).

**Задание 10.** Составить предложение, состоящее из строки «Нет! Да!», которая повторяется 4 раза (используйте умножение строк).

**Задание 11.** Запросить на ввод с консоли три числа, разделенных запятой, и записать их в три отдельные переменные. Затем первое и третье число сложить и результат целочисленно разделить на второе число. Конечный результат вывести в консоль в виде строки: Результат вычисления: <результат>.

**Задание 12.** Запросить на ввод с консоли слово, содержащее не менее 10 символов. С помощью срезов вывести:

- 1) первые 4 символа;
- 2) последние 2 символа;
- 3) символы от 4 до 8;
- 4) перевернутое слово.

## Содержание

1	Выполнение работы .....	3
1.1	Задание 1 .....	3
1.2	Задание 2 .....	3
1.3	Задание 3 .....	3
1.4	Задание 4 .....	4
1.5	Задание 5 .....	4
1.6	Задание 6 .....	5
1.7	Задание 7 .....	6
1.8	Задание 8 .....	6
1.9	Задание 9 .....	7
1.10	Задание 10 .....	7
1.11	Задание 11 .....	8
1.12	Задание 12 .....	9

## 1 Выполнение работы

### 1.1 Задание 1

В данном задании были созданы четыре переменные, каждая из которых относится к одному из основных типов данных: целые числа, числа с плавающей точкой, строки и логические значения. После объявления переменные были выведены на экран с помощью функции `print()`. На рисунке 1 представлен код полученной программы.

```
1 # task1.py
2 # Объявление переменных разных типов
3 a = 10          # int
4 b = 3.14        # float
5 c = "Привет"    # str
6 d = True        # bool
7
8 print(a, b, c, d)
```

Рисунок 1 – Листинг программы для задания 1

Пояснение работы программы:

- 1) переменная `a` имеет тип `int` и хранит целое число;
- 2) переменная `b` имеет тип `float` и хранит дробное число;
- 3) переменная `c` содержит строковое значение, заключенное в кавычки;
- 4) переменная `d` является логической и может принимать значения `True` или `False`.

После выполнения программы в консоль последовательно выводятся значения всех четырех переменных.

### 1.2 Задание 2

В данном задании были созданы две переменные: одна хранит имя пользователя, другая – его возраст. После присвоения значений обе переменные выводятся в консоль с помощью функции `print()`. На рисунке 2 представлен код программы.

```
1 # task2.py
2 name = "Марк"
3 age = 24
4 print(name, age)
```

Рисунок 2 – Листинг программы для задания 2

Пояснение работы программы:

- 1) Переменная `name` содержит строковое значение – имя пользователя.
- 2) Переменная `age` содержит целое значение – возраст пользователя.
- 3) Функция `print()` выводит оба значения через пробел в одной строке.

### 1.3 Задание 3

В этом задании необходимо было объявить три переменные: две числовые и одну строковую, содержащую число. Затем требовалось найти сумму всех трёх чисел и вывести результат в консоль. На рисунке 3 представлен код программы.

```
1 # task3.py
2 a = 342
3 b = 56.2
4 c = '43'
5
6 result = a + b + int(c)
7 print(result)
```

Рисунок 3 – Листинг программы для задания 3

Пояснение работы программы:

- 1) Переменная `a` хранит целое число 342.
- 2) Переменная `b` хранит дробное число 56.2.
- 3) Переменная `c` содержит строку «43», которая преобразуется в число функцией `int()`.

- 4) Сумма трёх чисел сохраняется в переменной `result` и выводится функцией `print()`.

В результате выполнения программа выводит сумму трёх значений, приведённых к числовому типу.

### 1.4 Задание 4

В задании необходимо было вычислить значение выражения  $(a + 4b)(a - 3b) + a^2$  для заданных переменных. На рисунке 4 представлен код решения.

```
1 # task4.py
2 a = 3
3 b = 8
4
5 result = (a + 4*b) * (a - 3*b) + a ** 2
6 print(result)
```

Рисунок 4 – Листинг программы для задания 4

Пояснение работы программы:

- 1) Переменным `a` и `b` присвоены значения 3 и 8.
- 2) Выражение вычисляется в соответствии с математическими правилами приоритета операций.
- 3) Оператор `**` используется для возведения числа `a` в квадрат.
- 4) Результат сохраняется в переменной `result` и выводится в консоль.

После выполнения программы в консоли отображается вычисленное значение выражения.

## 1.5 Задание 5

Цель задания – реализовать программу для вычисления площади и периметра прямоугольника, стороны которого вводятся пользователем с консоли. На рисунке 5 представлен код программы.

```
1 # task5.py
2 width = float(input("Введите ширину прямоугольника: "))
3 height = float(input("Введите высоту прямоугольника: "))
4
5 area = width * height
6 perimeter = 2 * (width + height)
7
8 print("Площадь:", area)
9 print("Периметр:", perimeter)
```

Рисунок 5 – Листинг программы для задания 5

Пояснение работы программы:

- 1) С помощью функции `input()` запрашиваются длины сторон прямоугольника, которые преобразуются в тип `float`.
- 2) Переменная `area` вычисляется как произведение сторон прямоугольника.
- 3) Переменная `perimeter` вычисляется по формуле  $2 * (width + height)$ .
- 4) Оба результата выводятся на экран при помощи функции `print()`.

После запуска программы пользователь вводит значения сторон и получает площадь и периметр прямоугольника.

## 1.6 Задание 6

В данном задании необходимо было вывести в консоль символическую (ASCII-art) букву `w`, составленную из символов `*`, расположенных в три строки. Для реализации задачи использовались последовательные вызовы функции `print()`, каждый из которых формирует одну строку рисунка. На рисунке 6 представлен код программы.

```
1 # task6.py
2 print("* * *")
3 print(" * * * *")
4 print("  *  *")
```

Рисунок 6 – Листинг программы для задания 6

Пояснение работы программы:

- 1) Каждый вызов функции `print()` формирует отдельную строку символов.
- 2) Расположение пробелов и звёзд подобрано таким образом, чтобы визуально получилась буква `w`.
- 3) Программа не требует ввода данных пользователем и выводит фиксированный результат.

В результате выполнения в консоли отображается следующая фигура, показанная на рисунке 7



```
mark@Mac-mini-Mark report-demo % python code/task6.py
* * *
* * *
* * *
mark@Mac-mini-Mark report-demo %
```

Рисунок 7 – Пример вывода программы из задания 6

## 1.7 Задание 7

Задача заключалась в демонстрации работы основных арифметических и логических операторов языка Python. Для этого были созданы две переменные с произвольными числовыми значениями и проведены вычисления с использованием семи арифметических и шести операторов сравнения. На рисунке 8 представлен код программы.

```
1 # task7.py
2 x = 10
3 y = 3
4
5 print("Сложение:", x + y)
6 print("Вычитание:", x - y)
7 print("Умножение:", x * y)
8 print("Деление:", x / y)
9 print("Целочисленное деление:", x // y)
10 print("Остаток от деления:", x % y)
11 print("Возведение в степень:", x ** y)
12
13 print("Равно:", x == y)
14 print("Не равно:", x != y)
15 print("Больше:", x > y)
16 print("Меньше:", x < y)
17 print("Больше или равно:", x >= y)
18 print("Меньше или равно:", x <= y)
```

Рисунок 8 – Листинг программы для задания 7

Пояснение работы программы:

- 1) Переменным  $x$  и  $y$  присвоены значения 10 и 3.
- 2) В коде последовательно выполняются операции сложения, вычитания, умножения, деления, целочисленного деления, нахождения остатка и возведения в степень.
- 3) Затем производится сравнение двух чисел по различным условиям (равенство, неравенство, больше, меньше и т.д.).
- 4) Каждая операция сопровождается пояснительным текстом в выводе.

После выполнения программы в консоли выводятся результаты всех арифметических и логических операций между переменными.



## 1.8 Задание 8

В этом задании необходимо было вывести строку, содержащую имя и возраст пользователя, с использованием f-строки. Такой способ форматирования позволяет удобно вставлять значения переменных непосредственно внутрь строки. На рисунке 9 представлен код программы.

```
1 # task8.py
2 name = "Марк"
3 age = 24
4 print(f"Меня зовут {name}, мне {age} лет")
```

Рисунок 9 – Листинг программы для задания 8

Пояснение работы программы:

- 1) Созданы две переменные: name – содержит имя, и age – возраст.
- 2) Для объединения текста и значений переменных используется f-строка.
- 3) F-строка позволяет вставлять значения переменных в нужные места строки без явной конкатенации.

После выполнения программы в консоли отображается сообщение вида: «Меня зовут Марк, мне 24 лет».

## 1.9 Задание 9

В этом задании нужно было собрать исходное предложение из нескольких переменных, каждая из которых хранит часть текста. Для объединения строк использовалась операция конкатенации при помощи знака +. На рисунке 10 представлен код программы.

```
1 # task9.py
2 part1 = "Съешь"
3 part2 = "еще этих"
4 part3 = "мягких французских"
5 part4 = "булок, да выпей чаю"
6
7 sentence = part1 + " " + part2 + " " + part3 + " " + part4
8 print(sentence)
```

Рисунок 10 – Листинг программы для задания 9

Пояснение работы программы:

- 1) Каждая из четырех переменных (part1, part2, part3, part4) содержит часть предложения.
- 2) При помощи операции + и добавления пробелов создаётся итоговая строка.
- 3) Полученная строка сохраняется в переменной sentence и выводится на экран.

Результатом выполнения программы является восстановленное предложение: «Съешь ещё этих мягких французских булок, да выпей чаю».

### 1.10 Задание 10

В данном задании необходимо было создать строку «Нет! Да!», повторяющуюся четыре раза подряд. Для решения использовалась операция умножения строк, позволяющая повторить заданную последовательность символов нужное количество раз. На рисунке 11 представлен код программы.

```
1 # task10.py
2 phrase = "Нет! Да!"
3 result = phrase * 4
4 print(result)
```

Рисунок 11 – Листинг программы для задания 10

Пояснение работы программы:

- 1) В переменной `phrase` хранится исходная строка «Нет! Да!».
- 2) С помощью операции умножения `phrase * 4` строка повторяется четыре раза подряд.
- 3) Результат сохраняется в переменной `result` и выводится в консоль.

После выполнения программы в консоли отображается результат: «Нет! Да!Нет! Да!Нет! Да!Нет! Да!».

### 1.11 Задание 11

В этом задании требовалось запросить у пользователя три числа, разделённые запятыми, а затем вычислить выражение  $(a + c) // b$  с использованием целочисленного деления. Результат необходимо вывести в консоль в виде поясняющего сообщения. На рисунке 12 представлен код программы.

```
1 # task11.py
2 numbers = input("Введите три числа через запятую: ")
3 a, b, c = map(float, numbers.split(","))
4
5 result = int((a + c) // b)
6 print(f"Результат вычисления: {result}")
```

Рисунок 12 – Листинг программы для задания 11

Пояснение работы программы:

- 1) Пользователь вводит три числа, разделённые запятыми, которые считываются функцией `input()`.
- 2) С помощью `split(",")` введённая строка разбивается на три элемента.
- 3) Функция `map(float, ...)` преобразует каждое значение в тип с плавающей точкой.
- 4) После этого вычисляется результат целочисленного деления суммы первого и третьего чисел на второе.
- 5) Итоговый результат выводится на экран с помощью форматированной строки.

После выполнения программа запрашивает ввод данных и выводит строку вида: «Результат вычисления: 15».

### 1.12 Задание 12

В последнем задании лабораторной работы необходимо было запросить у пользователя слово, содержащее не менее десяти символов, и при помощи срезов вывести отдельные части строки, а также её обратный вариант. На рисунке 13 представлен код программы.

```
1 # task12.py
2 word = input("Введите слово не( менее 10 символов): ")
3
4 print("Первые 4 символа:", word[:4])
5 print("Последние 2 символа:", word[-2:])
6 print("Символы с 4 по 8:", word[3:8])
7 print("Перевернутое слово:", word[::-1])
```

Рисунок 13 – Листинг программы для задания 12

Пояснение работы программы:

- 1) Программа запрашивает у пользователя ввод слова длиной не менее десяти символов.
- 2) Срез `word[:4]` возвращает первые четыре символа строки.
- 3) Срез `word[-2:]` возвращает последние два символа.
- 4) Срез `word[3:8]` выводит символы с четвёртого по восьмой.
- 5) Срез `word[::-1]` переворачивает строку, выводя её в обратном порядке.

После выполнения программа последовательно отображает результаты всех операций со срезами для введённого слова.