

Министерство науки и высшего образования Российской  
Федерации Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №7**  
**дисциплины «Основы кроссплатформенного**  
**программирования»**

Выполнил:

Боженко Александр Иванович

1 курс, группа ИТС-б-о-21-1,

11.03.02 «Инфокоммуникационные  
технологии и системы связи»,

направленность (профиль)

«Инфокоммуникационные системы и сети»,

очная форма обучения

\_\_\_\_\_  
(подпись)

Руководитель практики:

Воронкин Р.А., канд. техн. наук, доцент

кафедры инфокоммуникаций

\_\_\_\_\_  
(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2022 г.

## Тема: Работа с функциями в языке Python

**Цель работы:** приобретение навыков по работе с функциями при написании программ с помощью языка программирования Python версии 3.x.

### Ход работы:

#### Задание 1.

Решить следующую задачу: основная ветка программы, не считая заголовков функций, состоит из двух строки кода. Это вызов функции *test()* и инструкции `if __name__ == '__main__'`. В ней запрашивается на ввод целое число. Если оно положительное, то вызывается функция *positive()*, тело которой содержит команду вывода на экран слова "Положительное". Если число отрицательное, то вызывается функция *negative()*, ее тело содержит выражение вывода на экран слова "Отрицательное".

```
1  import numbers
2
3
4  def test():
5      number = int(input('введите число: '))
6      if number > 0:
7          positive()
8      elif number < 0:
9          negative()
10     else:
11         print("я вас не совсем понял. ;)")
12         test()
13
14 def positive():
15     print('положительное')
16
17 def negative():
18     print('отрицательное')
19
20 test()
```

Рис 1.

#### Задание 2.

Решите следующую задачу: в основной ветке программы вызывается функция *cylinder()*, которая вычисляет площадь цилиндра. В теле *cylinder()* определена функция *circle()*, вычисляющая площадь круга по формуле  $S = \pi r^2$ . В теле *cylinder()* у пользователя спрашивается, хочет ли он получить только площадь боковой поверхности цилиндра, которая вычисляется по формуле  $S_{\text{бок}} = 2\pi r h$ , или полную площадь цилиндра. В последнем случае к площади боковой поверхности цилиндра должен добавляться удвоенный результат вычислений функции *circle()*.

```

work-2.py
1  import math
2  import pi=3.14
3  import return
4  def cylinder(r, h):
5      from math import pi
6  def circle(r): return pi*r**2
7  s = 2*pi*r*h
8  if input('Full area? [y/n]: ') == 'y': \
9  s += 2*circle(r)
10 return s
11
12 r, h = 1, 1
13 print('s =', cylinder(r, h))

```

Рис 2.

### Задание 3

Решите следующую задачу: напишите функцию, которая считывает с клавиатуры числа и перемножает их до тех пор, пока не будет введен 0. Функция должна возвращать полученное произведение. Вызовите функцию и выведите на экран результат ее работы.

```

work-3.py > ...
AWS: Add Debug Configuration | AWS: Edit De
1  def test():
2      answer = 1
3      while 1:
4          num = int(input())
5          if not num: break
6          answer *= num
7      return(answer)
8  print(test())

```

Рис 3.

### Задание 4.

Решите следующую задачу: напишите программу, в которой определены следующие четыре функции:

1. Функция `get_input()` не имеет параметров, запрашивает ввод с клавиатуры и возвращает в основную программу полученную строку.
2. Функция `test_input()` имеет один параметр. В теле она проверяет, можно ли переданное ей значение преобразовать к целому числу. Если можно, возвращает логическое *True*. Если нельзя – *False*.
3. Функция `str_to_int()` имеет один параметр. В теле преобразовывает переданное значение к целочисленному типу. Возвращает полученное число.
4. Функция `print_int()` имеет один параметр. Она выводит переданное значение на экран и ничего не возвращает.

```
AWS: Add Debug Configuration | AWS: Edit Debug Configuration
1 def getInput(): return(input('запрашиваю ввод: '))
AWS: Add Debug Configuration | AWS: Edit Debug Configuration
2 def testInput(n):
3     try: n = int(n); return(True)
4     except: return(False)
AWS: Add Debug Configuration | AWS: Edit Debug Configuration
5 def strToInt(n): return(int(n))
AWS: Add Debug Configuration | AWS: Edit Debug Configuration
6 def printInt(n): print(n)
```

Рис 4.

#### Задание 5.

8. Определить, есть ли в кортеже хотя бы одна тройка соседних чисел, в которой средний элемент больше своих «соседей», т. е. предшествующего и последующего. В случае положительного ответа определить номера элементов первой из таких троек.

```
individual-1.py
C: > Users > user > job-6 > dev > individual-1.py
1  from random import randint
2  # создали список из случайных чисел
3  list = [(randint(1, 100)) for _ in range(31)]
4  print(list)
5
6  # i учитывает индекс элемента в списке
7  i = 1
8  # цикл перебирает все значения списка начиная с элемента с индексом 1
9  # и до конца (предыдущий элемент имеет индекс = 0)
10 for num in list[1::]:
11     # условие, которое проверяет, соседние элементы
12     if list[i - 1] < num and list[i + 1] < num:
13         print(list[i - 1], list[i], list[i + 1])
14         # при нахождении выходим из цикла
15         break
16     else:
17         # иначе прибавляем к индексу 1 и снова сравниваем
18         i += 1

[41, 17, 45, 2, 13, 4, 77, 54, 56, 67, 43, 59, 63, 61, 79, 1, 73, 76, 77, 64, 42, 41, 95, 58, 53, 36, 23, 68, 1, 73, 19]
17 45 2
PS C:\Users\user>
```

Рис 5.

## Ответы на вопросы

1. Каково назначение функций в языке программирования Python?

Функции можно сравнить с небольшими программками, которые сами по себе, т. е. автономно, не исполняются, а встраиваются в обычную программу. Нередко их так и называют – подпрограммы. Других ключевых отличий функций от программ нет. Функции также при необходимости могут получать и возвращать данные. Только обычно они их получают не с ввода (клавиатуры, файла и др.), а из вызывающей программы. Сюда же они возвращают результат своей работы.

2. Каково назначение операторов def и return ?

3. Каково назначение локальных и глобальных переменных при написании функций в Python?

В языке программирования Python функции определяются с помощью оператора `def` выход из функции и передача данных в то место, откуда она была вызвана, выполняется оператором `return`.

4. Как вернуть несколько значений из функции Python?

В Питоне позволительно возвращать из функции несколько объектов, перечислив их через запятую после команды `return`:

5. Какие существуют способы передачи значений в функцию?

Существует четыре спецификации передачи параметров функции:

вызов функции с передачей значений

вызов функции с передачей адресов переменных

вызов функции с использованием механизма ссылок при передаче параметров

посредством глобальных параметров

6. Как задать значение аргументов функции по умолчанию?

Чтобы присвоить параметру значение по умолчанию, просто используйте оператор присваивания внутри определения функции.

7. Каково назначение `lambda`-выражений в языке Python?

Лямбда-выражения на Python - конструкторы простых безымянных однострочных функций. Бывают полезны (или нет). Вы можете использовать их для создания замыканий там, где не хотите использовать для создания замыкания именованную функцию.

8. Как осуществляется документирование кода согласно PEP257?

Цель этого PEP - стандартизировать структуру строк документации: что они должны в себя включать, и как это написать (не касаясь вопроса синтаксиса строк документации). Этот PEP описывает соглашения, а не правила или синтаксис.

9. В чем особенность однострочных и многострочных форм строк документации?

Особенностью однострочного текста является то, что после нажатия клавиши. Enter в поле ввода текста, при переходе на новую строку, создается новый текстовый объект. Проще говоря: разные строки текста будут независимыми объектами, которые можно перемещать по своему усмотрению.

**Вывод:** в ходе работы приобрел навыки по работе с функциями при написании программ с помощью языка программирования Python версии 3.x.