

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ ФГАОУ ВО  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

Отчет по лабораторной работе №8

По дисциплине основы кроссплатформенного программирования

«Работа с функциями в языке Python»

Выполнила:

студентк группы ИТС-б-о-21-1

Абдикодиров Жахонгир Хуснитдин  
угли

---

(подпись)

Проверил: Доцент, к.т.н, доцент  
кафедры

инфокоммуникаций

Воронкин Р. А.

Работа защищена с оценкой:

---

(подпись)

Ставрополь, 2022

## Написанный код для 1го задания

```
8LR > ex_1.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import numbers
5
6  if __name__ == '__main__':
7      |
8      def test():
9          |   number = int(input('введите число: '))
10         |   if number > 0:
11         |       |   positive()
12         |       elif number < 0:
13         |           |   negative()
14         |       else:
15         |           |   print("я вас не совсем понял. ;)")
16         |           |   test()
17     def positive():
18         |   print('положительное')
19     def negative():
20         |   print('отрицательное')
21     test()
```

## Написанный код для 2го задания

```
8LR > ex_2.py > test
You, 46 секунд назад | 1 author (You)
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  if __name__ == '__main__':
7      |   def test():
8      |       |   answer = 1
9      |       |   while 1:
10      |       |       |   num = int(input())
11      |       |       |   if not num: break
12      |       |       |   answer *= num
13      |       |   return(answer)
14      |   print(test())
```

### Написанный код для 3го задания

```
8LR > ex_3.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  if __name__ == '__main__':
7      def getInput(): return(input('запрашиваю ввод: '))
8      def testInput(n):
9          try: n = int(n); return(True)
10         except: return(False)
11     def strToInt(n): return(int(n))
12     def printInt(n): print(n)
```

### Написанный код для индивидуального задания

```
8LR > individ.py > add
You, 21 секунду назад | 1 author (You)
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  if __name__ == '__main__':
7      # Функции
8      def add():
9          name = input("Название пункта назначения рейса ")
10         nameShop = input("Номер рейса ")
11         cost = input("Начального пункта ")
12
13         # Создать словарь.
14         product = {
15             'name': name,
16             'nameShop': nameShop,
17             'cost': cost,
18         }
19
20         # Добавить словарь в список.
21         products.append(product)
22         # Отсортировать список в случае необходимости.
23         if len(products) >> 1:
24             products.sort(key=lambda item: item.get('name', ''))
25
```

## **Ответы на вопросы:**

1. Каково назначение функций в языке программирования Python?

Ответ: Функция в программировании представляет собой обособленный участок кода, который можно вызывать, обратившись к нему по имени, которым он был назван. При вызове происходит выполнение команд тела функции.

Функции можно сравнить с небольшими программками, которые сами по себе, т. е. автономно, не исполняются, а встраиваются в обычную программу.

2. Каково назначение операторов `def` и `return` ?

Ответ: Оператор `def`, выполняемый внутри определения функции, определяет локальную функцию, которая может быть возвращена или передана. Свободные переменные, используемые во вложенной функции, могут обращаться к локальным переменным функции, содержащей `def`.

Оператор `return` возвращает значение из функции. `return` без аргумента возвращает `None`. Функции, у которых `return` не определен, также возвращает `None`.

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

Ответ: В Python переменная, объявленная вне функции или в глобальной области видимости, называется глобальной переменной. К глобальной переменной можно получить доступ как внутри, так и вне функции.

Переменная, объявленная внутри тела функции или в локальной области видимости, называется локальной переменной.

4. Как вернуть несколько значений из функции Python?

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

5. Какие существуют способы передачи значений в функцию?

Ответ: По умолчанию аргументы могут передаваться в функцию Python либо по положению, либо явно по ключевому слову. Для производительности

и удобочитаемости имеет смысл ограничить способ передачи аргументов. где символы / и \* являются НЕ обязательными. Эти символы указывают тип аргумента в зависимости от того, как они могут быть переданы в функцию: только по позиции, по позиции или по ключевому слову только по ключевому слову

#### 6. Как задать значение аргументов функции по умолчанию?

Ответ: Значения параметров по умолчанию создаются при определении функции, а НЕ каждый раз, когда она вызывается в коде программы. Это означает, что эти выражение вычисляется один раз, и что для каждого вызова используется одно и то же предварительно вычисленное значение. Если функция изменяет объект (например, путем добавления элемента в список, словарь), значение по умолчанию фактически изменяется.

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

Ответ: Python поддерживает интересный синтаксис, позволяющий определять небольшие однострочные функции на лету. Позаимствованные из Lisp, так называемые lambda-функции могут быть использованы везде, где требуется функция. lambda – это выражение, а не инструкция. По этой причине ключевое слово lambda может появляться там, где синтаксис языка Python не позволяет использовать инструкцию def , – внутри литералов или в вызовах функций, например.

#### 8. Как осуществляется документирование кода согласно PEP257?

Ответ: PEP 257 описывает соглашения, связанные со строками документации python, рассказывает о том, как нужно документировать python код. 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. При нарушении этих соглашений, самое худшее, чего можно ожидать – некоторых неодобрительных взглядов. Но некоторые программы (например, docutils), знают о соглашениях, поэтому следование им даст вам лучшие результаты. Строки документации - строковые литералы,

которые являются первым оператором в модуле, функции, классе или определении метода.

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

Ответ: Однострочные и Многострочные

**Вывод:** приобрел навык по работе с функциями при написании программ с помощью языка программирования Python версии 3.x.