

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

**Отчет по лабораторной работе №3**

**Работа с функциями в языке Python**

**По дисциплине «Теории программирования и алгоритмизации»**

Выполнил студент группы ИВТ-б-о-20-1

Плотников Д. В. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил Воронкин Р. А. \_\_\_\_\_

(подпись)

Ставрополь 2021

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

### Ход работы

1. Создал новый собственный репозиторий. Ссылка на репозиторий: [https://github.com/Dmitry-15/10\\_laba.git](https://github.com/Dmitry-15/10_laba.git).
2. С помощью команды `git clone` клонировал удаленный репозиторий на свой ПК. Дополнил файл `.gitignore` необходимыми правилами для работы с IDE PyCharm.
3. Проработал примеры лабораторной работы.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import sys
from datetime import date

def get_worker():
    """
    Запросить данные о работнике.
    """
    name = input("Фамилия и инициалы? ")
    post = input("Должность? ")
    year = int(input("Год поступления? "))
    # Создать словарь.
    return {
        'name': name,
        'post': post,
        'year': year,
    }

def display_workers(workers):
    """
    Отобразить список работников.
    """
    # Проверить, что список работников не пуст.
    if workers:
        # Заголовок таблицы.
        line = '+--{}--{}--{}--{}--+'.format(
            '-' * 4,
            '-' * 30,
            '-' * 20,
```

Рисунок 1. Код примера

```

C:\ProgramData\Anaconda3\python.exe C:/Users/Дмитрий/8_laba/primer1.py
>>> add
Фамилия и инициалы? Плотников Д. В.
Должность? Директор
Год поступления? 2018
>>> select 2
      1: Плотников Д. В.
>>> list
+-----+-----+-----+-----+
| № | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Плотников Д. В. | Директор | 2018 |
+-----+-----+-----+-----+

```

Рисунок 2. Выполнение примера

### Задание 1

1. Условие задачи: 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".

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def test():
    message = int(input('Введите целое число:'))
    if message > 0:
        positive(message)
    else:
        negative(message)

def negative(message):
    print(f'Число {message} отрицательное')

def positive(message):
    print(f'Число {message} положительное")

if __name__ == '__main__':
    test()

```

Рисунок 3. Код задания 1

```
C:\ProgramData\Anaconda3\python.exe C:/Users/Дмитрий/10_laba/zadanie1.py
Введите целое число: 10
Число 10 положительное

Process finished with exit code 0
```

Рисунок 4. Выполнение задания 1

## Задание 2

1. Условие задачи: в основной ветке программы вызывается функция *cylinder()*, которая вычисляет площадь цилиндра. В теле *cylinder()* определена функция *circle()*, вычисляющая площадь круга по формуле  $\pi r^2$ . В теле *cylinder()* у пользователя спрашивается, хочет ли он получить только площадь боковой поверхности цилиндра, которая вычисляется по формуле  $2\pi rh$ , или полную площадь цилиндра. В последнем случае к площади боковой поверхности цилиндра должен добавляться удвоенный результат вычислений функции *circle()*.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import math

def cylinder():
    def circle():
        return math.pi * r ** 2

    r = float(input("Введите радиус: "))
    h = float(input("Введите высоту: "))
    answer = input("Хотите получить 1) площадь боковой поверхности цилиндра,"
                  " или 2) полную площадь цилиндра? - ")
    side = 2 * math.pi * r * h

    if answer == "1":
        print(f"Площадь боковой поверхности: {side}")
    else:
        s_circle = circle()
        full = side + 2 * s_circle
        print(f"Полная площадь: {full}")

if __name__ == '__main__':
    cylinder()
```

Рисунок 5. Код задания 2

```
C:\ProgramData\Anaconda3\python.exe C:/Users/Дмитрий/10_laba/zadanie2.py
Введите радиус: 5
Введите высоту: 6
Хотите получить 1) площадь боковой поверхности цилиндра, или 2) полную площадь цилиндра? - 2
Полная площадь: 345.5751918948772

Process finished with exit code 0
```

Рисунок 6. Выполнение задания 2

### Задание 3

1. Условие задачи: 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.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def main():
    num = 1
    while True:
        message = int(input("Введите число: "))
        num *= message
        if num == 0:
            print("Произведение равно 0")
            break
        else:
            print(f"Произведение равно: {num}")

if __name__ == '__main__':
    main()
```

Рисунок 7. Код задания 3

```
C:\ProgramData\Anaconda3\python.exe C:/Users/Дмитрий/10_laba/zadanie3.py
Введите число: 5
Произведение равно: 5
Введите число: 6
Произведение равно: 30
Введите число: 0
Произведение равно 0

Process finished with exit code 0
```

Рисунок 8. Выполнение задания 3

1. Условие задачи: напишите программу, в которой определены следующие четыре функции:

- Функция `get_input()` не имеет параметров, запрашивает ввод с клавиатуры и возвращает в основную программу полученную строку.
- 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*.
- Функция `str_to_int()` имеет один параметр. В теле преобразовывает переданное значение к целочисленному типу. Возвращает полученное число.
- Функция `print_int()` имеет один параметр. Она выводит переданное значение на экран и ничего не возвращает.

В основной ветке программы вызовите первую функцию. То, что она вернула, передайте во вторую функцию. Если вторая функция вернула *True*, то те же данные (из первой функции) передайте в третью функцию, а возвращенное третьей функцией значение – в четвертую.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def test_input(message):
    try:
        str_to_int(message)
    except ValueError:
        print("Невозможно преобразовать в число")

def str_to_int(message):
    a = int(message)
    print_int(a)

def print_int(a):
    print(a)

def get_input():
    message = input("Введите строку: ")
    test_input(message)

if __name__ == '__main__':
    while True:
        get_input()
```

Рисунок 9. Код задания 4



```

C:\ProgramData\Anaconda3\python.exe C:/Users/Дмитрий/Pictures/8_laba/individ.py
>>> add
Фамилия и инициалы? Плотников Д.В.
Знак Зодиака? лев
Дата рождения? 15 8 2002
>>> add
Фамилия и инициалы? Бобров Н.В.
Знак Зодиака? стрелец
Дата рождения? 19 12 2002
>>> list
+-----+-----+-----+-----+
| № | Ф.И.О. | Знак Зодиака | Дата рождения |
+-----+-----+-----+-----+
| 1 | Плотников Д.В. | лев | 15 8 2002 |
| 2 | Бобров Н.В. | стрелец | 19 12 2002 |
+-----+-----+-----+-----+
>>> whois
Кого ищем?: Плотников Д.В.
+-----+-----+-----+-----+
| № | Ф.И.О. | Знак Зодиака | Дата рождения |
+-----+-----+-----+-----+
| 1 | Плотников Д.В. | лев | 15 8 2002 |
+-----+-----+-----+-----+
>>> whois
Кого ищем?: Галакс
+-----+-----+-----+-----+
| № | Ф.И.О. | Знак Зодиака | Дата рождения |
+-----+-----+-----+-----+
Никто не найден

```

Рисунок 8. Выполнение индивид. задания

## Контрольные 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.

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

`def` – создаёт функцию, `return` – возвращает параметр из функции

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

К глобальной переменной можно обратиться из локальной области видимости. К локальной переменной нельзя обратиться из глобальной области видимости, потому что локальная переменная существует только в момент выполнения тела функции.

4. Как вернуть несколько значений из функции Python?



Перечислить все значения через запятую в return.

5. Какие существуют способы передачи значений в функцию?

Любая функция может обратиться к глобальной переменной. А также в функцию можно передать значение при вызове: `function(значение)`

6. Как задать значение аргументов функции по умолчанию?

При определении функции, в скобках указать переменные и их значения:  
`function(параметр=значение)`

7. Каково назначение lambda-выражений в языке Python?

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?

PEP 257 описывает соглашения, связанные со строками документации python, рассказывает о том, как нужно документировать python код.

Строки документации - строковые литералы, которые являются первым оператором в модуле, функции, классе или определении метода. Такая строка документации становится специальным атрибутом `__doc__` этого объекта.

Все модули должны, как правило, иметь строки документации, и все функции и классы, экспортируемые модулем также должны иметь строки документации. Публичные методы (в том числе `__init__`) также должны иметь строки документации. Пакет модулей может быть документирован в `__init__.py`.

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

Однострочная строка документации не должна быть "подписью" параметров функции / метода (которые могут быть получены с помощью интроспекции).

Многострочные строки документации состоят из однострочной строки документации с последующей пустой строкой, а затем более подробным описанием. Первая строка может быть использована автоматическими средствами индексации, поэтому важно, чтобы она находилась на одной строке и была отделена от остальной документации пустой строкой. Первая строка может быть на той же строке, где и открывающие кавычки, или на следующей строке. Вся документация должна иметь такой же отступ, как кавычки на первой строке

**Вывод:** в ходе выполнения лабораторной работы успешно приобрел навыки по работе с функциями при написании программ с помощью языка программирования Python3.