

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

**Кафедра инфокоммуникаций**

**Отчет по лабораторной работе №8**

**Работа с функциями в языке Python.**

**по дисциплине «Технологии программирования и алгоритмизации»**

Выполнил студент группы ИВТ-б-о-21-1

Лысенко И.А. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

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

Проверил Воронкин Р.А. \_\_\_\_\_

(подпись)

Ставрополь 2022

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

**Ход работы:**

1. Решил 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  #!/usr/bin/env python3
2  # -*- coding^ utf-8 -*-
3
4
5  def te():
6      number = int(input("Введите число: "))
7      if number > 0:
8          positive()
9      else:
10         negative()
11
12
13 def positive():
14     print("Положительное")
15
16
17 def negative():
18     print("Отрицательное")
19
20
21 if __name__ == '__main__':
22     te()
```

Run: Task 1 (1) x

C:\Users\user\AppData\Local\Programs\Python\

Введите число: 5

Положительное

Process finished with exit code 0

Рисунок 1 – Решение задачи

2. Решил задачу: в основной ветке программы вызывается функция `cylinder()`, которая вычисляет площадь цилиндра. В теле `cylinder()` определена функция `circle()`, вычисляющая площадь круга по формуле. В теле `cylinder()` у

пользователя спрашивается, хочет ли он получить только площадь боковой поверхности цилиндра, которая вычисляется по формуле, или полную площадь цилиндра. В последнем случае к площади боковой поверхности цилиндра должен добавляться удвоенный результат вычислений функции `circle()`.

```
1  #!/usr/bin/env python3
2  # -*- coding^ utf-8 -*-
3
4
5  def cylinder(r, h):
6      from math import pi
7      def circle(r): return pi * r ** 2
8
9      s = 2 * pi * r * h
10     if input('Хотите получить полную площадь? Если откажетесь, \
11     то будет выведена площадь боковой поверхности цилиндра. [y/n]: ') == 'y':
12         s += 2 * circle(r)
13     return s
14
15
16 if __name__ == '__main__':
17     r = int(input("Введите радиус: "))
18     h = int(input("Введите высоту: "))
19     print('s =', cylinder(r, h))
20
```

Run: Task 2 (1) ×

C:\Users\user\AppData\Local\Programs\Python\Python39\python.exe "C:\Users\user\..."

Введите радиус: 4

Введите высоту: 9

Хотите получить полную площадь? Если откажетесь, то будет выведена площадь

s = 326.7256359733385

Рисунок 2 – Решение задачи

3. Решил 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.

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

IsSveshuD
def t():
    r = 1
    while 1:
        ch = int(input())
        if not ch: break
        r *= ch
        print(r)
    return (r)

if __name__ == '__main__':
    print(t())
```

Task 3 x

C:\Users\user\AppData\Local

4

4

5

20

0

20

Рисунок 3 – Решение задачи

4. Решил задачу: 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.

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

IsSveshuD
def get_input():
    return input('Введите число: ')

IsSveshuD
def tes_input(n):
    try:
        n = int(n)
        return True
    except:
        return False

IsSveshuD
def str_to_int(n):
    return int(n)

IsSveshuD
def print_int(n):
    print(n)

out() > try
Task 4 x
Введите число: 4
True
4
4
None
```

Рисунок 4 – Решение задачи

## 5. Выполнил индивидуальное задание.

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

import sys

def get_contact()::
    """
    Запросить данные о человеке
    """
    family = input("Фамилия? ")
    name = input("Имя? ")
    number = int(input("Номер телефона? "))
    born = list(map(int, input("Дата рождения? ").split('.', 2)))

    return {
        'family': family,
        'name': name,
        'number': number,
        'born': born,
    }

def display_contact(contacts)::
    """
    Отобразить список контактов
    """
    if contacts:
        line = '+-{}-+-{}-+-{}-+-{}-+'.format(
            '-' * 4,
            '-' * 30,
            '-' * 20,
            '-' * 30,
            '-' * 20
        )
        print(line)
        print(
            '| {:^4} | {:^30} | {:^20} | {:^30} | {:^20} |'.format(
                "№",
                "Фамилия",
                "Имя",
                "Номер телефона",
                "Дата Рождения"
            )
        )
        print(line)

        for idx, contact in enumerate(contacts, 1):
            print(
                '| {:>4} | {:<30} | {:<20} | {:<30} | {:>20} |'.format(
                    idx,
                    contact.get('family', ''),
                    contact.get('name', ''),
                    contact.get('number', 0),
                    '.'.join((str(i) for i in contact['born']))
                )
            )
        print(line)
```

```

else:
    print("Список контактов пуст.")

def select_contact(contacts, period):
    """
    Выбрать маршрут
    """
    result = []
    for contact in contacts:
        if contact.get('family') == period:
            result.append(contact)

    return result

def main():
    """
    Главная функция программы
    """
    contacts = []

    while True:
        command = input(">>> ").lower()
        if command == 'exit':
            break

        elif command == 'add':
            contact = get_contact()
            contacts.append(contact)
            if len(contacts) > 1:
                contacts.sort(key=lambda item: item.get('number', [0 - 2]))

        elif command == 'list':
            display_contact(contacts)

        elif command.startswith('select'):
            period = input('Введите фамилию человека, информацию по которому Вы хотите найти: ')

            selected = select_contact(contacts, period)
            display_contact(selected)

        elif command == 'help':
            print("Список команд:\n")
            print("add - Добавить контакт;")
            print("list - Вывести список контактов;")
            print("select - Поиск по фамилии;")
            print("help - Отобразить справку;")
            print("exit - Завершить работу с программой.")

        else:
            print(f"Неизвестная команда {command}", file=sys.stderr)

if __name__ == '__main__':
    main()

```

Рисунок 5 – Индивидуальное задание

## Ответы на вопросы:

1. Каково назначение функций в языке программирования Python? Внедрение функций позволяет решить проблему дублирования кода в разных местах программы.

2. Каково назначение операторов def и return?

Def сообщает интерпретатору, что перед ним определение функции.

Return осуществляет выход из функции и передачу данных в то место, откуда она была вызвана.

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

Локальные переменные видны только в локальной области видимости, которой может выступать отдельно взятая функция. Глобальные переменные видны во всей программе.

4. Как вернуть несколько значений из функции Python?

Перечислить их через запятую после команды return.

5. Какие существуют способы передачи значений в функцию?

Передача данных и передача ссылок на данные.

6. Как задать значение аргументов функции по умолчанию?

Вписать вместо неё в вызове функции другое значение.

7. Каково назначение lambda-выражений в языке Python?

Их используют для сокращения записи функций до одной строки.

8. Как осуществляется документирование кода согласно PEP257?

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?

Одиночные строки документации предназначены для действительно очевидных случаев. Этот тип строк документации подходит только для C функций (таких, как встроенные модули), где интроспекция не представляется возможной. Многострочные строки документации состоят из однострочной строки документации с последующей пустой строкой, а затем более подробным описанием.

**Вывод:** в ходе лабораторной работы были изучены все аспекты работы с функциями в языке Python.