МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное

образовательное учреждение высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2.8

Работа с функциями в языке Python.

по дисциплине «Технологии программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-20-1

Хашиев Х.М. « »_____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.

Ход работы: Пример 1

https://github.com/Mirror-Shard/L2.8

1. Создал репозиторий на github с лицензией MIT, добавил .gitignore и выбрал язык Python.

2. Проработал пример из учебника:

```python
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(staff):
    """
    Отобразить список работников.
    """

    # Проверить, что список работников не пуст.
    if staff:

        # Заголовок таблицы.
        line = '+-{}-+-{}-+-{}-+-{}-+'.format(
            '-' * 4,
            '-' * 30,
            '-' * 20,
            '-' * 8
        )
```

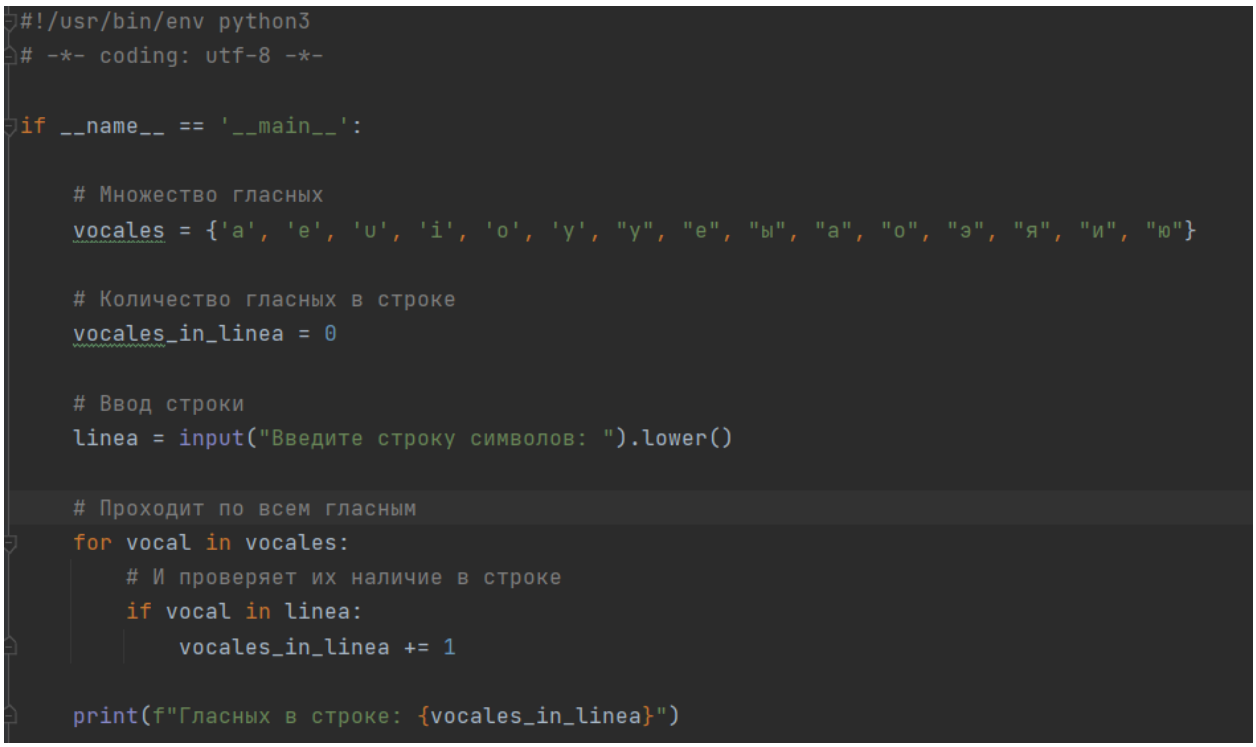
Рисунок 1 – Код примера(1)

```
        print(line)
        print(
            '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
                "No",
                "Ф.И.О.",
                "Должность",
                "Год"
            )
        )
        print(line)

        # Вывести данные о всех сотрудниках.
        for idx, worker in enumerate(staff, 1):
            print(
                '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
                    idx,
                    worker.get('name', ''),
                    worker.get('post', ''),
                    worker.get('year', 0)
                )
            )
        print(line)

    else:
        print("Список работников пуст.")


def select_workers(staff, period):
    """
    Выбрать работников с заданным стажем.
```

Рисунок 2 – Код примера(2)

3. Результат:

```
Фамилия и инициалы? Хашиев Х. М.
Должность? Президент
Год поступления? 2012
>>> list
+------+----------------------------------+----------------------+----------+
|  No  |             Ф.И.О.               |      Должность       |   Год    |
+------+----------------------------------+----------------------+----------+
|    1 | Хашиев Х. М.                     | Президент            |     2012 |
+------+----------------------------------+----------------------+----------+
```



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

if __name__ == '__main__':

    # Множество гласных
    vocales = {'a', 'e', 'u', 'i', 'o', 'y', "у", "е", "ы", "а", "о", "э", "я", "и", "ю"}

    # Количество гласных в строке
    vocales_in_linea = 0

    # Ввод строки
    linea = input("Введите строку символов: ").lower()

    # Проходит по всем гласным
    for vocal in vocales:
        # И проверяет их наличие в строке
        if vocal in linea:
            vocales_in_linea += 1

    print(f"Гласных в строке: {vocales_in_linea}")
```

Рисунок 3 – Результат работы примера

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. Код задания 1(порядок определения функций не имеет значения):

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

def positive():
    print("Число положительное")


def negative():
    print("Число отрицательное")


def test(a):
    if a >= 0:
        positive()
    else:
        negative()


if __name__ == '__main__':
    num = float(input("Введите число: "))
    test(num)
```

Рисунок 4 – Код задания 1

2. Результат работы:

```
Введите число: 5
Число положительное

Process finished with exit code 0
```

Рисунок 5 – Код второго задания

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 . 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. Код задания 2, часть первая:

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import sys
import math

def cylinder():

    # Вычисляет площадь круга
    def cylinder_circle(r):
        circle = math.pi * r ** 2
        return circle

    # Вычисляет боковую сторону цилиндра
    def cylinder_side(r, h):
        side = 2 * math.pi * r * h
        return side

    # Вычисляет площадь всего цилиндра
    def full_cylinder(r, h):
        full = cylinder_side(r, h) + cylinder_circle(r) * 2
        return full

    # Запрос радиуса и высоты
    radius = float(input("Введите радиус: "))
    height = float(input("Введите выстоу: "))

    # Запрос выбора
    print("Вычислить площадь всего цилиндра или только боковой стороны?")
    print("1 - всего цилиндра, 2 - только боковой стороны")
    choice = int(input())

```

Рисунок 6 – Код задания 2, часть первая

2. Кода задания 2, часть вторая

```
32
33      # Если выбор 1
34      if choice == 1:
35          print("Площадь всего цилиндра равна:")
36          print(full_cylinder(radius, height))
37
38      # Если выбор 2
39      elif choice == 2:
40          print("Площадь боковой стороны цилиндра равна:")
41          print(cylinder_side(radius, height))
42
43      # Недопустимый выбор
44      else:
45          print("Введено недопустимое значение", file=sys.stderr)
46          exit(1)
47
48
49  if __name__ == '__main__':
50      cylinder()
51
```

Рисунок 7 – Код задания 2, часть вторая

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.

1. Код задания 3

```python
# Перемножает пока не введётся ноль
def multiplier():

    # Запрос значений
    a = float(input("a = "))
    b = float(input("b = "))

    # Если оба значения числа - вычисляется произведение
    if isDigit(a) and isDigit(b):
        c = a * b
    # Иначе ошибка
    else:
        print("Введено недопустимое значение!", file=sys.stderr)

    # Если бы введён ноль, программа завершается
    if c == 0:
        print("Адиос!", file=sys.stderr)
        exit(1)
    # Если нет, возвращает произведение
    else:
        return c


def main():

    print("Вводите числа, они будут перемножаться, пока вы не введёте 0")

    # Бесконечный запрос функции перемножения
    while True:
        print(multiplier())
```

Рисунок 8 – Код задания 3(кроме вызова главной функции main)

2. Результат выполнения:

```
a = 8
b = 8
64.0
a = 0
b = 0
Адиос!
```

Рисунок 9 – Результат работы третьего задания

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.

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.

1. Код задания 4:

```python
def get_input():
    x = input("Введите что-нибудь: ")
    return x


def test_input(x):
    try:
        int(x)
        return True
    except ValueError:
        return False


def str_to_int(x):
    int(x)
    return x


def print_int(x):
    print(x)


if __name__ == '__main__':

    a = get_input()

    if test_input(a):
        b = str_to_int(a)
        print_int(b)
```

Рисунок 10 – Код задания 4

2. Результат работы четвёртого задания:

```
Введите что-нибудь: 54
54
```

Рисунок 11 – Результат работы задания 4

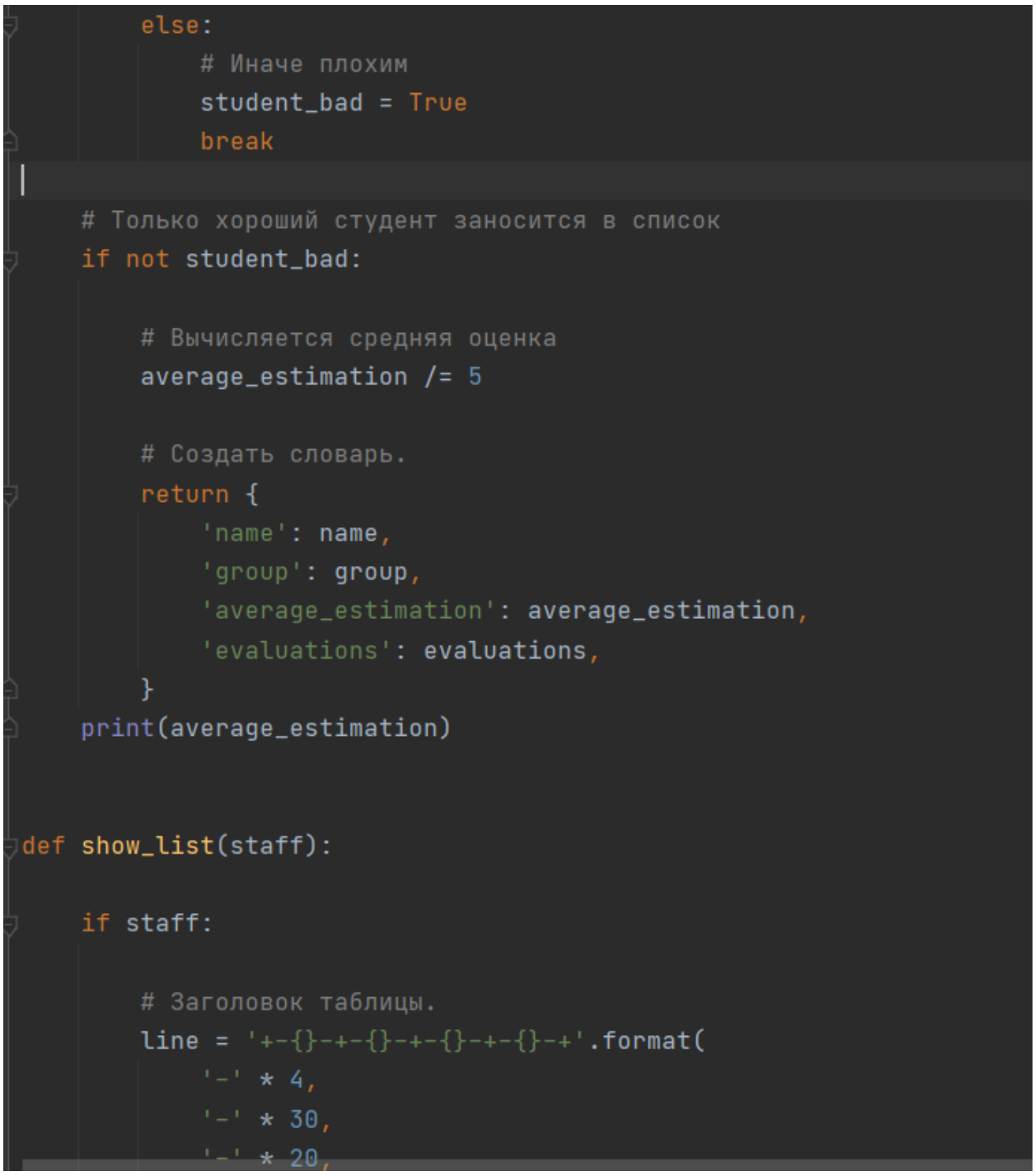
Индивидуальное задание

Решить индивидуальное задание лабораторной работы 2.6, оформив каждую команду в виде отдельной функции.

1. Переделал все команды из задания в отдельные функции:

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


# Создаёт словарь - студент и возвращает его
def get_student():

    # Переменная для студента с оценкой хуже 4
    student_bad = False

    # Все оценки
    evaluations = []

    # Средняя оценка
    average_estimation = 0

    # Запросить данные о студенте.
    name = input("Фамилия и инициалы: ")
    group = input("Номер группы: ")

    # Ввод 5-ти оценок
    print("Введите 5 оценок через Enter:")
    for i in range(5):
        estimation = int(input())
        evaluations.append(estimation)

    # Проходит по оценкам
    for i, x in enumerate(evaluations):
        # Если оценки только 4 и 5, то он считается хорошим
        if evaluations[i] == 4 or evaluations[i] == 5:
            average_estimation += evaluations[i]
```

Рисунок 12 – Код индивидуального задания, часть 1



```
        else:
            # Иначе плохим
            student_bad = True
            break

    # Только хороший студент заносится в список
    if not student_bad:

        # Вычисляется средняя оценка
        average_estimation /= 5

        # Создать словарь.
        return {
            'name': name,
            'group': group,
            'average_estimation': average_estimation,
            'evaluations': evaluations,
        }
    print(average_estimation)


def show_list(staff):

    if staff:

        # Заголовок таблицы.
        line = '+-{}-+-{}-+-{}-+-{}-+'.format(
            '-' * 4,
            '-' * 30,
            '-' * 20,
```

Рисунок 13 – Код индивидуального задания, часть 2

Рисунок 14 – Код индивидуального задания, часть 3

```
        print("Список пуст")


def show_help():
    # Вывести справку о работе с программой.
    print("Список команд:\n")
    print("add - добавить студента;")
    print("list - вывести список студентов;")
    print("help - отобразить справку;")
    print("exit - завершить работу с программой.")


if __name__ == '__main__':
    """
    Главная функция
    """

    # Список студентов.
    students = []

    # Организовать бесконечный цикл запроса команд.
    while True:

        # Запросить команду из терминала.
        command = input(">>> ").lower()

        # Выполнить действие в соответствие с командой.
        if command == 'exit':
            break
```

Рисунок 15 – Код индивидуального задания, часть 4

```python
        # Студент адд ##################################
        if command == 'add':

            student = get_student()

            # Добавить словарь в список.
            students.append(student)

            # Отсортировать список в случае необходимости.
            if len(students) > 1:
                students.sort(key=lambda item: item['average_estimation'], reverse=True)

        # Лист #########################################
        elif command == 'list':
            show_list(students)

        elif command == 'help':
            show_help()

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

Рисунок 16 – Код индивидуального задания, часть 5

3. Результат работы индивидуального задания:

```
>>> add
Фамилия и инициалы: Хашиев Х. М.
Номер группы: 20
Введите 5 оценок через Enter:
5
5
5
5
5
>>> list
+------+--------------------------------+----------------------+----------+
|  No  |             Ф.И.О.             |        Группа        | Средняя оценка |
+------+--------------------------------+----------------------+----------+
|    1 | Хашиев Х. М.                   | 20                   |      5.0 |
+------+--------------------------------+----------------------+----------+
```

Рисунок 17 – Результат работы

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?

Просто перечислить их через запятую в return

5. Какие существуют способы передачи значений в функцию?

1)Любая функция может обратиться к глобальной переменной. 2)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.

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