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 №11
дисциплины «Основы программной инженерии»

Выполнил:
Плугатырев Владислав Алексеевич
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка и сопровождение программного обеспечения», очная форма обучения

_____
(подпись)

Доцент кафедры инфокоммуникаций
Воронкин Роман Александрович

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 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.

Порядок выполнения работы

1. Создал репозиторий GitHub.

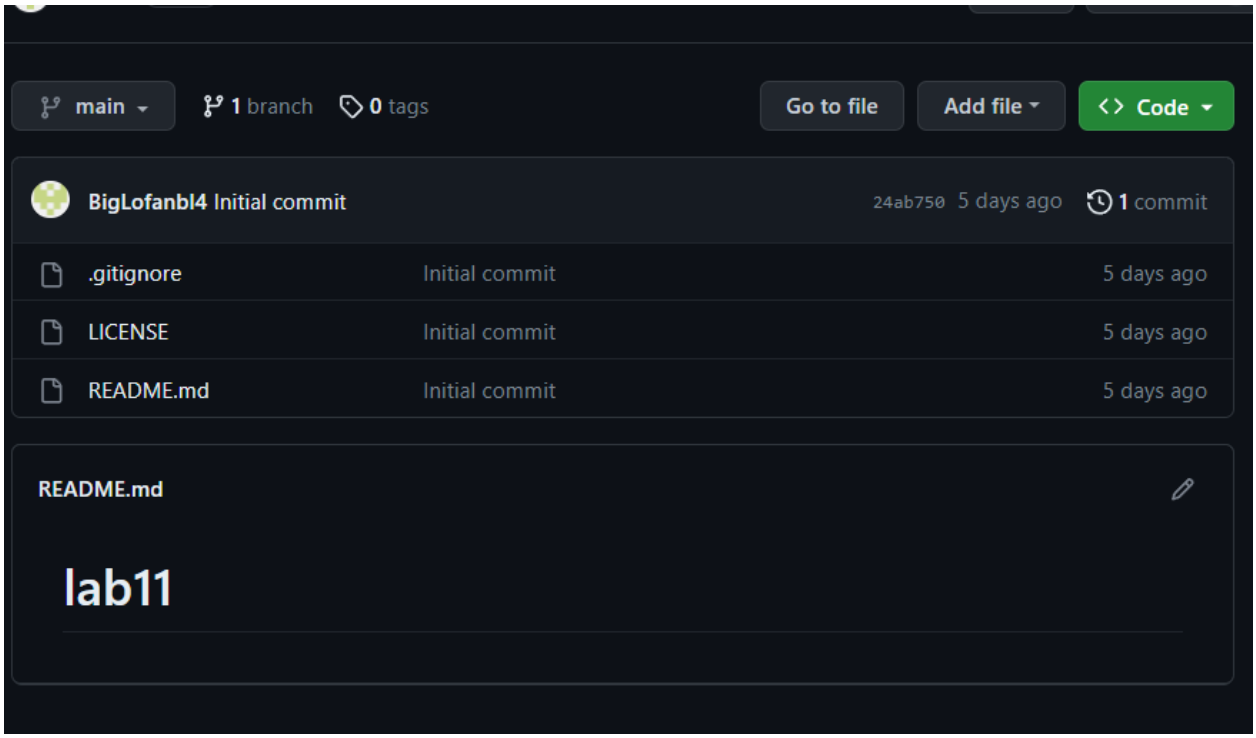

Рисунок 1.1 – Созданный репозиторий

2. Проработал примеры из лабораторной работы.

```
>>> add
Фамилия и инициалы? Плугатырев В.А.
Должность? Дурачок
Год поступления? 2022
>>> list
+------+----------------------------------+----------------------+----------+
|  №   |              Ф.И.О.              |      Должность       |   Год    |
+------+----------------------------------+----------------------+----------+
|    1 | Плугатырев В.А.                  | Дурачок              |     2022 |
+------+----------------------------------+----------------------+----------+
```

Рисунок 2.1 – Результат вывода программы из примера

3. 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".

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


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


def positive():
    print("Положительное")


def negative():
    print("Отрицательное")


if __name__ == "__main__":
    test()
```

Рисунок 3.1 – Код программы

```
Введите целое число: 12
Положительное
PS C:\Users\vladi\OneDrive\Рабо
pData/Local/Programs/Python/Pyt
раммной инженерии/11/lab11/ex1.
Введите целое число: -2
Отрицательное
```

Рисунок 3.2 – Вывод программы

Имеет ли значение порядок определения самих функций? То есть должны ли определения positive() и negative() предшествовать test() или могут следовать после него?

Ответ: в данном случае разницы нет, т.к. определение функций предшествует их вызовам, в противном случае (если бы определение функций positive и negative было позже, чем вызов функции test) возникнет ошибка.

4. 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().

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

from math import pi


def cylinder(rad, h):
    circle = lambda rad: pi * rad**2
    command = input("Найти площадь боковой поверхности или полную? (боковой/полную) ").lower()

    match command:
        case "боковой":
            return 2 * pi * rad * h
        case "полную":
            return 2 * circle(rad) + 2 * pi * rad * h


if __name__ == "__main__":
    print(cylinder(float(input("Введите радиус: ")), float(input("Введите высоту: "))))
```

Рисунок 4.1 – Код программы

```
Введите радиус: 12.4
Введите высоту: 2.4
Найти площадь боковой поверхности или полную? (боковой/полную) боковой
186.98759474166448
PS C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\11\lab
vladi/OneDrive/Рабочий стол/Основы программной инженерии/11/lab11/ex2.py"
Введите радиус: 14.4
Введите высоту: 2.8
Найти площадь боковой поверхности или полную? (боковой/полную) полную
1556.21933688224
```

Рисунок 4.2 – Вывод программы

5. 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.

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


def multiply():
    num = int(input("Введите число не равное 0: "))
    res = 1

    while num != 0:
        res *= num
        num = int(input("Введите число или 0, чтобы закончить: "))
    return res


if __name__ == "__main__":
    print(multiply())
```

Рисунок 5.1 – Код программы

```
Введите число не равное 0: 3
Введите число или 0, чтобы закончить: 4
Введите число или 0, чтобы закончить: 12
Введите число или 0, чтобы закончить: 2
Введите число или 0, чтобы закончить: 4
Введите число или 0, чтобы закончить: 0
1152
```

Рисунок 5.2 – Вывод программы

6. 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.

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


def get_input():
    return input()


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


def str_to_int(string):
    return int(string)


def print_int(num):
    print(num)


if __name__ == "__main__":
    inp = get_input()
    if test_input(inp):
        print_int(str_to_int(inp))
    else:
        print("Incorrect input")
```

Рисунок 6.1 – Код программы

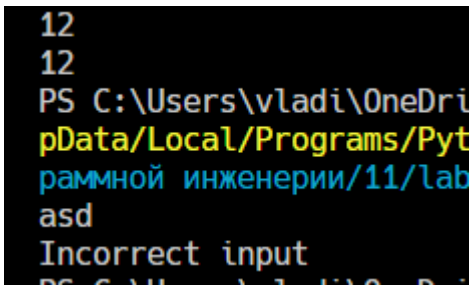

Рисунок 6.2 – Вывод программы

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

Листинг программы:

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

import sys
from datetime import datetime

def get_person():
    """
    Запросить данные о человеке.
    """
    person = {}
    person["surname"] = input("Введите фамилию: ")
    person["name"] = input("Введите имя: ")
    person["zodiac"] = input("Введите знак зодиака: ")
    person["birthday"] = input("Дата рождения (число.месяц.год):").split(".")
    return person

def display_people(people):
    """
    Отобразить список людей.
    """
    if people:
        line = "+-{}-+-{}-+-{}-+-{}-+-{}-+".format(
            "-" * 4, "-" * 30, "-" * 30, "-" * 20, "-" * 20
        )
        print(line)
        print(
            "| {:^4} | {:^30} | {:^30} | {:^20} | {:^20} |".format(
                "№", "Фамилия", "Имя", "Знак зодиака", "Дата рождения"
            )
        )
        print(line)

        for idx, person in enumerate(people, 1):
            print(
                "| {:>4} | {:<30} | {:<30} | {:<20} | {:>20} |".format(
                    idx,
                    person.get("surname", ""),
```

```
                person.get("name", ""),
                person.get("zodiac", ""),
                ".".join(person.get("birthday", "")),
            )
        )
    print(line)
  else:
    print("Список пуст")

def select_people(surname, people):
  """
  Выбрать людей с заданной фамилией.
  """
  result = []
  for i in people:
    if i.get("surname", "") == surname:
      result.append(i)
  return result

def get_instructions():
  print("add - добавление нового человека")
  print("info - данные о человеке по его фамилии")
  print("exti - завершение программы")
  print("list - вывод информации о всех людях")

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

  while True:
    command = input("Введите команду (add, info, list, exit, help): ").strip().lower()

    match command:
      case "exit":
        break

      case "add":
        person = get_person()
        people.append(person)
        people.sort(
          key=lambda x: datetime.strptime(".".join(x["birthday"]), "%d.%m.%Y")
        )

      case "info":
        surname = input("Введите фамилию: ")
        selected = select_people(surname, people)
        display_people(selected)

      case "list":
        display_people(people)

      case "help":
```

```
            get_instructions()

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

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

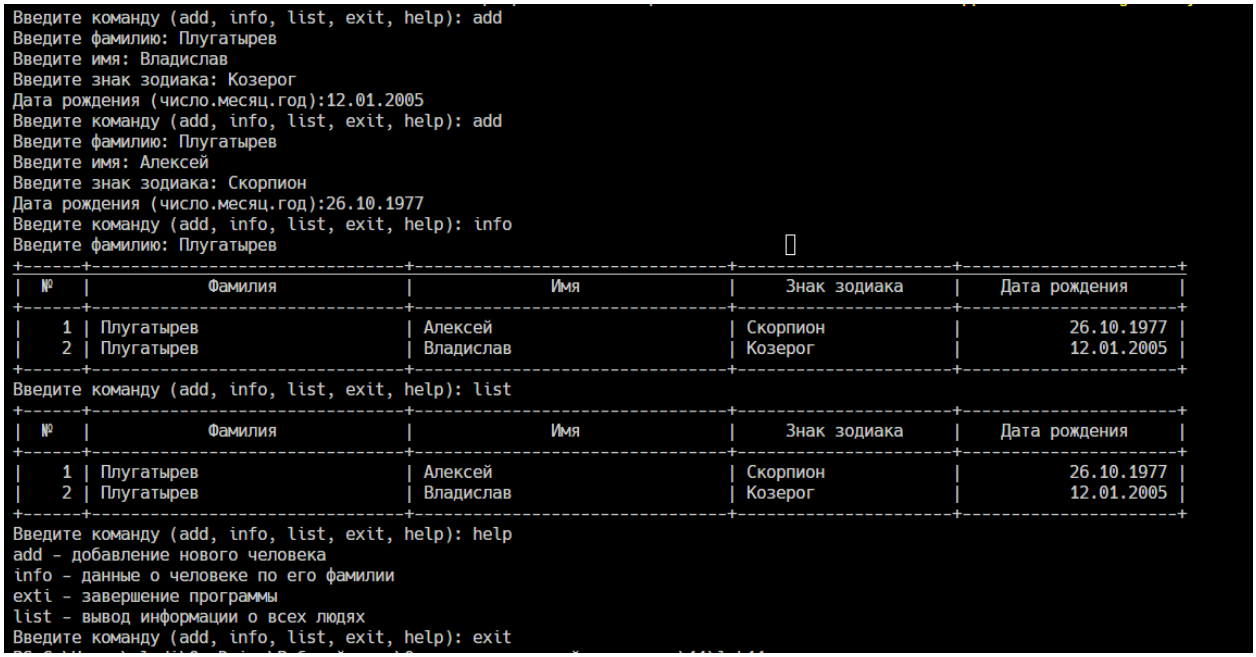

Рисунок 7.1 – Вывод программы

Ответы на контрольные вопросы

1. Функции в Python - это блоки кода, которые могут принимать данные в качестве входных параметров, выполнять определенные действия и возвращать результат. Они позволяют определять и повторно использовать определенную функциональность в компактной форме.

2. Оператор def используется для определения функции и ее параметров. После def следует имя функции, а затем в круглых скобках указываются параметры функции. Тело функции начинается с отступа и содержит инструкции, которые выполняются при вызове функции. Оператор return используется для возврата значения из функции.

3. Локальные переменные - это переменные, которые определены внутри функции и доступны только внутри этой функции. Глобальные

переменные - это переменные, которые определены вне функции и доступны в любой части программы.

4. Чтобы вернуть несколько значений из функции Python, можно использовать кортеж, список или словарь.

5. Существуют различные способы передачи значений в функцию, такие как передача по позиции, передача по имени и передача по значению.

6. Значения аргументов функции могут быть заданы по умолчанию, используя оператор =.

7. Lambda-выражения - это анонимные функции, которые могут содержать только одно выражение. Они определяются с помощью ключевого слова lambda.

8. Документирование кода в Python осуществляется согласно PEP257.

9. Однострочные строки документации начинаются с символа #, а многострочные строки документации заключаются в тройные кавычки.