

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 алгоритмизация»

Выполнил

студент группы ИВТ-б-о-20-1

Дыбов Д.В. « » _____ 20__ г.

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 репозиторий для лабораторной работы №2.8;
2. Клонировал созданный репозиторий на компьютер;
3. Создал новый PyCharm проект в папке репозитория;
4. Проработал пример:
5. Проверил пример на работоспособность;

```
>>> add
Фамилия и инициалы? Сидоров И.В.
Должность? Стажёр
Год поступления? 1999
>>> list
```

№	Ф.И.О.	Должность	Год
1	Сидоров И.В.	Стажёр	1999

```
>>> select 22
```

№	Ф.И.О.	Должность	Год
1	Сидоров И.В.	Стажёр	1999

Рисунок 1 – Результат выполнения примера

6. Написал код для решения первой задачи;
7. Проверил код на работоспособность и получил результат:

```
Введите число 1
Положительное
Process finished with exit code 0
```

Рисунок 2 – Результат выполнения первой задачи

8. Написал код для решения второй задачи;
9. Проверил код на работоспособность и получил результат:

```
Радиус: 5
Высота: 6
Вывести площадь боковой поверхности цилиндра? [y/n]: y
s = 188.49555921538757

Process finished with exit code 0
```

Рисунок 3 – Результат выполнения второй задачи

10. Написал код для решения третьей задачи;

11. Проверил код на работоспособность и получил результат:

```
1
2
3
4
5
6
7
8
9
0
362880
```

Рисунок 4 – Результат выполнения третьей задачи

12. Написал код для решения четвертой задачи;

13. Проверил код на работоспособность и получил результат:

```
Запрос ввода: 45
45

Process finished with exit code 0
```

Рисунок 5 – Результат выполнения четвертой задачи

14. Выполнил индивидуальное задание;

```

>>> add
Название пункта назначения рейса Москва
Номер рейса 2
Тип самолёта 1
>>> list
+-----+-----+-----+-----+
| № | Пункт назначения | Номер рейса | Тип самолёта |
+-----+-----+-----+-----+
| 1 | Москва | 2 | 1.0 |
+-----+-----+-----+-----+
>>> select Москва
1: Москва
Номер рейса 2
Тип самолёта: 1.0
>>>

```

Рисунок 6 – Результат выполнения индивидуального задания

14. С помощью сайта проверил пример на наличие ошибок;
15. Результат не выдал ошибок;

Python code

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 from datetime import date
5 import sys
6
7 # Список работников.
8 workers = []
9
10
11 def get_worker():
12     """
13     Запросить данные о работнике.
14     """

```

No syntax errors detected :)

Рисунок 7 – Проверка кода на наличие ошибок

16. Проверил все задачи на наличие ошибок;

Python code

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 def positive():
6     print("Положительное")
7
8
9 def negative():
10    print("Отрицательное")
11
12
13 def test():
14     pass
```

No syntax errors detected :)

Рисунок 8 – Проверка первой задачи на наличие ошибок

Python code

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import math
5
6
7 def cylinder():
8     from math import pi
9
10     def circle():
11         return math.pi * r ** 2
12
13     r = int(input("Радиус: "))
```

No syntax errors detected :)

Рисунок 9 – Проверка второй задачи на наличие ошибок

Python code

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 def test():
6     answer = 1
7     while 1:
8         num = int(input())
9         if not num:
10             break
11         answer *= num
12     return answer
13
14 ..
```

No syntax errors detected :)

Рисунок 10 – Проверка третьей задачи на наличие ошибок

Python code

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 def get_input():
6     return input('Запрос ввода: ')
7
8
9 def test_input(a):
10     if type(a) == int or type(a) == float:
11         return True
12     elif a.isnumeric():
13         return True
14     ,
```

No syntax errors detected :)

Рисунок 11 – Проверка четвёртой задачи на наличие ошибок

17. Проверил индивидуальное задание на наличие ошибок;

Python code

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 from datetime import date
5 import sys
6
7 airplanes = []
8
9
10 def get_airplane():
11
12     path = input("Название пункта назначения рейса ")
13     number = input("Номер рейса ")
14     # ...
```

No syntax errors detected :)

Рисунок 12 – Проверка индивидуального задания

18. Отправил все изменения на репозиторий;

Контрольные вопросы

1. Каково назначение функций в языке программирования Python?

Функция в языке python – это обособленный участок кода, который можно вызывать, обратившись к нему по имени, которым он был вызван

2. Каково назначение операторов def и return ?

Ключевое слово def сообщает интерпретатору, что перед ним определение функции.

Оператор return передаёт данные из тела, которого была вызвана в ветку программы.

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

Глобальные переменные нужны для того, чтобы была возможность обратиться к ним по имени и получить связанное с ними значение, глобальные переменные видны во всей программе.

Локальные переменные существуют только в момент выполнения, Поэтому к локальным переменным нельзя обращаться из глобальной области видимости.

4. Как вернуть несколько значений из функции Python?

Чтобы вернуть несколько значений из функции Python необходимо использовать команду `return` и написать функции, которым нужно вернуть значение через запятую.

5. Какие существуют способы передачи значений в функцию?

Для передачи значений в функцию с изменяемым типом данных необходимо использовать ссылки.

6. Как задать значение аргументов функции по умолчанию?

Чтобы задать значение аргументов функции по умолчанию, необходимо указать один из аргументов при вызове данной функции.

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

Ключевое слово `lambda` может появляться там, где нельзя использовать инструкцию `def`. Например: внутри литералов или в вызовах функций.

8. Как осуществляется документирование кода согласно PEP257?

Для документации кода в Python используют тройные кавычки.

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

Однострочные строки документации используются для описания очевидных случаев. Однострочные строки документации должны уместиться на одной строке.

Многострочные строки документации используются для более подробного описания. Многострочные строки документации начинаются с

тройных кавычек. 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 были приобретены навыки по работе с функциями при написании программ с помощью языка программирования Python версии 3.x.