

и 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 «Программирование на Python»
Вариант 31

Выполнил:

Репкин Александр Павлович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной
техники и автоматизированных
систем», очная форма обучения

(подпись)

Руководитель практики:

Воронкин Р.А., канд. техн. наук,
доцент кафедры инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____
Ставрополь, 2023 г.

Тема: Работа с функциями в языке Python.

Цель: приобрести навыки работы с функциями при написании программ с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

1. Выполнен первый пример. В нём создавался словарь, хранящий информацию о сотрудниках – ФИО, должность, год поступления на работу. Требовалось обрабатывать несколько команд при помощи функций – вывод всех сотрудников (list), вывод сводки о имеющихся командах (help), добавление сотрудника (add), вывод сотрудников относительно требуемой продолжительности работы (select “срок”), выход из программы (exit).

```
>>> list
Список работников пуст.
>>> add
Фамилия и инициалы? Волков Ж.Д.
Должность? Зоолог
Год поступления? 1978
>>> add
Фамилия и инициалы? Сладёнов В.Ф.
Должность? Кондитер
Год поступления? 2000
>>> add
Фамилия и инициалы? Глазкова Ю.А.
Должность? Офтальмолог
Год поступления? 1977
>>> list
```

№	Ф.И.О.	Должность	Год
1	Волков Ж.Д.	Зоолог	1978
2	Глазкова Ю.А.	Офтальмолог	1977
3	Сладёнов В.Ф.	Кондитер	2000

```
>>> select 45
```

№	Ф.И.О.	Должность	Год
1	Волков Ж.Д.	Зоолог	1978
2	Глазкова Ю.А.	Офтальмолог	1977

```
>>> select 46
```

№	Ф.И.О.	Должность	Год
1	Глазкова Ю.А.	Офтальмолог	1977

Рисунок 1. Полученный результат примера.

2. Выполнено задание №8, в котором требовалось создать три функции: test (относительно введенного пользователем числа вызывает positive или negative), positive (Выводит на экран сообщение, что введенное число положительное), negative (Выводит на экран сообщение, что введенное число отрицательно). При перемещении объявлений функций не возникало ошибок, так как Python исполняет функции только при их вызове. Поэтому, если функции определены до момента вызова в основном коде (за

исключением объявления после условия `if name == "main"`), то не возникает проблем, и программа выполняется корректно.

```
Good day! Please, enter your value - 3
It seems to us that value is positive!
```

Рисунок 2. Полученный результат задания №8 – после `test()`.

```
Good day! Please, enter your value - -5
It seems to us that value is negative!
```

Рисунок 3. Полученный результат задания №8 – до `test()`.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  def negative():
5      |   print("It seems to us that value is negative!")
6
7
8  def test():
9      |   """Определение числа, положительное или отрицательное."""
10
11      |   number = int(input("Good day! Please, enter your value - "))
12      |   if number >= 0:
13      |       |   positive()
14      |   else:
15      |       |   negative()
16
17
18  def positive():
19      |   print("It seems to us that value is positive!")
20
21
22  if __name__ == '__main__':
23      |   test()
24
```

Рисунок 4. Полученный код задания №8.

3. Выполнено задание №10, в котором требовалось создать функцию `circle` внутри функции `cylinder` и, относительно данных пользователя, найти площадь боковой или полной поверхности цилиндра.

```
Good day! If you want to get square of cylinder's border, print 1.
If you want to get square of the whole cylinder, print anything else - 2

Great! Now, please, give us the radius - 5.6
And height - 3.4

According to our calculations, the answer is - 316.6725394818511
```

Рисунок 5. Полученный результат задания №10 – площадь всей поверхности.

```
Good day! If you want to get square of cylinder's border, print 1.
If you want to get square of the whole cylinder, print anything else - 1

Great! Now, please, give us the radius - 5.6
And height - 3.4

According to our calculations, the answer is - 119.63184824869931
```

Рисунок 6. Полученный результат задания №10 – площадь боковой поверхности.

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  from math import pi # Весь math не нужен, его импорт уже будет антипаттерном
5
6
7  def cylinder():
8
9      def circle(radius):
10         """Определение площади круга"""
11
12         return pi * (radius ** 2)
13
14         variant = input(
15             '''Good day! If you want to get square of cylinder's border, print 1.
16             If you want to get square of the whole cylinder, print anything else - '''
17         )
18         radius = float(input("\nGreat! Now, please, give us the radius - "))
19         height = float(input("And height - "))
20         answer = 2*pi*radius*height
21
22         """Определение требуемой площади, если нужна всего цилиндра, то домножается ответ на 2 круга"""
23
24         if variant == "1":
25             print("\nAccording to our calculations, the answer is -", answer)
26         else:
27             print("\nAccording to our calculations, the answer is -",
28                 answer + 2*circle(radius))
29
30 if __name__ == '__main__':
31     cylinder()
32

```

Рисунок 7. Полученный код задания №10.

4. Выполнено задание №12, в котором требовалось создать функцию, перемножающую вводимые пользователем числа до тех пор, пока не будет введен 0, после чего функция возвращает результат перемножений.

```

Good day! Please, enter your first value - 1
Next value - 2
Next value - 3
Next value - 4
Next value - 5
Next value - 0
Very well, multiplying all the values gives us - 120

```

Рисунок 8. Полученный результат задания №12.

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  def multiply():
5      new_value = int(input("Good day! Please, enter your first value - "))
6      if new_value == 0:
7          return 0
8      answer = 1
9
10     """Ввод чисел до тех пор, пока не будет получен 0."""
11
12     while new_value != 0:
13         answer *= new_value
14         new_value = int(input("Next value - "))
15     return answer
16
17
18 if __name__ == '__main__':
19     # answer = multiply()
20
21     """Перемножение введенных пользователем чисел."""
22
23     print("Very well, multiplying all the values gives us - ", multiply())
24

```

Рисунок 9. Полученный код задания №12.

5. Выполнено задание №14, в котором требовалось определить 4 функции: `get_input` (запрашивает ввод и возвращает строку), `test_input` (проверяет, можно ли преобразовать строку в целое число, возвращает `True` или `False`), `str_to_int` (Преобразует строку в целое число), `print_int` (Выводит число на экран). Данные из 1 функции передаются во вторую, если из второй функции получено `True`, то строка передается третьей функции, после чего выводится четвертой функцией.

```

Good day! Please, enter the line - 15
15

```

Рисунок 10. Полученное значение при вводе числа.

```

Good day! Please, enter the line - 13,5

```

Рисунок 11. Полученное значение при вводе строки.

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  def get_input():
5      """Получение строки"""
6
7      return input("Good day! Please, enter the line - ")
8
9
10 def test_input(word):
11     """Анализ строки, можно ли преобразовать в число"""
12
13     if word.isdigit():
14         return True
15     else:
16         return False
17
18
19 def str_to_int(value):
20     """Преобразование в число"""
21
22     return int(value)
23
24
25 def print_int(value):
26     """Вывод числа на экран"""
27
28     print(value)
29
30
31 if __name__ == '__main__':
32     """Вызов функций для анализа строки. Если число - преобразовать и вывести."""
33
34     value = get_input()
35     if test_input(value):
36         print_int(str_to_int(value))

```

Рисунок 12. Полученный код задания №14.

6. Выполнено индивидуальное задание. В нём требовалось преобразовать код из лабораторной работы №9, внедрив функции (В индивидуальном задании 9 лабораторной работы в нём создавался словарь, хранящий информацию о людях – фамилия, имя, телефонный номер, дата рождения. Требовалось обрабатывать несколько команд – вывод всех людей (list), вывод сводки о имеющихся командах (help), добавление человека (add), вывод людей относительно требуемого месяца рождения (select “месяц”), выход из программы (exit)).

```
>>> list
There are no people in list!
>>> add
Name - Lapis
Surname - Lazuli
Telephone number - 83321005435
Date of birth (Day.Month.Year) - 3.1.2000
>>> add
Name - Amethyst
Surname - Ruby
Telephone number - 87655672332
Date of birth (Day.Month.Year) - 1.03.2001
>>> list
```

Nº	Name	Surname	Telephone	Birthday
1	Lapis	Lazuli	83321005435	3.1.2000
2	Amethyst	Ruby	87655672332	1.03.2001

```
>>> select 2
There are no people in list!
>>> select 3
```

Nº	Name	Surname	Telephone	Birthday
1	Amethyst	Ruby	87655672332	1.03.2001

Рисунок 13. Полученный результат индивидуального задания.

```

1 #!/usr/bin/env python3
2 #-*- coding: utf-8 -*-
3
4 import sys
5
6
7 def new_human():
8     """Запросить данные о человеке."""
9
10    name = input("Name - ")
11    surname = input("Surname - ")
12    telephone = input("Telephone number - ")
13    happy_birthday = input("Date of birth (Day.Month.Year) - ")
14    # Создать словарь.
15    return {
16        "name": name,
17        "surname": surname,
18        "telephone": telephone,
19        "birthday": happy_birthday
20    }
21
22
23 def display_people(people):
24     """Отобразить список людей."""
25
26    if people:
27        # Заголовки таблицы.
28        line = "|{-}{-}{-}{-}{-}{-}|".format(
29            "-" * 5, "-" * 25, "-" * 25, "-" * 25, "-" * 18)
30        print(line)
31        print("|{:^5} | {:^24} | {:^25} | {:^25} | {:^18} |".f
32              "№", "Name", "Surname", "Telephone", "Birthday"))
33        print(line)
```

Рисунок 14. Полученный код индивидуального задания.

7. Ответы на вопросы.

1) Каково назначение функций в языке Python?

Ответ: функция – обособленный участок кода, который можно вызывать, обратившись к нему по имени, которым он был назван. При вызове происходит выполнение команд тела функции. Функции можно сравнить с

небольшими программками, которые сами по себе, т. е. автономно, не исполняются, а встраиваются в обычную программу. Других ключевых отличий функций от программ нет. Функции также при необходимости могут получать и возвращать данные (обычно они их получают не с ввода (клавиатуры, файла и др.), а из вызывающей программы. Сюда же они возвращают результат своей работы). Существует множество встроенных в язык программирования функций (`print()`, `input()`, `int()`, `float()`, `str()`, `type()`). Программист всегда может определять свои функции (это пользовательские функции).

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

Ответ: в Python оператор `def` используется для определения функций, а `return` для выхода из функции и передачи данных в то место, откуда она была вызвана.

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

Ответ: локальные переменные в Python создаются внутри функции и видны только внутри этой функции (используются для временного хранения данных внутри функции). Глобальные переменные создаются вне функций и могут быть использованы в любом месте программы. Однако, если нужно изменить значение глобальной переменной внутри функции, необходимо использовать ключевое слово `global`.

4) Как вернуть несколько значений из функции в Python?

Ответ: в Python функция может вернуть несколько значений, используя кортеж (`tuple`). Для этого необходимо указать значения через запятую после `return` или заключить значения в фигурные скобки, что создаст кортеж.

5) Какие существуют способы передачи значений в функцию?

Ответ: Значения могут быть переданы в функцию в Python через позиционные аргументы (передаются в порядке, в котором они определены в определении функции), ключевые аргументы (передаются с указанием имени

параметра) и аргументы по умолчанию (значения уже установлены по умолчанию => могут быть пропущены при вызове функции).

6) Как задать значение аргументов функции по умолчанию?

Ответ: в Python значения аргументов функции можно задать по умолчанию, что позволяет вызывать функцию без указания значений для этих аргументов, если они необходимы. Задание значений по умолчанию осуществляется в определении функции, например: `def function(smth1, smth2=555, smth3="smth_else")`.

7) Каково назначение lambda-выражений в языке Python?

Ответ: lambda функции в Python – такие функции, которые не имеют названия. Их также называют анонимными. Слово «lambda» является служебным, и не отражает сути конкретной функции. Основная причина применения лямбда функций — создание функции, которая используется в коде единожды.

8) Как осуществляется оформление кода согласно PEP257?

Ответ: PEP 257 содержит рекомендации по документированию строк (docstrings) в коде на языке Python: документационные строки следует писать в тройных кавычках ("""smth"""); для модуля, класса или функции они должны быть расположены в начале их определения; для функций строки должны включать описание параметров, возвращаемого значения и возможных исключений.

9) В чём особенность однострочных и многострочных форм строк документации?

Ответ: строки документации начинаются и заканчиваются тремя кавычками, однако однострочная форма занимает не больше одной строки и предназначена для кратких описаний, в то время как многострочная форма занимает несколько строк и предоставляет более развёрнутое описание.

Вывод: в ходе выполнения лабораторной работы были приобретены навыки работы с функциями при написании программ с помощью языка программирования Python версии 3.x.