

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 5.

Выполнила:
Михеева Елена Александровна
2 курс, группа ИВТ-б-з-20-1,
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, доцент кафедры
инфокоммуникаций

(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. Был проработан пример лабораторной работы.

```
>>> help
Список команд:

add – добавить работника;
list – вывести список работников;
select <стаж> – запросить работников со стажем;
help – отобразить справку;
exit – завершить работу с программой.
>>> add
Фамилия и инициалы? Михеева Е. А.
Должность? Студент
Год поступления? 2021
>>> list
+-----+-----+-----+-----+
| No |          Ф.И.О.          | Должность | Год |
+-----+-----+-----+-----+
| 1 | Михеева Е. А. | Студент | 2021 |
+-----+-----+-----+-----+
Список работников пуст.
>>> exit
(base) mikheeva@MacBook-Pro-Elena Python_2.8 %
```

Рисунок 1. Выполнение программы примера

2. Было выполнено задание: основная ветка программы, не считая заголовков функций, состоит из двух строки кода. Это вызов функции *test()* и инструкции *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".

```
5 def positive():
6     print("Положительное")
7
8
9 def negative():
10    print("Отрицательное")
11
12
13 def test():
14    num = int(input("Введите целое число: "))
15    if num > 0:
16        positive()
17    elif num < 0:
18        negative()
19
20
21 if __name__ == "__main__":
22    test()
23
```

Рисунок 2. Программа задания №1

3. Было выполнено задание №2: 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  import math
5
6
7  def circle(radius):
8      return math.pi * radius ** 2
9
10
11 def cylinder():
12     radius = float(input("Введите радиус цилиндра: "))
13     height = float(input("Введите высоту цилиндра: "))
14
15     print("Рассчитать только площадь боковой поверхности? (да/нет): ")
16     surface_only = input().lower() == 'да'
17
18     lateral_surface_area = 2 * math.pi * radius * height
19
20     if surface_only:
21         print("Площадь боковой поверхности цилиндра:", lateral_surface_area)
22     else:
23         total_surface_area = lateral_surface_area + 2 * circle(radius)
24         print("Полная площадь цилиндра:", total_surface_area)
25
26
27 if __name__ == "__main__":
28     cylinder()
29
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Введите радиус цилиндра: 56
Введите высоту цилиндра: 102
Рассчитать только площадь боковой поверхности? (да/нет):
да
Площадь боковой поверхности цилиндра: 35889.554474609795
(base) mikheeva@MacBook-Pro-Elena Python_2.8 %
```

Рисунок 3. Программа для задания №2

4. Было выполнено задание №3: 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  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def multiply_until_zero():
6      product = 1
7      while True:
8          num = float(input("Введите число (для завершения введите 0): "))
9          if num == 0:
10             break
11             product *= num
12         return product
13
14
15 if __name__ == "__main__":
16     result = multiply_until_zero()
17     print("Произведение введенных чисел (до ввода 0):", result)
18
```

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS
	Введите число (для завершения введите 0): 2			
	Введите число (для завершения введите 0): 2			
	Введите число (для завершения введите 0): 2			
	Введите число (для завершения введите 0): 2			
	Введите число (для завершения введите 0): 2			
	Введите число (для завершения введите 0): 0			
	Произведение введенных чисел (до ввода 0): 32.0			

Рисунок 4. Программа для задания №3

5. Было выполнено задние №4: 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.

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.

```

5     def get_input():
6         user_input = input("Введите значение: ")
7         return user_input
8
9
10    def test_input(value):
11        if value.isdigit():
12            return True
13        else:
14            return False
15
16
17    def str_to_int(value):
18        return int(value)
19
20
21    def print_int(value):
22        print("Преобразованное значение:", value)
23
24
25    if __name__ == "__main__":
26        input_str = get_input()
27
28        if test_input(input_str):
29            int_value = str_to_int(input_str)
30
31            print_int(int_value)
32        else:
33            print("Невозможно преобразовать введенное значение к целому числу.")
34

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

Введите значение: 46
Преобразованное значение: 46
(base) mikheeva@MacBook-Pro-Elena Python_2.8 %

Рисунок 5. Программа для задания №4

6. Было выполнено индивидуальное задание согласно варианту №5.

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  # Использовать словарь, содержащий следующие ключи:
5  # название пункта назначения рейса; номер рейса; тип самолета.
6  # Написать программу, выполняющую следующие действия: ввод с клавиатуры
7  # данных в список, состоящий из словарей заданной структуры;
8  # записи должны быть размещены в алфавитном порядке по названию
9  # пунктов назначения; вывод на экран пунктов назначения и номеров рейсов,
10 # обслуживаемых самолетом, тип которого введен с клавиатуры;
11 # если таких рейсов нет, вывести на экран соответствующее сообщение.
12
13
14 def add_flight():
15     destination = input("Введите название пункта назначения: ")
16     flight_number = input("Введите номер рейса: ")
17     plane_type = input("Введите тип самолета: ")
18
19     new_flight = {
20         'destination': destination,
21         'flight number': flight_number,
22         'type of plane': plane_type
23     }
24
25     flights.append(new_flight)
26     flights.sort(key=lambda x: x['destination'])
27
28
29 def list_flights():
30     line = '{:~4}--{:~4}--{:~4}--{:~4}'.format(
31         '-.' * 4,
32         '-.' * 30,
33         '-.' * 20,
34         '-.' * 20
35     )
36     print(line)
37
38     print(
39         '{:~4} | {:~30} | {:~20} | {:~20}'.format(
40             "No",
41             "Пункт назначения",
42             "Номер рейса",
43             "Тип самолета"
44         )
45     )
46
47     print(line)
48
49     for idx, flight in enumerate(flights, 1):
50         print(
51             '{:~4} | {:~30} | {:~20} | {:~20}'.format(
52                 idx,
53                 flight.get('destination', ''),
54                 flight.get('flight number', ''),
55                 flight.get('type of plane', 0)
56             )
57         )
58         print(line)
59
60 def find_flights():
61     find_type = input("Введите тип самолета для поиска: ")
62     found = []
63
64
65     for flight in flights:
66         if flight['type of plane'] == find_type:
67             found.append(flight)
68
69     if not found:
70         print("Печаль! Самолет типа '{find_type}' не найден.")
71     else:
72         line = '{:~4}--{:~4}--{:~4}--{:~4}'.format(
73             '-.' * 4,
74             '-.' * 30,
75             '-.' * 20,
76             '-.' * 20
77         )
78         print(line)
79
80         print(
81             '{:~4} | {:~30} | {:~20} | {:~20}'.format(
82                 "No",
83                 "Пункт назначения",
84                 "Номер рейса",
85                 "Тип самолета"
86             )
87         )
88
89         print(line)
90
91         for idx, found_flight in enumerate(found, 1):
92             print(
93                 '{:~4} | {:~30} | {:~20} | {:~20}'.format(
94                     idx,
95                     found_flight.get('destination', ''),
96                     found_flight.get('flight number', ''),
97                     found_flight.get('type of plane', 0)
98                 )
99             )
100             print(line)
101
102
103 if __name__ == '__main__':
104     flights = []
105
106     print(">>> Выберите нужную команду: add, list, find или exit")
107
108     while True:
109         command = input(">>> ").lower()
110
111         if command == 'exit':
112             break
113
114         elif command == 'add':
115             add_flight()
116
117         elif command == 'list':
118             list_flights()
119
120         elif command == 'find':
121             find_flights()
122

```

Рисунок 6. Программа для индивидуального задания №1

```

>>> Выберите нужную команду: add, list, find или exit
>>> add
Введите название пункта назначения: stavoropol
Введите номер рейса: 23189
Введите тип самолета: boeing
>>> list

```

No	Пункт назначения	Номер рейса	Тип самолета
1	stavoropol	23189	boeing

```

>>>

```

Рисунок 7. Результат работы программы индивидуального задания

Ответы на контрольные вопросы.

1. Каково назначение функций в языке программирования Python?

Функции в языке программирования Python используются для группировки кода, чтобы он мог быть многократно использован, делая программу более читаемой и легко управляемой. Функции также позволяют разделить большие программы на более мелкие, что упрощает разработку и управление кодом.

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

Оператор `def` используется для определения функции в Python. Он указывает интерпретатору, что следующий блок кода является телом функции. Оператор `return` используется для возврата значения из функции. Когда интерпретатор Python достигает оператора `return`, он возвращает указанное значение и завершает выполнение функции.

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

Локальные переменные объявляются внутри функции и доступны только внутри этой функции. Глобальные переменные объявляются вне функций и доступны во всем коде программы. При написании функций в Python, локальные переменные используются для временного хранения данных, в то время как глобальные переменные могут быть использованы в различных частях программы.

4. Как вернуть несколько значений из функции Python?

Для возврата нескольких значений из функции в Python используется механизм кортежей. Функция может вернуть кортеж, содержащий несколько значений, и затем эти значения могут быть присвоены различным переменным при вызове функции.

5. Какие существуют способы передачи значений в функцию?

Значения могут быть переданы в функцию в Python через позиционные аргументы, ключевые аргументы и аргументы по умолчанию. Позиционные аргументы передаются в порядке, в котором они определены в сигнатуре

функции. Ключевые аргументы передаются с указанием имени параметра. Аргументы по умолчанию имеют значения по умолчанию и могут быть пропущены при вызове функции.

6. Как задать значение аргументов функции по умолчанию?

Для задания значения аргументов функции по умолчанию в Python используется синтаксис "переменная=значение" в сигнатуре функции. Если значение не передается при вызове функции, будет использовано значение по умолчанию.

7. Каково назначение lambda-выражений в языке Python?

Lambda-выражения в Python представляют собой анонимные функции, которые могут содержать только одно выражение. Они обычно используются в ситуациях, когда требуется небольшая функция в одном месте кода.

8. Как осуществляется документирование кода согласно PEP257?

Документирование кода в Python согласно PEP257 включает в себя использование строк документации (docstrings) для описания модулей, функций, классов и методов. Строки документации должны быть заключены в тройные кавычки и предоставлять информацию о назначении, использовании и возвращаемых значениях функций, классов и методов.

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

Однострочные строки документации начинаются и заканчиваются тройными кавычками и предназначены для краткого описания модулей, функций, классов или методов. Они обычно используются для кратких пояснений и описаний.

Многострочные строки документации также начинаются и заканчиваются тройными кавычками, но могут занимать несколько строк и предоставлять более подробное описание. Они обычно используются для более полного и подробного документирования, включая информацию о назначении, использовании и возвращаемых значениях функций, классов и методов.