

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 №2.8
дисциплины «Основы кроссплатформенного программирования»
Вариант ____

Выполнил:

Быковская Стефания Станиславовна
1 курс, группа ИТС-б-0-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи», направ-
ленность (профиль) «Инфокоммуника-
ционные системы и сети», очная форма
обучения

(подпись)

Руководитель практики:

Воронкин Р.А. доцент кафедры инфо-
коммуникаций

(подпись)

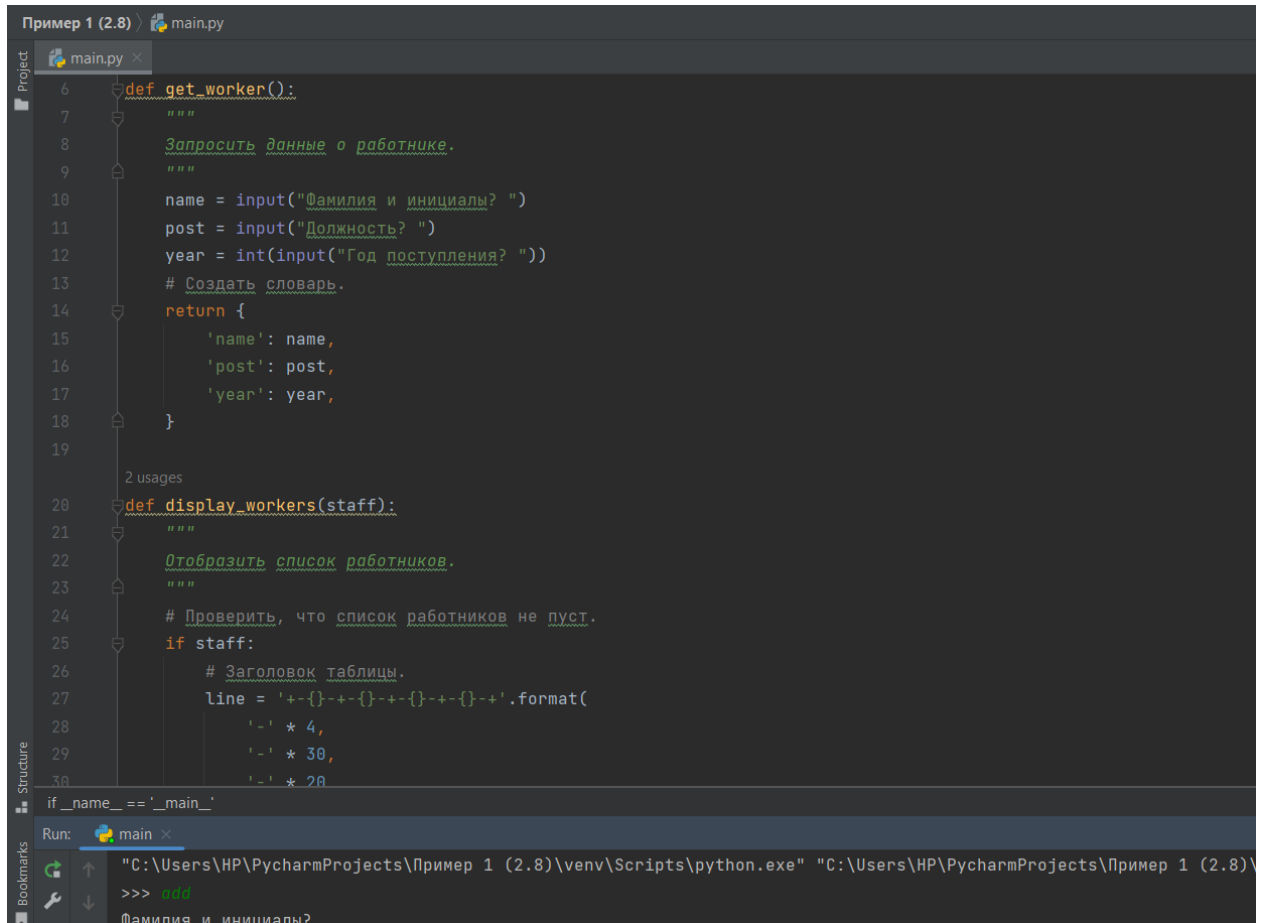
Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Работа с функциями в языке Python

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

Порядок выполнения работы:



```
Пример 1 (2.8) > main.py
main.py
6 def get_worker():
7     """
8     Запросить данные о работнике.
9     """
10    name = input("Фамилия и инициалы? ")
11    post = input("Должность? ")
12    year = int(input("Год поступления? "))
13    # Создать словарь.
14    return {
15        'name': name,
16        'post': post,
17        'year': year,
18    }
19
20    2 usages
21 def display_workers(staff):
22     """
23     Отобразить список работников.
24     """
25     # Проверить, что список работников не пуст.
26     if staff:
27         # Заголовок таблицы.
28         line = '+-{}-+-{}-+-{}-+-{}-+'.format(
29             '-' * 4,
30             '-' * 30,
31             '-' * 20
32         )
33
34 if __name__ == '__main__':
35     test()
36     if __name__ == '__main__':
```

Рисунок 1 – Выполнение примера 1.

Общее 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".

```
Общее задание 1 (2.8) > main.py
main.py x
1 usage
1 def test():
2     number = int(input("Enter an integer : "))
3     if number > 0:
4         positive()
5     else:
6         negative()
7
8     1 usage
8 def positive():
9     print("Positive")
10    1 usage
10 def negative():
11     print("Negative")
12     test()

Run: main x
"C:\Users\HP\PycharmProjects\Общее задание 1 (2.8)\venv\Scripts\python.exe"
Enter an integer : 5
Positive
```

Рисунок 2 – Выполнение общего задания №1.

Общее задание 2.

в основной ветке программы вызывается функция `cylinder()`, которая вычисляет площадь цилиндра. В теле `cylinder()` определена функция `circle()`, вычисляющая площадь круга по формуле $S = \pi r^2$. В теле `cylinder()` у пользователя спрашивается, хочет ли он получить только площадь боковой поверхности цилиндра, которая вычисляется по формуле $S_{бок} = 2\pi r h$, или полную площадь цилиндра. В последнем случае к площади боковой поверхности цилиндра должен добавляться удвоенный результат вычислений функции `circle()`.

```
Общее задание 2 (2.8) > main.py
main.py x
1 usage
2 def cylinder():
3     r = float(input("Enter radius: "))
4     h = float(input("Enter height: "))
5     # площадь боковой поверхности цилиндра:
6     side = 2 * 3.14 * r * h
7     # площадь одного основания цилиндра:
8     circle = 3.14 * r**2
9     # полная площадь цилиндра:
10    full = side + 2 * circle
11    return full
12
13    square = cylinder()
14    print(square)
```

Run: main x

"C:\Users\HP\PycharmProjects\Общее задание 2 (2.8)\venv\Scripts\python.exe" "C:\Us
Enter radius: 8
Enter height: 4
602.88

Рисунок 3 – Выполнение общего задания №2.

Общее 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.

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 def get_input():
5     return input('Введите параметры: ')
6
7 def test_input(num):
8     try:
9         if int(num):
10             return True
11     except ValueError:
12         return False
13
14 def str_to_int(num):
15     return int(num)
16
17 def print_int(num):
18     print(num)
19
20 if __name__ == '__main__':
21     number = get_input()
22     if test_input(number):
23         number = str_to_int(number)
24         print_int(number)
25     else:
26         print('число нельзя преобразовать')
```

Run: main ×

"C:\Users\HP\PycharmProjects\Общее 4 (2.8)\venv\Scripts\python.exe" "C:\U
Введите параметры: 12
12

Рисунок 4 – Общее задание №4.

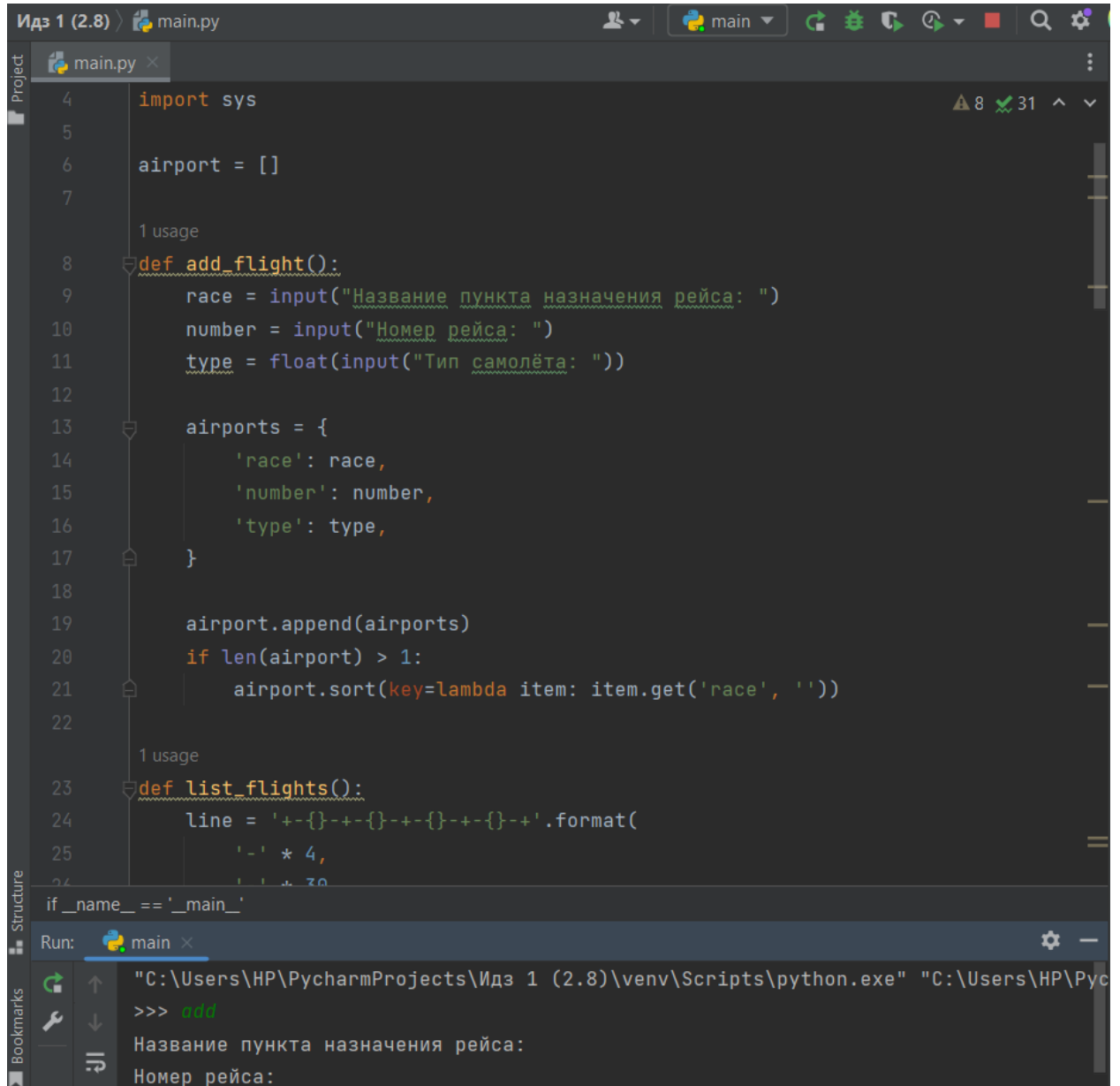
Индивидуальное задание

Вариант 5.

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

5. Использовать словарь, содержащий следующие ключи: название пункта назначения рейса; номер рейса; тип самолета. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть размещены в ал-

фавитном порядке по названиям пунктов назначения; вывод на экран пунктов назначения и номеров рейсов, обслуживаемых самолетом, тип которого введен с клавиатуры; если таких рейсов нет, выдать на дисплей соответствующее сообщение.



```
Идз 1 (2.8) > main.py
main.py x
4 import sys
5
6 airport = []
7
8 1 usage
9 def add_flight():
10     race = input("Название пункта назначения рейса: ")
11     number = input("Номер рейса: ")
12     type = float(input("Тип самолёта: "))
13
14     airports = {
15         'race': race,
16         'number': number,
17         'type': type,
18     }
19
20     airport.append(airports)
21     if len(airport) > 1:
22         airport.sort(key=lambda item: item.get('race', ''))
23
24 1 usage
25 def list_flights():
26     line = '+-{}-+-{}-+-{}-+-{}-+'.format(
27         '-' * 4,
28         '-' * 4,
29         '-' * 4,
30         '-' * 4,
31     )
32
33 if __name__ == '__main__':
34     Run: main x
35     "C:\Users\HP\PycharmProjects\Идз 1 (2.8)\venv\Scripts\python.exe" "C:\Users\HP\Pyc
36     >>> add
37     Название пункта назначения рейса:
38     Номер рейса:
```

Ответы на контрольные вопросы

1) Каково назначение функций в языке программирования Python?

Функции в языке программирования Python используются для организации кода в отдельные блоки, которые можно вызывать и использовать многократно в программе. Функции позволяют повысить читаемость, упростить отладку и сократить количество дублирующегося кода. Также функции мо-

гут быть использованы для создания модулей, которые могут быть использованы в различных проектах.

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

Оператор `def` используется для определения функции, а оператор `return` используется для возврата значения из функции. При вызове функции, ее тело выполняется, а затем значение, указанное в операторе `return`, возвращается в вызывающую программу.

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

Локальные переменные создаются внутри функции и существуют только в рамках этой функции. Глобальные переменные создаются вне функций и могут быть доступны из любой части программы. При написании функций в Python важно учитывать область видимости переменных и использовать их правильно. Назначение локальных переменных - это изолировать код функции от других частей программы, чтобы избежать изменения переменных из других частей программы, которые могут негативно повлиять на работу функции. Назначение глобальных переменных - это обеспечить доступ к данному объекту из любой части программы.

Однако, существует опасность перезаписи глобальных переменных, и использование глобальных переменных следует использовать с осторожностью.

4) Как вернуть несколько значений из функции Python?

В Python можно вернуть несколько значений из функции, используя кортеж. Для этого возвращаемые значения перечисляются через запятую внутри круглых скобок. В выбираемые значения можно обратиться по индексу.

5) Какие существуют способы передачи значений в функцию?

В языке программирования Python значения могут быть переданы в функцию несколькими способами:

- позиционные аргументы (передача аргументов в порядке их следования);
- именованные аргументы (передача аргументов с указанием их имени);
- аргументы по умолчанию (передача аргументов со значениями по умолчанию);
- распаковывание списков и словарей.

6) Как задать значение аргументов функции по умолчанию?

Для того, чтобы задать значение аргументов функции по умолчанию в Python, нужно указать это значение после имени аргумента в определении функции.

7) Каково назначение lambda-выражений в языке Python?

Lambda-выражения в языке Python представляют собой способ создания анонимных функций без явного определения имени функции. В lambda-выражениях объединяются три элемента: аргументы, операторразделитель и тело функции.

8) Как осуществляется документирование кода согласно PEP257?

Документирование кода в Python осуществляется согласно рекомендациям PEP257. Для документирования функций и модулей используются строки документации (docstrings), которые должны быть помещены в начало определения функции или модуля. Для того чтобы оформить документацию в соответствии с PEP257, используют следующие рекомендации:

- Docstring должен начинаться с однострочного описания того, что делает объект (функция, класс и т.д.). Это описание следует начинать с заглавной буквы и заканчивать точкой.
- За однострочным описанием должна следовать пустая строка.
- Если это функция или метод, то следует указать, какие аргументы она принимает, какие они должны быть по типу и за что они отвечают.
- Если функция или метод что-то возвращает, то это также должно быть указано в документации.

— Если объект имеет какие-то особенности или ограничения, то их нужно описать.

— Если есть примеры использования, то их нужно привести.

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

Однострочная форма документации заключается в одном ряду и применяется для краткого описания. Она начинается со знака # и двух пробелов, специально размещенных после знака #. Многострочная форма документации позволяет вставлять более детальные описания. Она начинается и заканчивается тройными кавычками и предоставляет возможность размещения внутри описания более одного абзаца. Эта форма чаще применяется при описании функций и модулей.

Выводы: приобрела навыки по работе с функциями при написании программ с помощью языка программирования Python 3.x.