

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»

Выполнил:
Говоров Егор Юрьевич
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. технических
наук, доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

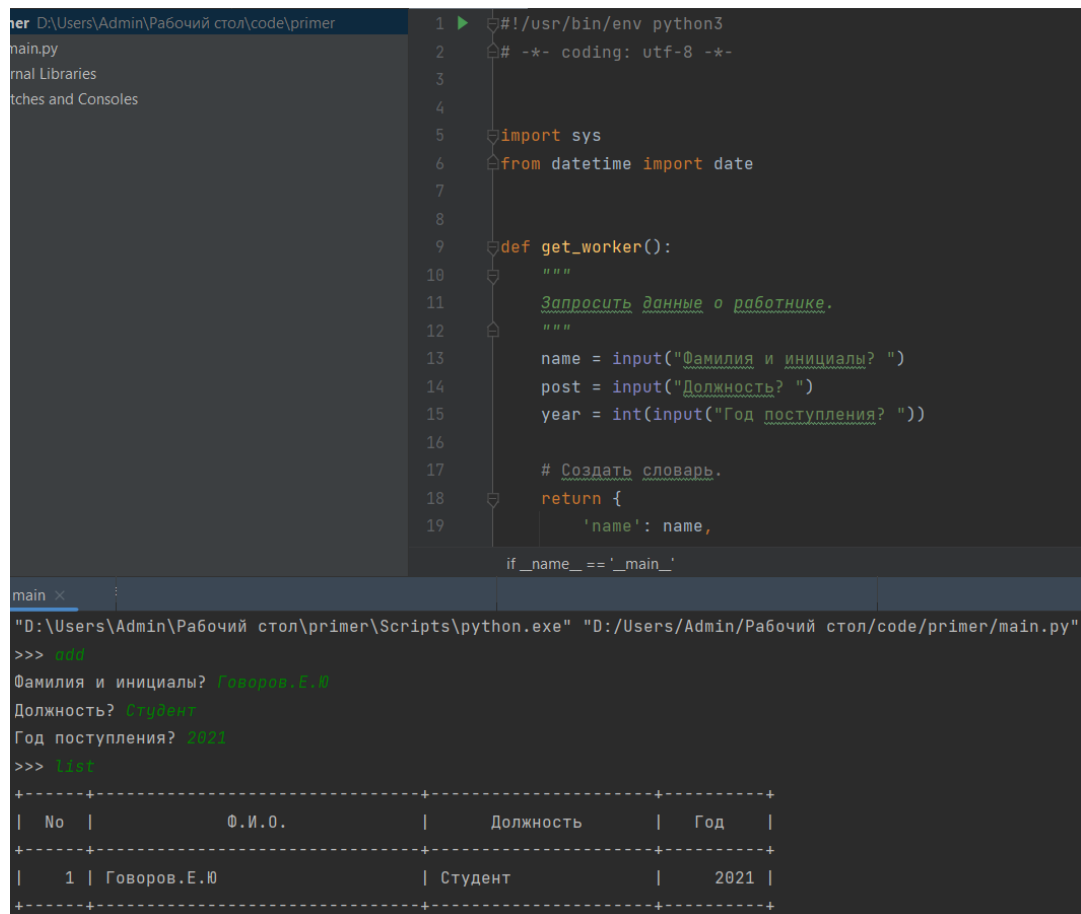
Ставрополь, 2023 г.

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.

Ход работы

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python. Выполнил клонирование созданного репозитория.
2. Дополнил файл .gitignore необходимыми правилами.
3. Организовал созданный репозиторий в соответствие с моделью ветвления git-flow.
4. Проработал пример лабораторной работы. Создал для него отдельный модуль языка Python. Привел в отчете скриншоты результата выполнения программы примера.

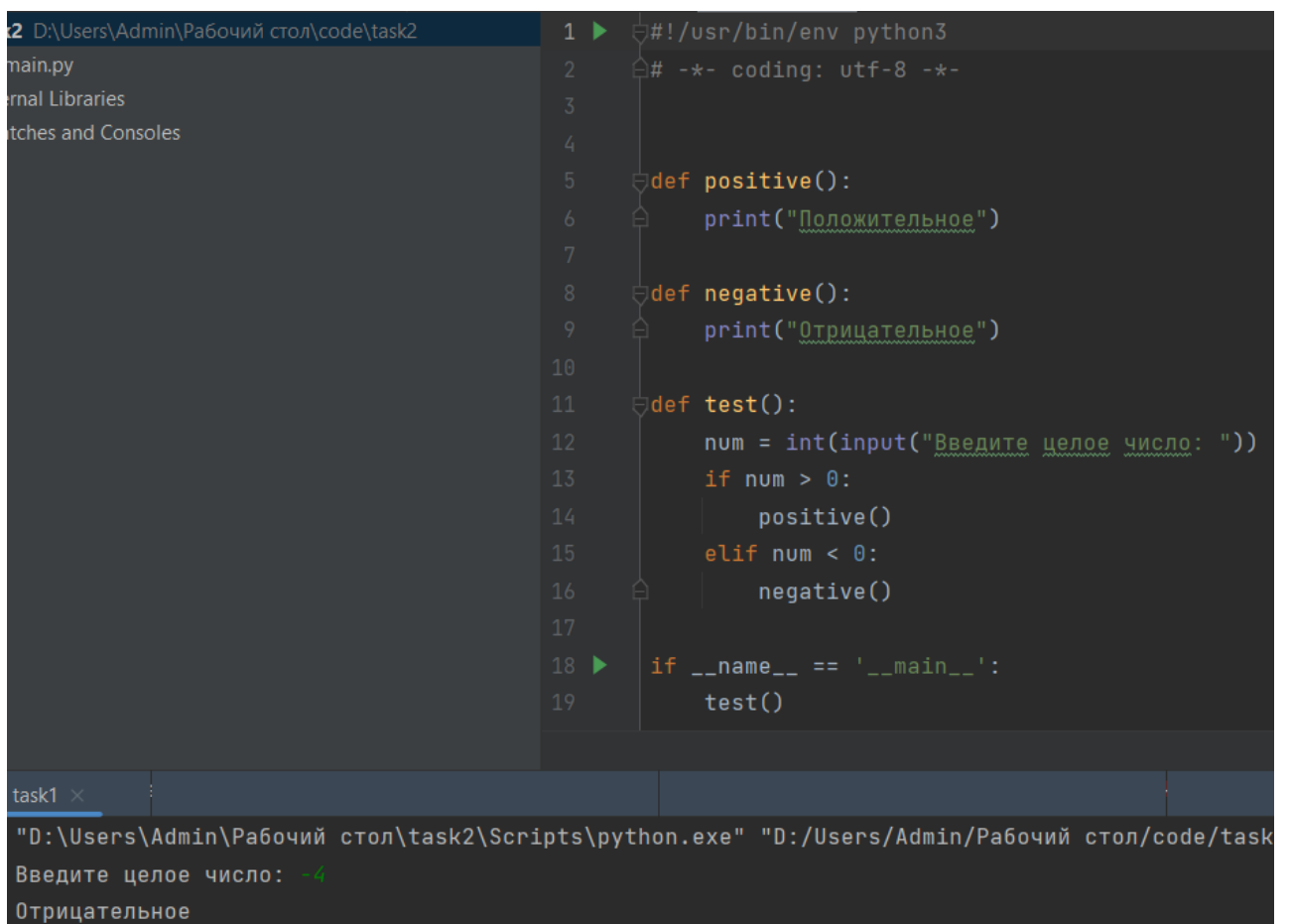


```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 import sys
6 from datetime import date
7
8
9 def get_worker():
10     """
11     Запросить данные о работнике.
12     """
13     name = input("Фамилия и инициалы? ")
14     post = input("Должность? ")
15     year = int(input("Год поступления? "))
16
17     # Создать словарь.
18     return {
19         'name': name,
```

```
main x
D:\Users\Admin\Рабочий стол\primer\Scripts\python.exe "D:/Users/Admin/Рабочий стол/code/primer/main.py"
>>> add
Фамилия и инициалы? Говоров.Е.Ю
Должность? Студент
Год поступления? 2021
>>> list
+-----+-----+-----+-----+
| No |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Говоров.Е.Ю              |      Студент       |      2021     |
+-----+-----+-----+-----+
```

Рисунок 1. Результат работы программы из примера 1

5. Решил 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



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 def positive():
6     print("Положительное")
7
8 def negative():
9     print("Отрицательное")
10
11 def test():
12     num = int(input("Введите целое число: "))
13     if num > 0:
14         positive()
15     elif num < 0:
16         negative()
17
18 if __name__ == '__main__':
19     test()
```

task1 x

"D:\Users\Admin\Рабочий стол\task2\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/code/task2/main.py"

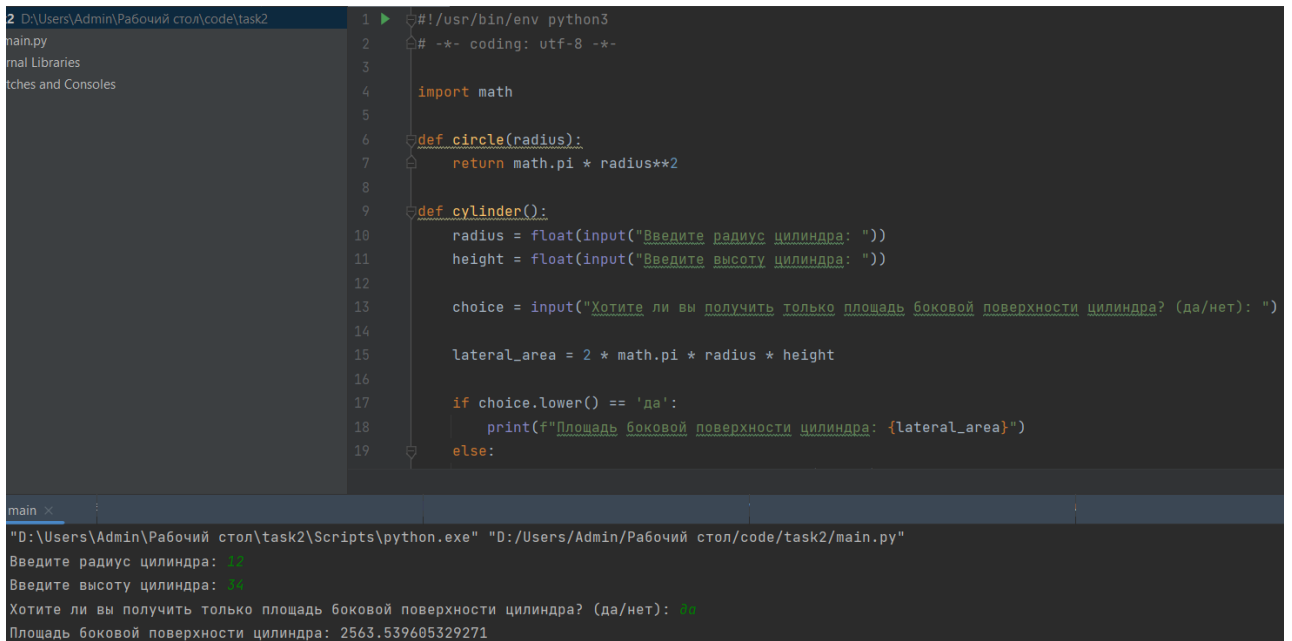
Введите целое число: -4

Отрицательное

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".

Рисунок 2. Результат работы программы из задачи 1

6. Решил следующую задачу: в основной ветке программы вызывается функция `cylinder()`, которая вычисляет площадь цилиндра. В теле `cylinder()` определена функция `circle()`, вычисляющая площадь круга по формуле πr^2 . В теле `cylinder()` у пользователя спрашивается, хочет ли он получить только площадь боковой поверхности цилиндра, которая вычисляется по формуле $2\pi rh$ или полную площадь цилиндра. В последнем случае к площади боковой поверхности цилиндра должен добавляться удвоенный результат вычислений функции `circle()`.



```
2 D:\Users\Admin\Рабочий стол\code\task2
main.py
ernal Libraries
atches and Consoles

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import math
5
6  def circle(radius):
7      return math.pi * radius**2
8
9  def cylinder():
10     radius = float(input("Введите радиус цилиндра: "))
11     height = float(input("Введите высоту цилиндра: "))
12
13     choice = input("Хотите ли вы получить только площадь боковой поверхности цилиндра? (да/нет): ")
14
15     lateral_area = 2 * math.pi * radius * height
16
17     if choice.lower() == 'да':
18         print(f"Площадь боковой поверхности цилиндра: {lateral_area}")
19     else:
```

main x

"D:\Users\Admin\Рабочий стол\task2\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/code/task2/main.py"

Введите радиус цилиндра: 13

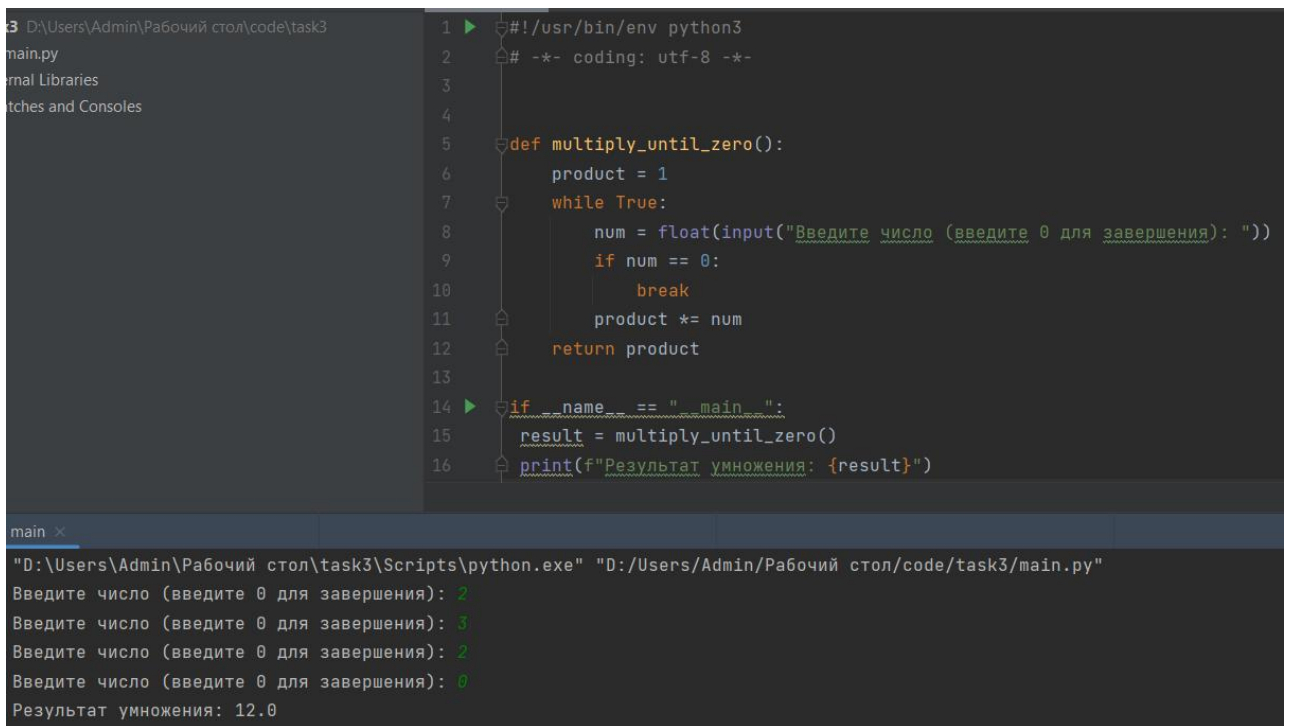
Введите высоту цилиндра: 14

Хотите ли вы получить только площадь боковой поверхности цилиндра? (да/нет): да

Площадь боковой поверхности цилиндра: 2563.539605329271

Рисунок 3. Результат работы программы из задачи 2

7. Решил 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.



The image shows a code editor with a Python script and its execution output. The script defines a function `multiply_until_zero()` that calculates the product of numbers entered by the user until they enter 0. The output shows the user entering 2, 3, 2, and 0, resulting in a product of 12.0.

```
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 (введите 0 для завершения): "))
9         if num == 0:
10             break
11         product *= num
12     return product
13
14 if __name__ == "__main__":
15     result = multiply_until_zero()
16     print(f"Результат умножения: {result}")
```

main x

"D:\Users\Admin\Рабочий стол\task3\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/code/task3/main.py"

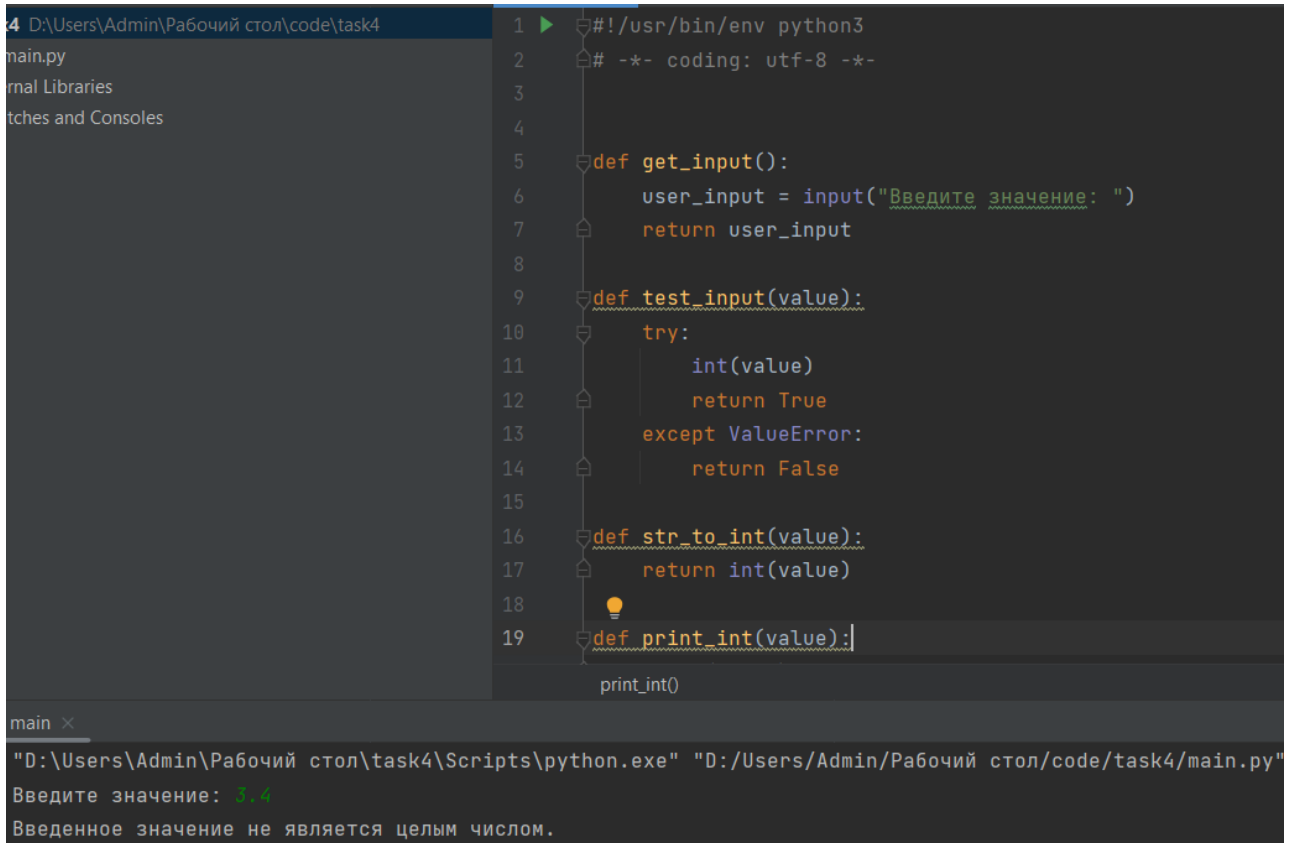
Введите число (введите 0 для завершения): 2
Введите число (введите 0 для завершения): 3
Введите число (введите 0 для завершения): 2
Введите число (введите 0 для завершения): 0
Результат умножения: 12.0

Рисунок 4. Результат работы программы из задачи 3

8. Решил 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.



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 def get_input():
6     user_input = input("Введите значение: ")
7     return user_input
8
9 def test_input(value):
10     try:
11         int(value)
12         return True
13     except ValueError:
14         return False
15
16 def str_to_int(value):
17     return int(value)
18
19 def print_int(value):
20     print_int()
```

main x

"D:\Users\Admin\Рабочий стол\task4\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/code/task4/main.py"

Введите значение: 3.4

Введенное значение не является целым числом.

Рисунок 5. Результат работы программы из задачи 4

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

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 def add_flight():
5     destination = input("Введите название пункта назначения: ")
6     flight_number = input("Введите номер рейса: ")
7     aircraft_type = input("Введите тип самолета: ")
8
9     return {'название пункта назначения': destination,
10            'номер рейса': flight_number,
11            'тип самолета': aircraft_type}
12
13 def print_flights(flights):
14     line = '+-{}-+-{}-+-{}-+'.format(
15         '-' * 30,
16         '-' * 20,
17         '-' * 15
18     )
19     print(line)
```

main x

Введите номер действия: 1

Название пункта назначения	Номер рейса	Тип самолета
Москва	2321	Груз

Выберите действие:

1. Добавить рейс
2. Вывести список рейсов
3. Поиск рейсов по типу самолета
4. Выйти

Рисунок 6. Результат работы программы из индивидуального задания

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.

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