

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 ПРАКТИЧЕСКОЙ РАБОТЕ №16
дисциплины «Программирование на Python»
Вариант 23

Выполнил:
Мотовилов Вадим Борисович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Информатика и вычислительная
техника», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Порядок выполнения работы:

1. Создал репозиторий и скопировал его

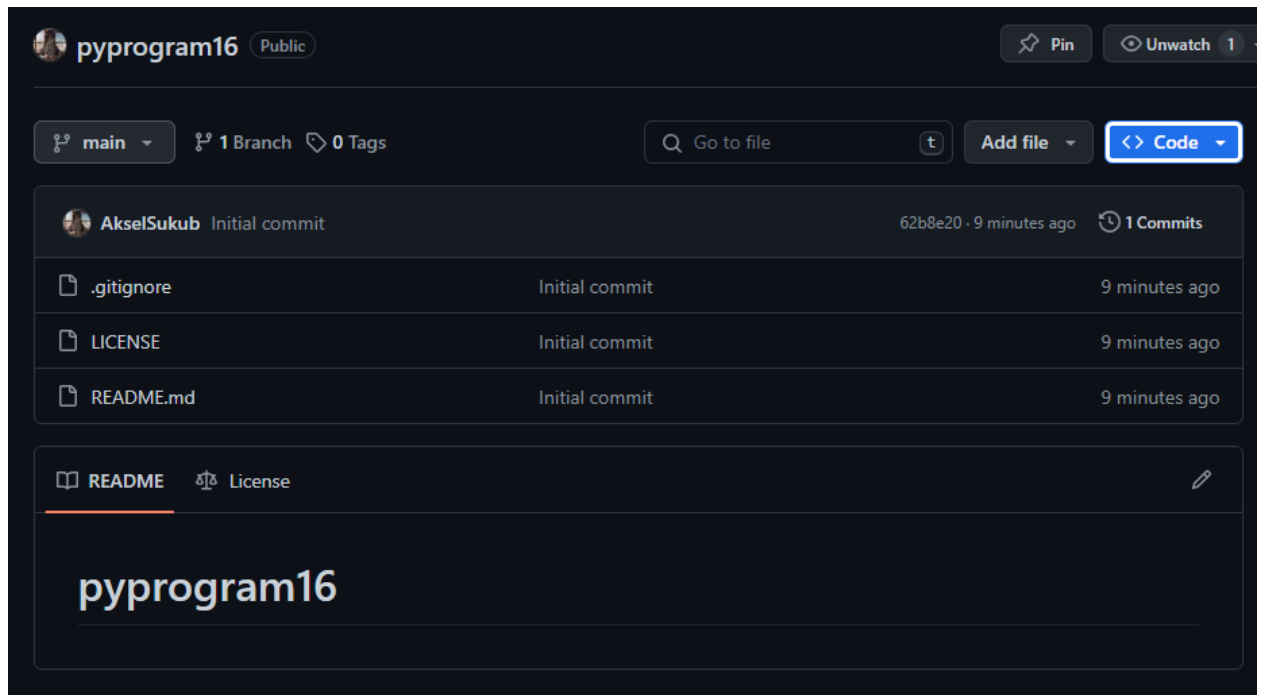


Рисунок 1. Созданный репозиторий

2. Изменил файл .gitignore и README.rm и добавил git flow

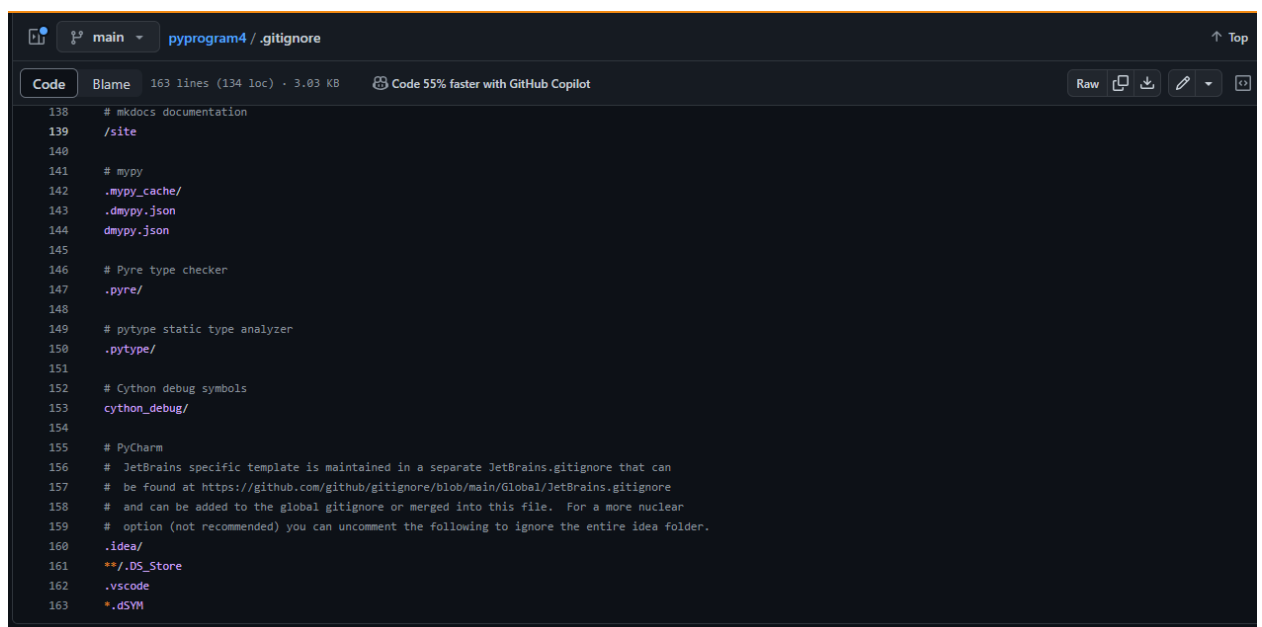
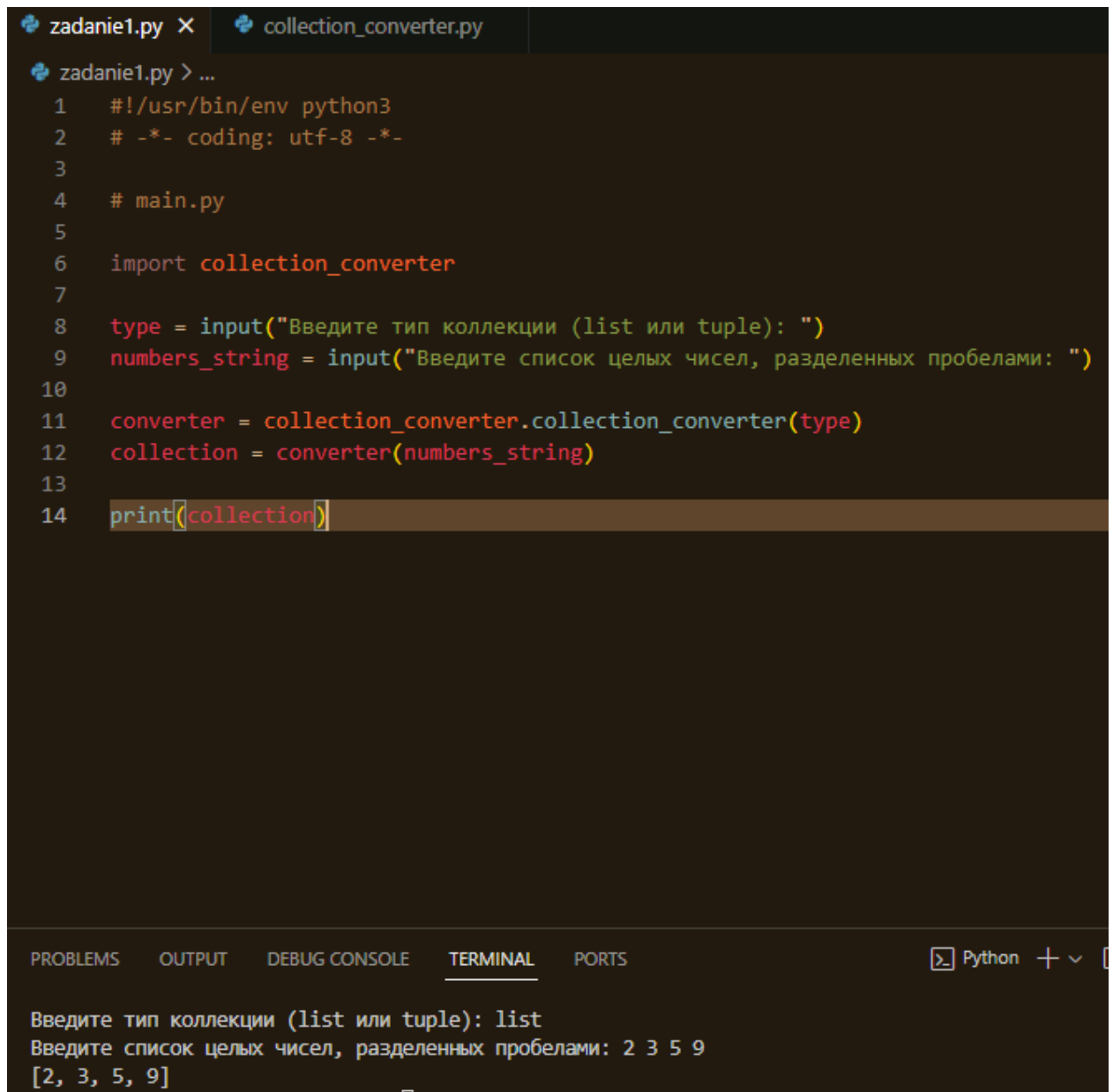


Рисунок 2. Измененный файл .gitignore

3. Выполнил задания

Выполнить индивидуальное задание лабораторной работы 2.11, оформив все функции программы в виде отдельного модуля. Разработанный модуль должен быть подключен в основную программу с помощью одного из вариантов команды `import`. Номер варианта уточнить у преподавателя.

```
collection_converter.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  # collection_converter.py
5
6  def collection_converter(type):
7      def inner_func(string):
8          numbers = map(int, string.split())
9          if type == 'list':
10             return list(numbers)
11         else:
12             return tuple(numbers)
13     return inner_func
```



The image shows a code editor with two tabs: `zadanie1.py` and `collection_converter.py`. The `zadanie1.py` tab is active, displaying a Python script. The script imports `collection_converter` and uses `input` to get a collection type and a list of numbers. It then uses the `collection_converter` module to convert the input and prints the result. The terminal at the bottom shows the execution of the script, with the user inputting `list` and `2 3 5 9`, resulting in the output `[2, 3, 5, 9]`.

```
zadanie1.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  # main.py
5
6  import collection_converter
7
8  type = input("Введите тип коллекции (list или tuple): ")
9  numbers_string = input("Введите список целых чисел, разделенных пробелами: ")
10
11  converter = collection_converter.collection_converter(type)
12  collection = converter(numbers_string)
13
14  print(collection)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + v [

Введите тип коллекции (list или tuple): list
Введите список целых чисел, разделенных пробелами: 2 3 5 9
[2, 3, 5, 9]

Рисунок 3 . Выполнение 1 задания

Выполнить индивидуальное задание лабораторной работы 2.8, оформив все классы программы в виде отдельного пакета. Разработанный пакет должен быть подключен в основную программу с помощью одного из вариантов команды `import`. Настроить соответствующим образом переменную `__all__` в файле `__init__.py` пакета. Номер варианта уточнить у преподавателя.

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

# flight.py

def input_flight_info():
    flights = []
    while True:
        destination = input("Введите пункт назначения рейса (или 'q' для выхода): ")
        if destination.lower() == 'q':
            break
        flight_number = int(input("Введите номер рейса: "))
        aircraft_type = input("Введите тип самолета: ")
        flights.append({
            'destination': destination,
            'flight_number': flight_number,
            'aircraft_type': aircraft_type
        })
    flights.sort(key=lambda x: x['flight_number'])
    return flights

def print_flights_to_destination(flights, destination):

```

```

__init__.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  # __init__.py
5
6  from .flight_info import input_flight_info, print_flights_to_destination
7
8  __all__ = ['input_flight_info', 'print_flights_to_destination']
9

```

```
zadanie2.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  # main.py
5
6  from flight_info import *
7
8  def main():
9      flights = input_flight_info()
10     destination = input("Введите пункт назначения, чтобы увидеть информацию о рейсах: ")
11     print_flights_to_destination(flights, destination)
12
13 if __name__ == "__main__":
14     main()
15
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Введите пункт назначения рейса (или 'q' для выхода): Москва
Введите номер рейса: 3
Введите тип самолета: Миг-8
Введите пункт назначения рейса (или 'q' для выхода): Ставрополь
Введите номер рейса: 7
Введите тип самолета: Ил-2
Введите пункт назначения рейса (или 'q' для выхода): q
Введите пункт назначения, чтобы увидеть информацию о рейсах: Москва
Номер рейса: 3, тип самолета: Миг-8

Рисунок 4 . Выполнение 2 задания

Ответы на контрольные вопросы:

1. **Модуль в Python** - это файл, содержащий определения и инструкции Python. Модули используются для организации кода и облегчения его повторного использования. Модуль может определять функции, классы и переменные. Модуль может также включать исполняемый код.

2. **Способы подключения модулей в Python:**

`import module_name`: Импортирует модуль с указанным именем. После этого вы можете обращаться к функциям и классам этого модуля, используя `module_name.function_name` или `module_name.ClassName`.

`from module_name import function_name:` Импортирует только указанную функцию или класс из модуля. После этого вы можете использовать функцию или класс напрямую, без указания имени модуля.

`from module_name import *:` Импортирует все функции и классы из модуля. После этого вы можете использовать их напрямую, без указания имени модуля. Этот метод не рекомендуется из-за возможности конфликтов имен.

3. **Пакет в Python** - это способ организации связанных модулей Python в общую иерархию каталогов. Пакеты представляют собой простой способ распределения множества модулей, которые могут быть упакованы и распространены вместе. Пакеты могут содержать подпакеты, которые, в свою очередь, могут содержать модули.

4. **Файл `__init__.py`** - это специальный файл, который Python обрабатывает каждый раз, когда он импортирует пакет. Этот файл может быть пустым или содержать код Python, который нужно выполнить при импорте пакета. Файл `__init__.py` также используется для указания того, что каталог должен считаться пакетом Python, так что он может быть импортирован как модуль.

5. **Переменная `__all__` в файле `__init__.py`** - это список имен модулей или объектов, которые должны быть импортированы, когда клиент импортирует пакет с использованием конструкции `from package import *`. Если `__all__` не определен, оператор `import *` импортирует все имена, которые не начинаются с подчеркивания.