**МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение высшего образования**

**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Институт цифрового развития**

**Кафедра информационных систем и технологий**

Отчет по лабораторной работе №9.

Дисциплина: **«**Основы программной инженерии**»**

**Выполнил:**

Студент группы ПИЖ-б-о-22-1, направление подготовки: 09.03.04

«Программная инженерия»

ФИО: Гуртовой Ярослав Дмитриевич

**Проверил:**

Воронкин Р. А.

Ставрополь 2023

Тема: Лабораторная работа 2.6 Работа со словарями в языке Python.

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

Выполнение работы:

1. Изучил теоретический материал работы.
2. Создал репозиторий на git.hub.

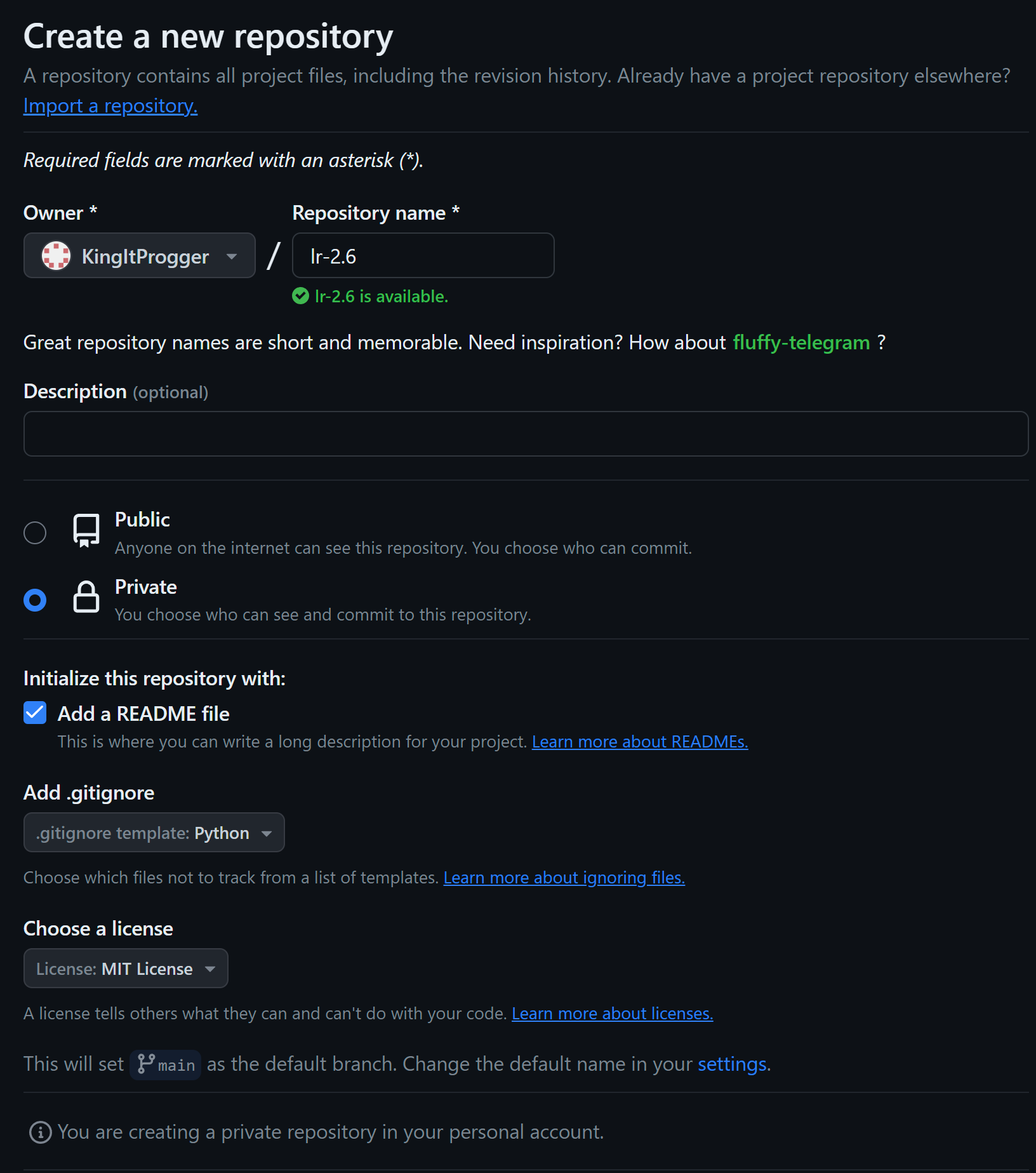


Рисунок 1 – создание репозитория

1. Клонировал репозиторий.

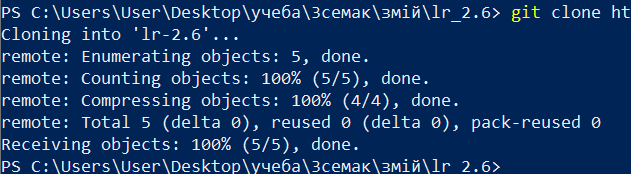


Рисунок 2 – клонирование репозитория 4.

Дополнить файл gitignore необходимыми правилами.

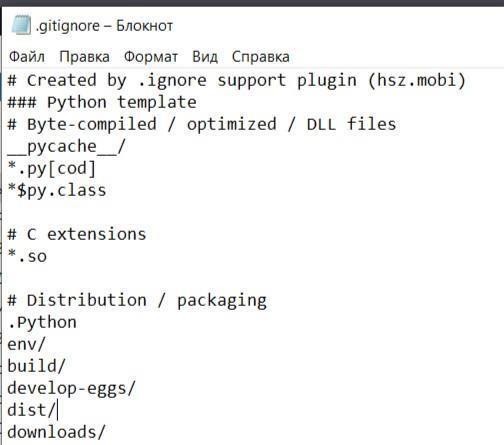


Рисунок 3 – .gitignore для IDE PyCharm

1. Организовать свой репозиторий в соответствии с моделью ветвления git-flow.

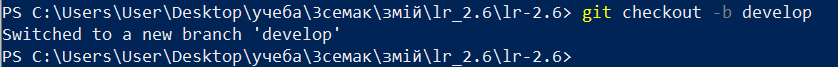
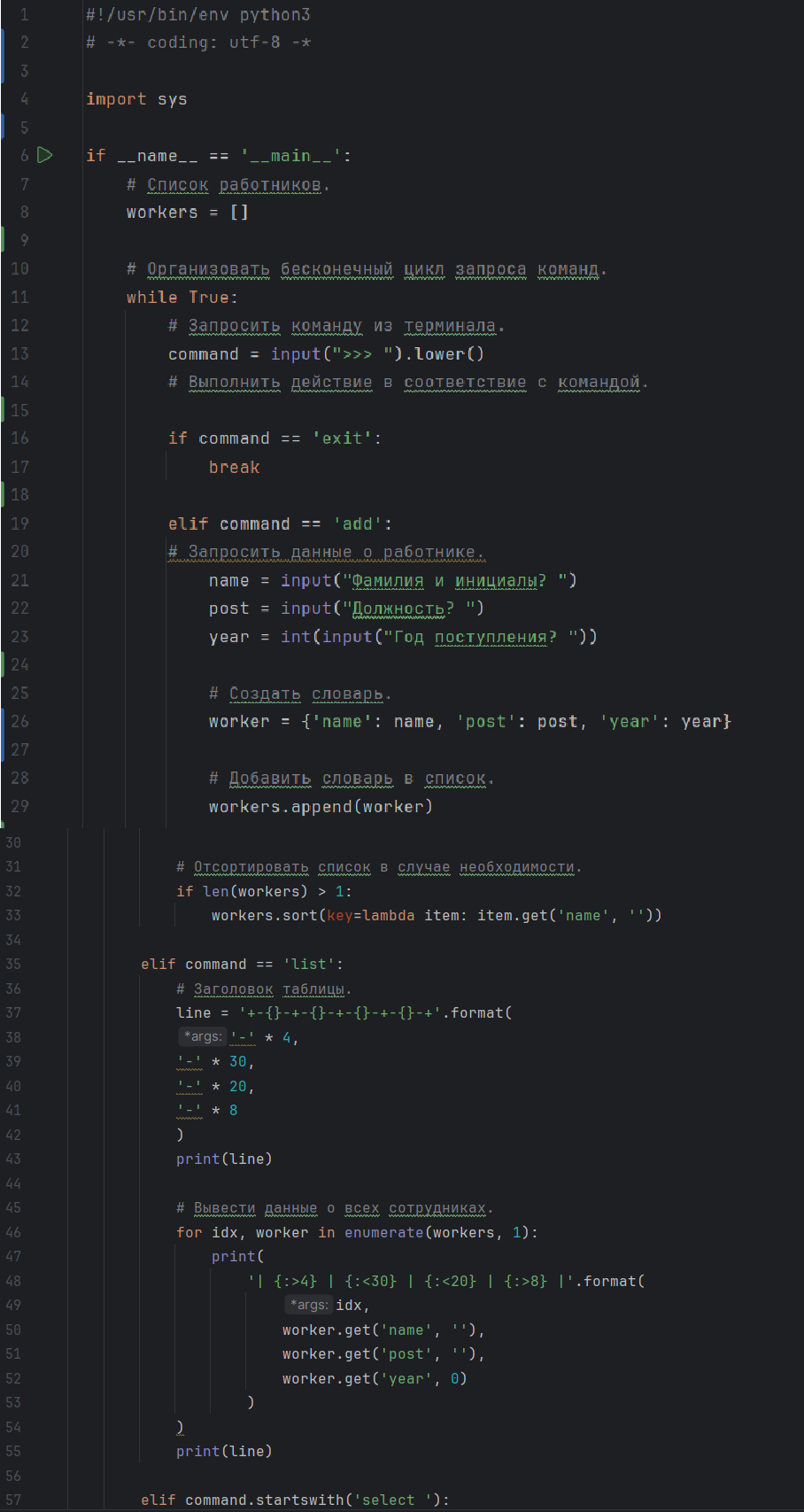


Рисунок 4 – создание ветки develop

1. Проработал примеры из методички.



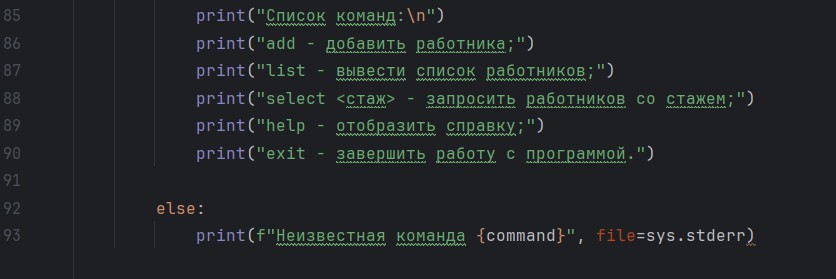
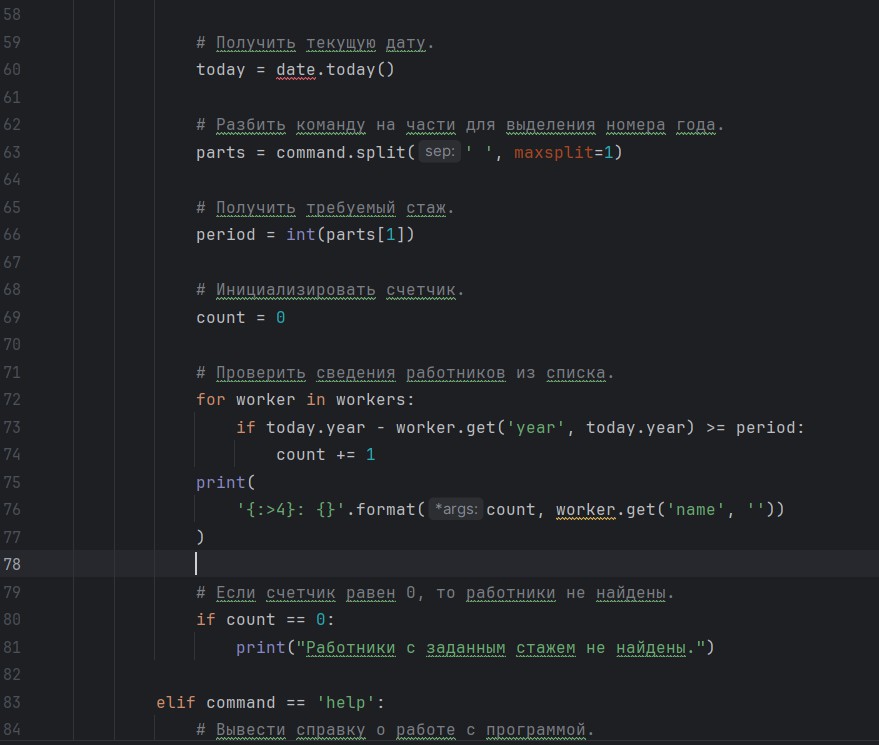


Рисунок 5 – пример 1

Рисунок 6 – пример выполнения примера 1

1. Решите задачу: создайте словарь, связав его с переменной school , и наполните данными, которые бы отражали количество учащихся в разных классах (1а, 1б, 2б, 6а, 7в и т. п.). Внесите изменения в словарь согласно следующему: а) в одном из классов изменилось количество учащихся, б) в школе появился новый класс, с) в школе был расформирован (удален) другой класс. Вычислите общее количество учащихся в школе.

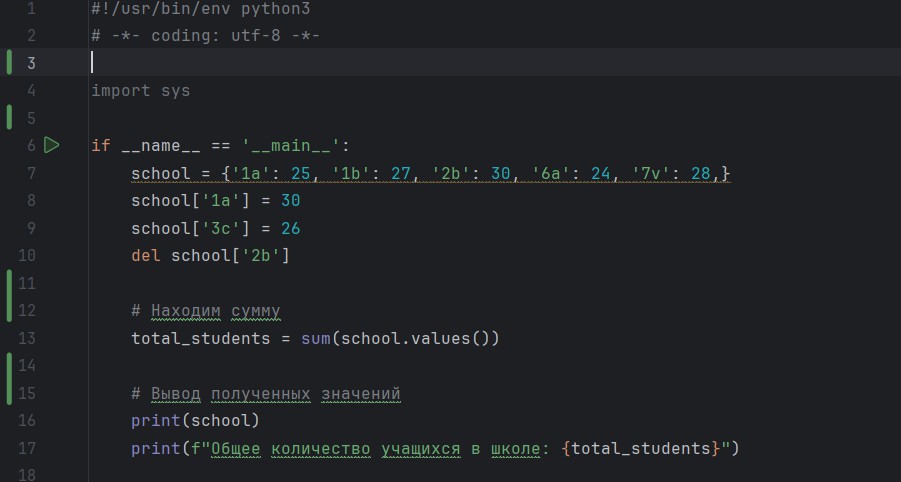


Рисунок 7 – задание 9

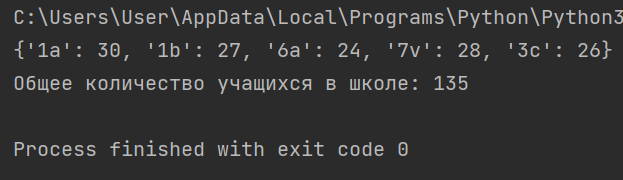


Рисунок 8 – пример выполнения 9 задания

1. Решите задачу: создайте словарь, где ключами являются числа, а значениями – строки. Примените к нему метод items(), c с помощью полученного объекта dict\_items создайте новый словарь, "обратный" исходному, т. е. ключами являются строки, а значениями –числа.

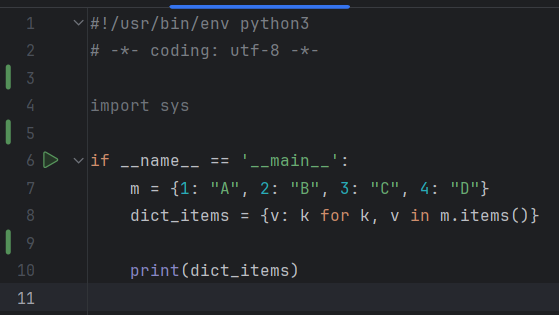


Рисунок 9 – выполнение задания 11

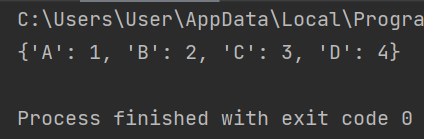


Рисунок 10 – результат выполнения задания 11

1. 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 алфавитном 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.

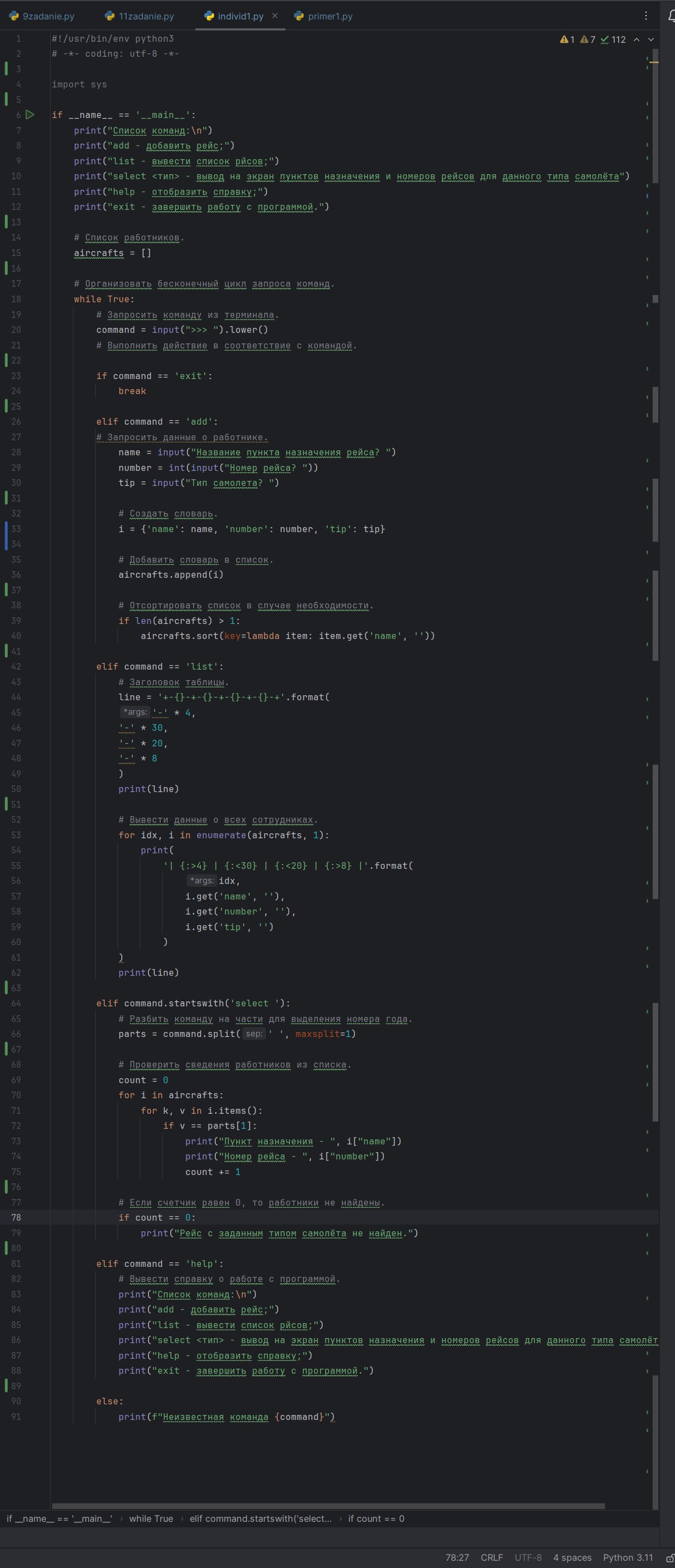


Рисунок 11 – выполнение индивидуального задания

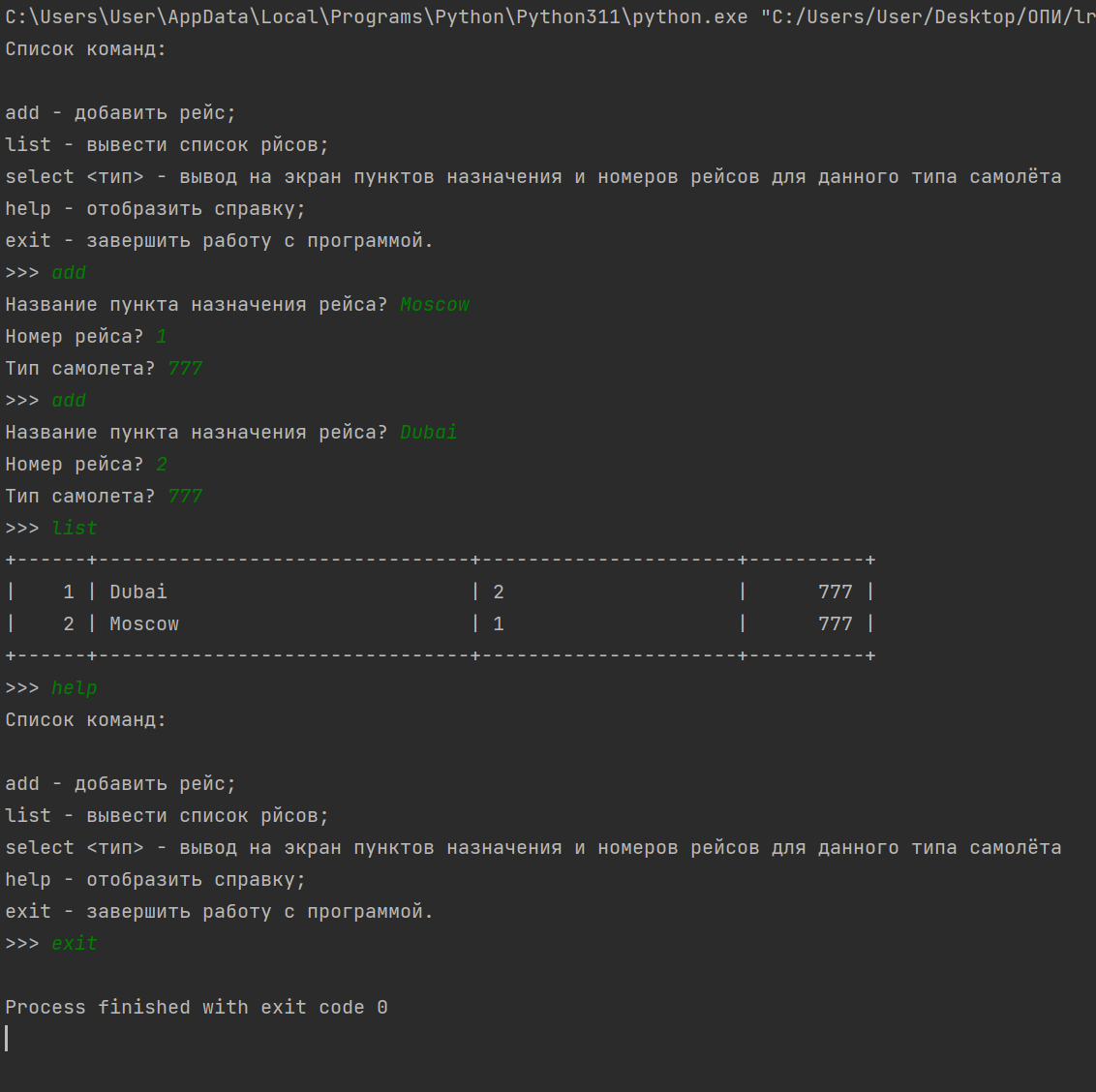


Рисунок 12 – результат выполнения индивидуального задания

9.Зафиксировал все изменения в github в ветке develop.

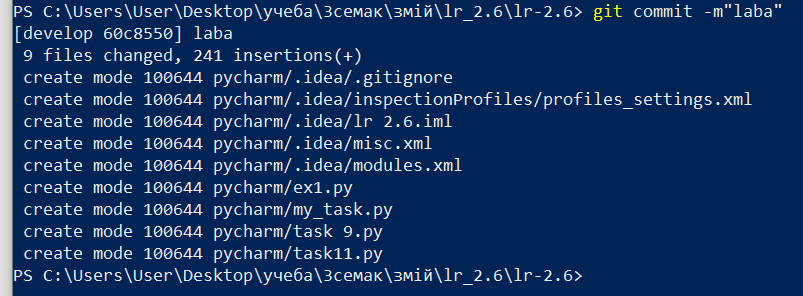


Рисунок 13 – фиксация изменений в ветку develop 10.Слил ветки.

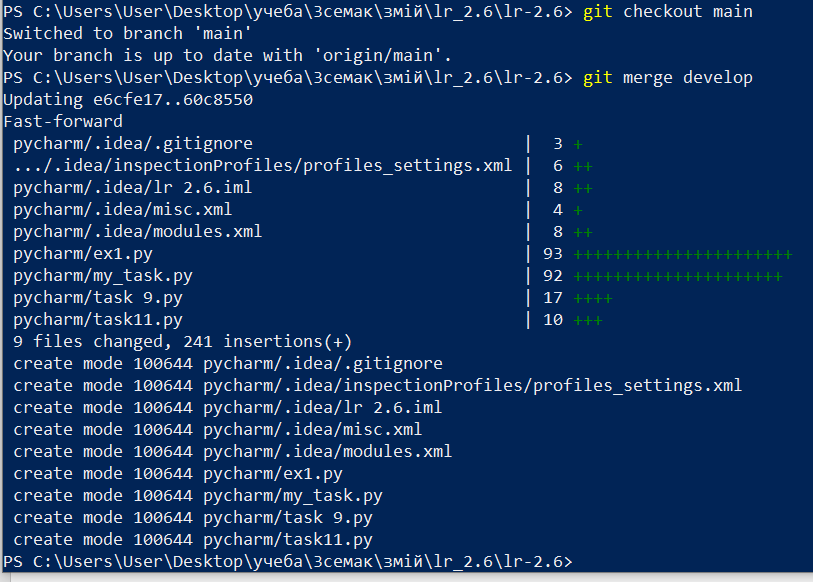


Рисунок 14 – сливание ветки develop в ветку main

# Контрольные вопросы:

1. **Что такое словари в языке Python?**

Словарь ( dict ) представляет собой структуру данных (которая ещё называется ассоциативный массив), предназначенную для хранения произвольных объектов с доступом по ключу. Данные в словаре хранятся в формате ключ – значение.

# Может ли функция len() быть использована при работе со словарями?

Да, функция len() может быть использована для работы со словарями в Python. Она возвращает количество элементов (пар ключ-значение) в словаре.

# Какие методы обхода словарей Вам известны?

* + Цикл for по ключам
  + Использование метода items(), который возвращает пары ключ- значение
  + Обход только ключей с использованием метода keys()
  + Обход только значений с использованием метода values()

# Какими способами можно получить значения из словаря по ключу?

my\_dict = {'a': 1, 'b': 2, 'c': 3} for value in my\_dict.values():

print(value)

# Какими способами можно установить значение в словаре по ключу?

my\_dict = {} my\_dict['ключ'] = 'значение'

# Что такое словарь включений?

Словарь включений аналогичен списковым включениям, за исключением того, что он создаёт объект словаря вместо списка.

{x: x \* x for x in (1, 2, 3, 4)}

{1: 1, 2: 4, 3: 9, 4: 16}

# Самостоятельно изучите возможности функции zip() приведите примеры ее использования.

Функция zip() в Python используется для объединения двух или более итерируемых объектов (списков, кортежей, и т. д.) в один объект, создавая пары значений. Это может быть полезно, когда вам нужно объединить данные из нескольких источников. Вот примеры использования функции zip().

names = ['Анна', 'Петр', 'Мария'] scores = [85, 92, 78]

student\_data = list(zip(names, scores)) print(student\_data)

Результат:

[('Анна', 85), ('Петр', 92), ('Мария', 78)]

# Самостоятельно изучите возможности модуля datetime. Каким функционалом по работе с датой и временем обладает этот модуль?

Модуль datetime в Python предоставляет обширный функционал для работы с датой и временем. Вот некоторые из его основных возможностей

1. Создание объектов даты и времени:
   * datetime.date: Представляет дату (год, месяц, день).
   * datetime.time: Представляет время (час, минута, секунда, микросекунда).
   * datetime.datetime: Представляет комбинацию даты и времени. 2.Получение текущей даты и времени:
   * datetime.datetime.now(): Возвращает текущую дату и время.
2. Разбор и форматирование даты и времени:
   * datetime.datetime.strptime(): Разбор строки в объект datetime.
   * datetime.datetime.strftime(): Преобразование объекта datetime в строку с заданным форматом.
3. Арифметика с датой и временем:
   * Можно выполнять операции сложения и вычитания времени и даты, а также вычислять разницу между двумя моментами времени.
4. Работа с таймзонами:
   * Модуль datetime поддерживает работу с часовыми поясами и таймзонами.
5. Извлечение информации:
   * Можно получать год, месяц, день, часы, минуты, секунды и другую информацию о дате и времени.
6. Выполнение сравнений:
   * Можно сравнивать даты и времена на предмет того, какой из них раньше или позже.
7. Работа с интервалами времени:
   * Модуль datetime поддерживает интервалы времени, которые позволяют выражать продолжительность времени.