

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 РАБОТЕ №2.13
дисциплины «Основы кроссплатформенного программирования»

Выполнил:
Кондратенко Даниил Витальевич
1 курс, группа ИТС-б-о-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. тех. наук, доцент,
доцент кафедры инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: модули и пакеты в языке Python.

Цель работы: приобретение навыков по работе с модулями и пакетами языка программирования Python версии 3.x.

Порядок выполнения работы:

Задание 1.

Изучил теоретический материал работы, создал общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавил файл .gitignore с необходимыми правилами.

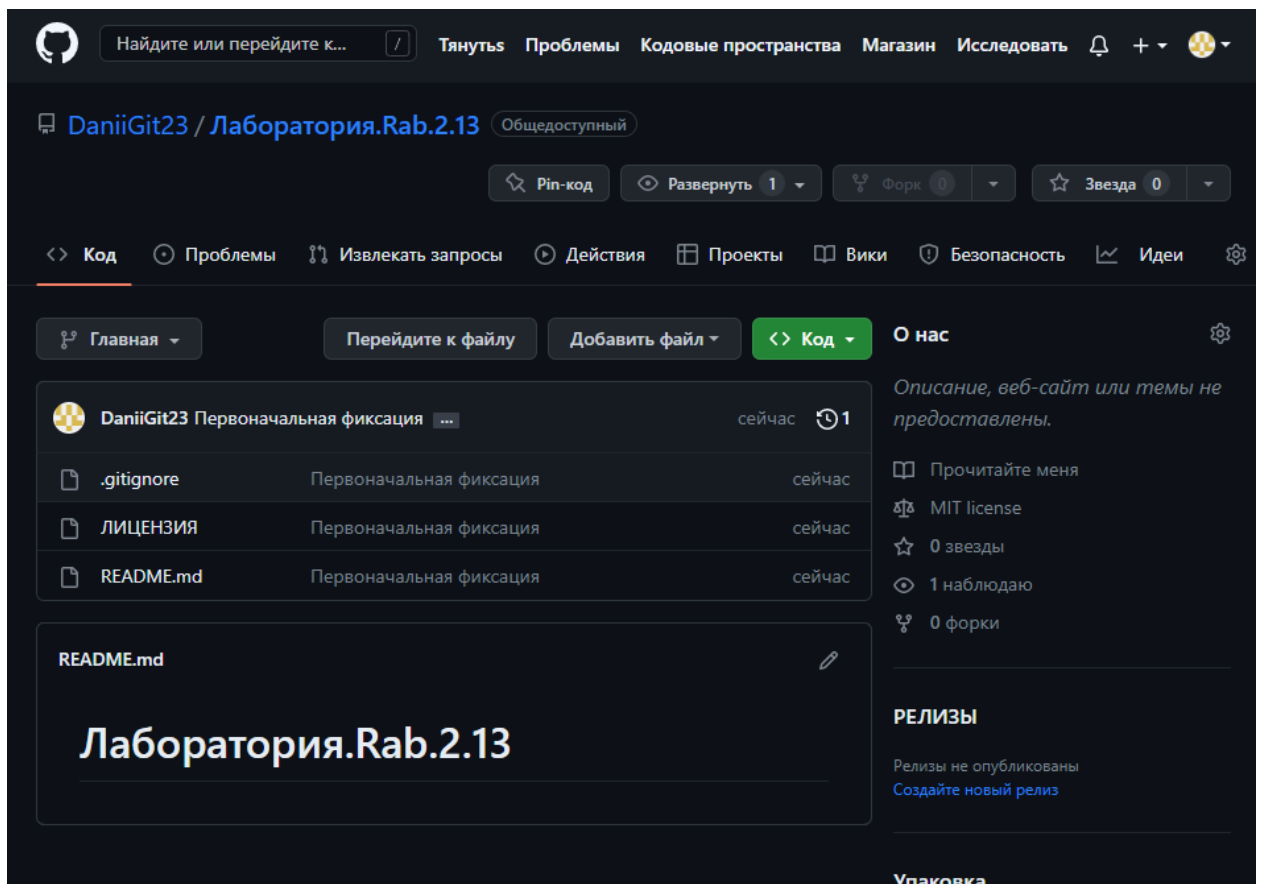


Рисунок 1 – Новый репозиторий

Задание 2.

Проклонировал свой репозиторий на свой компьютер.

Организовал свой репозиторий в соответствие с моделью ветвления git-flow, появилась новая ветка develop.

```
C:\Users\HUAWEI>git clone https://github.com/DaniiGit23/Lab.Rab.2.13.git
Cloning into 'Lab.Rab.2.13'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.

C:\Users\HUAWEI>cd C:\Users\HUAWEI\Lab.Rab.2.13

C:\Users\HUAWEI\Lab.Rab.2.13>git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? [] t
Hooks and filters directory? [C:/Users/HUAWEI/Lab.Rab.2.13/.git/hooks]
```

Рисунок 2 – Клонирование и модель ветвления git-flow

Реализовывал примеры и индивидуальные задания на основе ветки develop, без создания дополнительной ветки feature/(название ветки) по указанию преподавателя.

Задание 3.

Создал проект PyCharm в папке репозитория.

Добавил новый файл *individual1.py*, и файл (модуль) под названием *tagger.py*, который в дальнейшем использовал.

Условие задания: оформить все функции программы в виде отдельного модуля. Разработанный модуль должен быть подключен в основную программу с помощью одного из вариантов команды `import`. Задание: Используя замыкания функций, объявите внутреннюю функцию, которая заключает строку `s` (`s` – строка, параметр внутренней функции) в

произвольный тег, содержащийся в переменной tag – параметре внешней функции. Далее, на вход программы поступает две строки: первая с тегом, вторая с некоторым содержимым. Вторую строку нужно поместить в тег из первой строки с помощью реализованного замыкания. Результат выведите на экран.

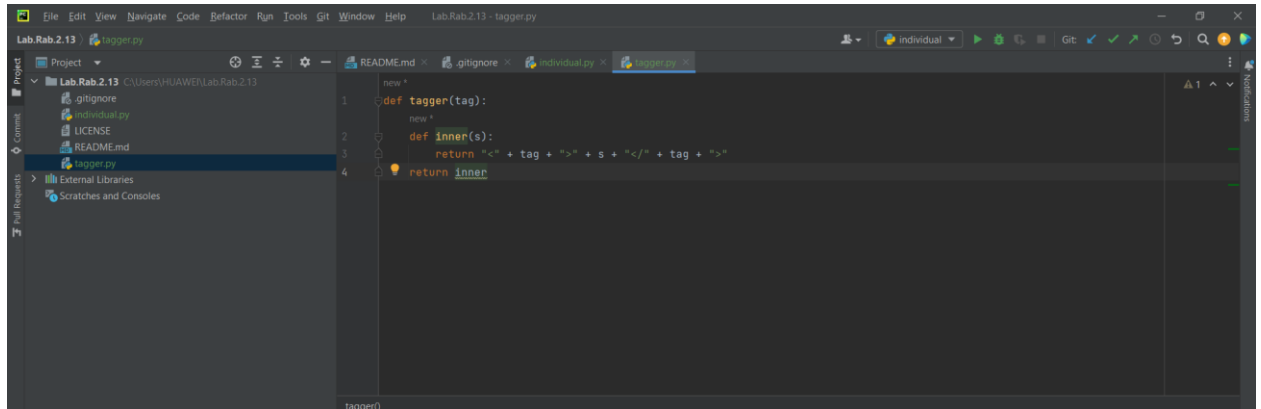


Рисунок 3 – Модуль tagger.py

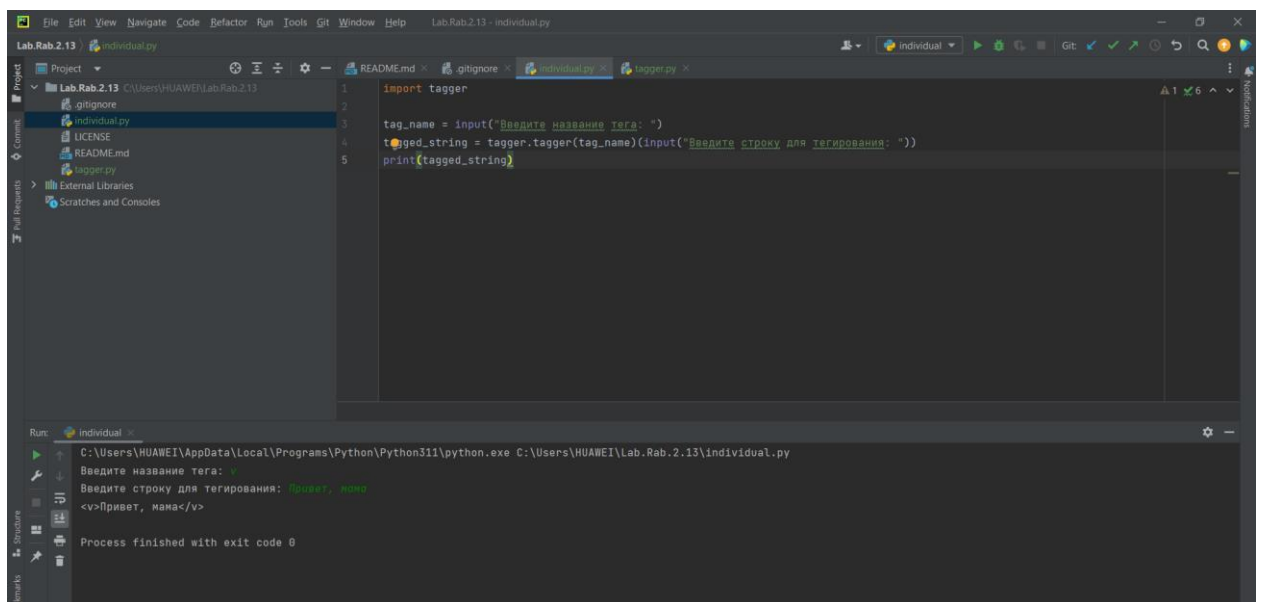


Рисунок 4 – Основной файл с кодом и использование ранее созданного модуля

Зафиксировал данные изменения.

Задание 4.

Выполнение индивидуального задания №2.

Создал целый пакет, который вмещает в себя несколько модулей, которые разбиты на подпрограммы (все названия модулей и пакета соответственно изображены на Рисунке 5), также использовал файл `__init__.py` нужно было определить переменную `all`, которая будет содержать имена всех модулей, которые должны быть импортированы при импорте пакета.

Условие задания: написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть размещены по алфавиту; вывод на экран информации о людях, чьи дни рождения приходятся на месяц, значение которого введено с клавиатуры; если таких нет, выдать на дисплей соответствующее сообщение. Оформить 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.

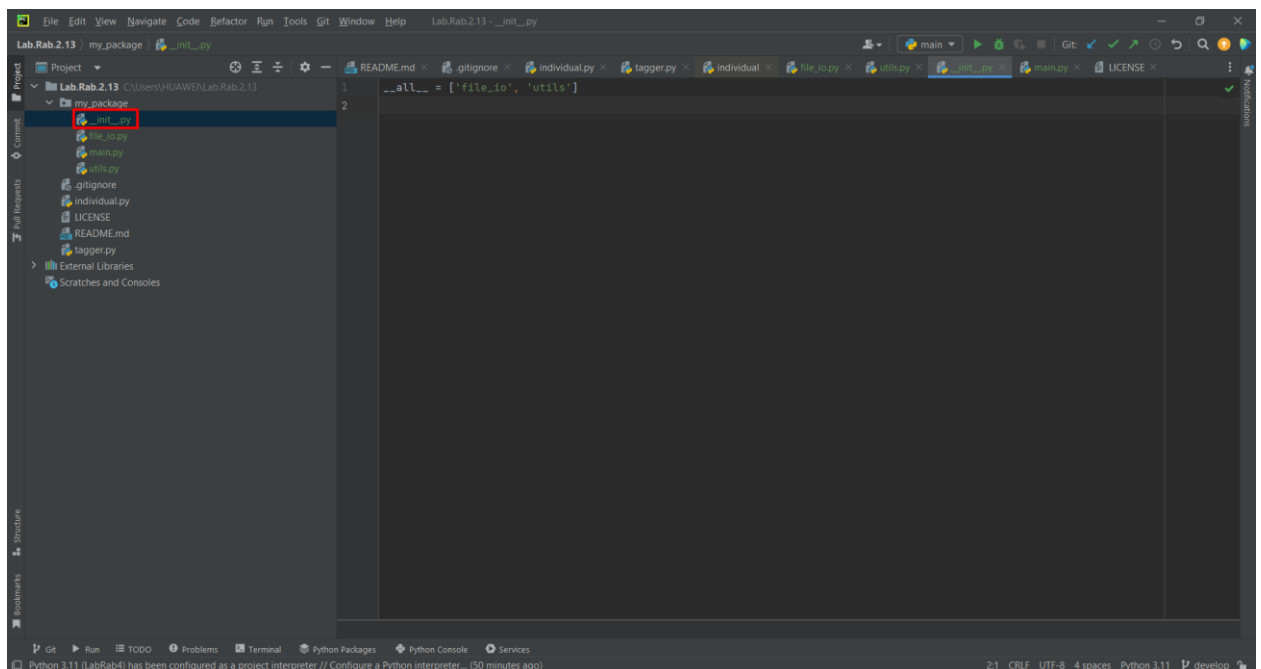


Рисунок 5 – Модуль `__init__.py`

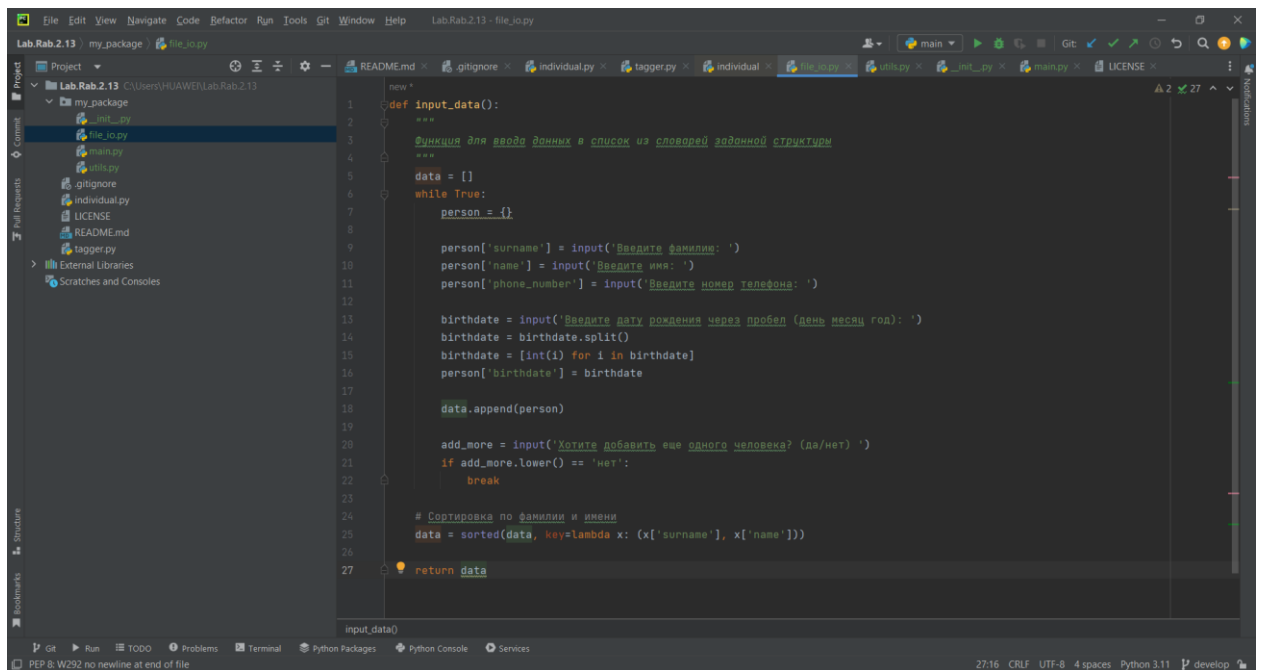


Рисунок 6 – Модуль file_io.py

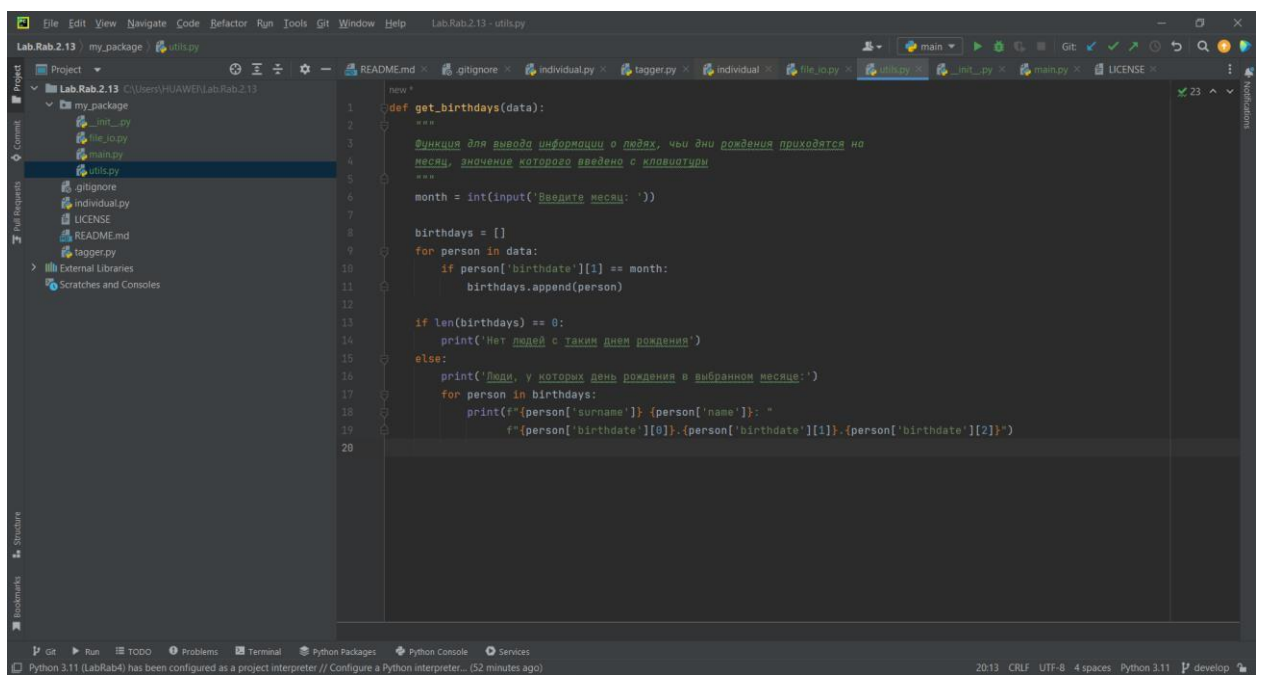


Рисунок 7 – Модуль utils.py

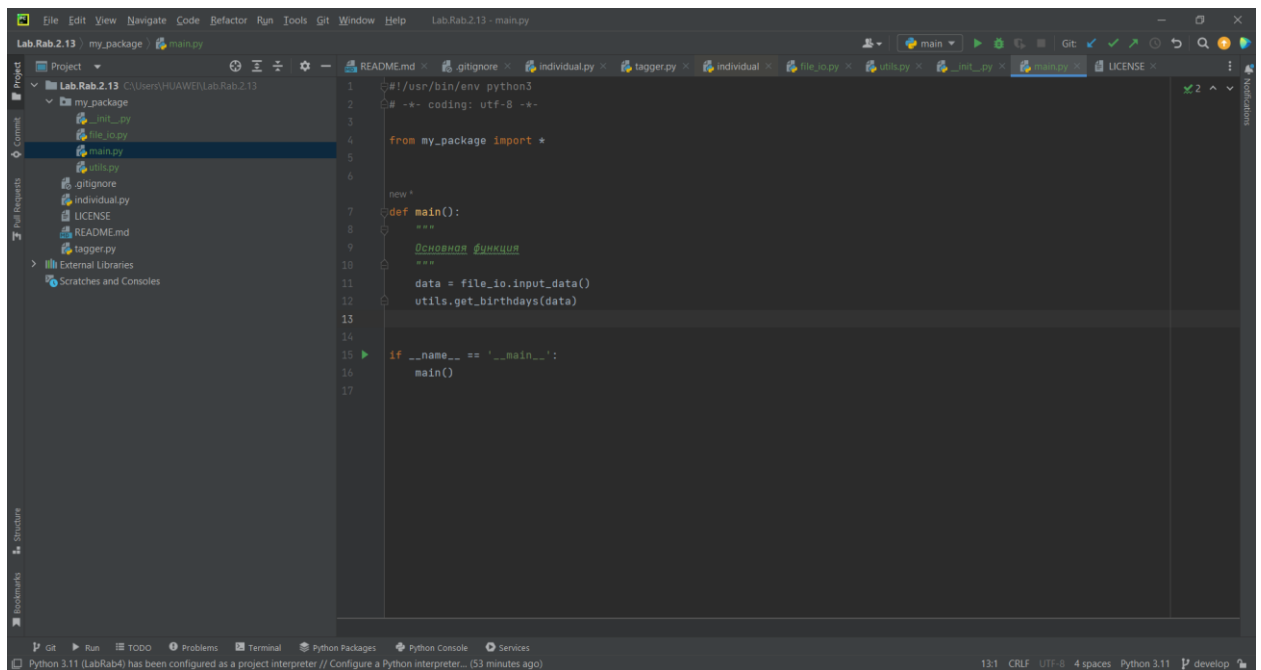


Рисунок 8 – Модуль main.py

При запуске main.py будет выполняться программа.

Результат программы.

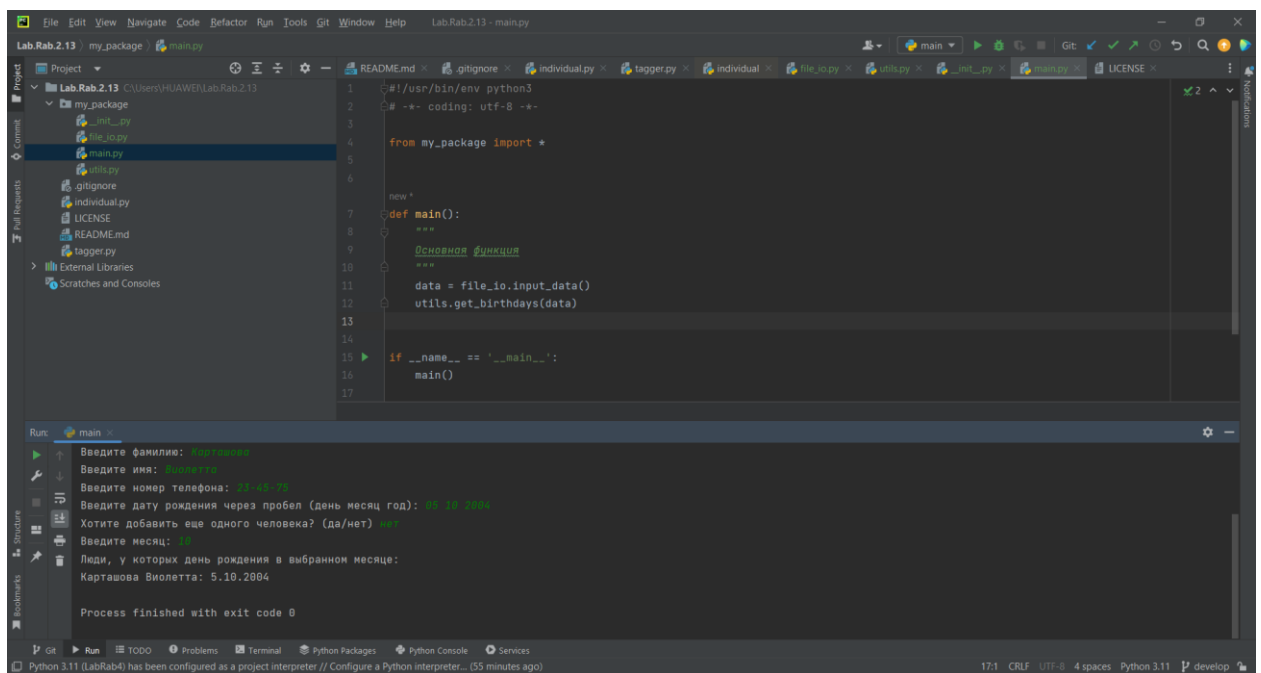


Рисунок 9 – Выполнение программы

Зафиксировал данные изменения.

Задание 5.

Слил ветку develop с веткой main и отправил на удаленный сервер.

```
C:\Users\HUAWEI\Lab.Rab.2.13>git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

C:\Users\HUAWEI\Lab.Rab.2.13>git merge develop
Merge made by the 'ort' strategy.
 individual.py | 5 +++++
 my_package/__init__.py | 1 +
 my_package/file_io.py | 27 ++++++
 my_package/main.py | 16 ++++++
 my_package/utils.py | 19 ++++++
 tagger.py | 4 +++++
 6 files changed, 72 insertions(+)
 create mode 100644 individual.py
 create mode 100644 my_package/__init__.py
 create mode 100644 my_package/file_io.py
 create mode 100644 my_package/main.py
 create mode 100644 my_package/utils.py
 create mode 100644 tagger.py
```

Рисунок 9 – Слияние веток

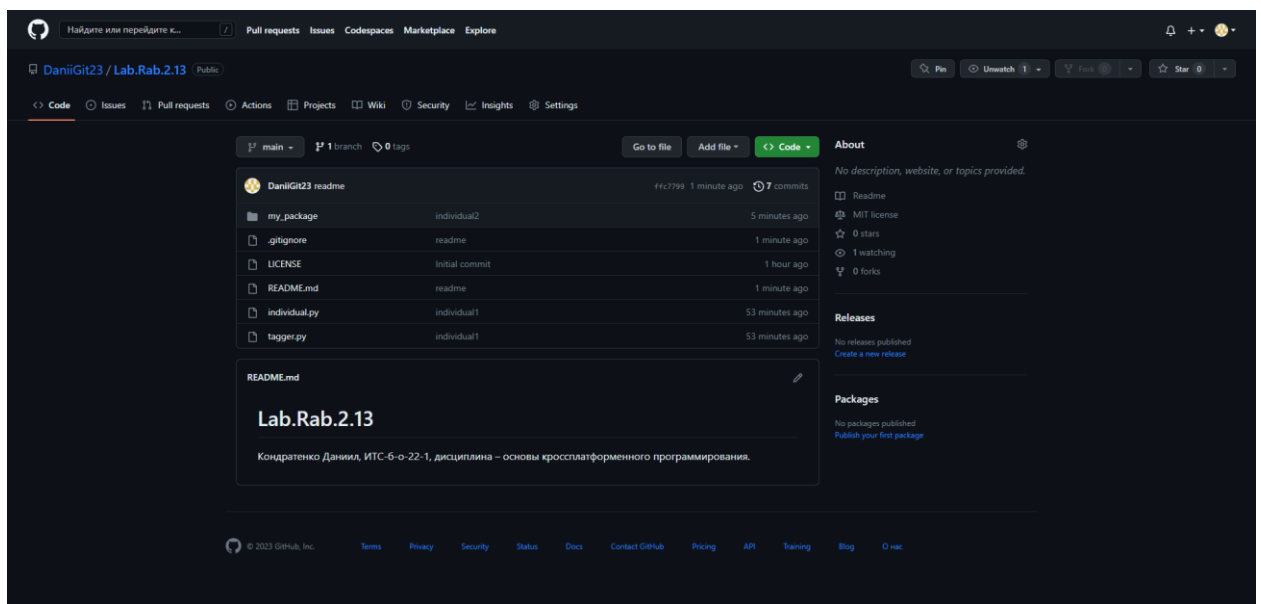


Рисунок 9 – Удаленный сервер

Ссылка на репозиторий: <https://github.com/DaniiGit23/Lab.Rab.2.13.git>

Контрольные вопросы:

1) Что является модулем языка Python?

Модуль в языке программирования Python - это файл с расширением .py, который содержит код, определения функций и переменных, и другие инструкции, которые могут быть импортированы и использованы в других программах. Модуль может также содержать классы и объекты.

2) Какие существуют способы подключения модулей в языке Python?

1. С помощью ключевого слова `import`: `import module_name`

2. С помощью ключевого слова `from`: `from module_name import object_name`

3. С помощью ключевого слова `import` с переименованием: `import module_name as new_name`

4. С помощью импортирования всего модуля: `from module_name import *`

3) Что является пакетом языка Python?

Пакет в Python - это специальная директория с модулями и файлом `__init__.py`, которые могут быть импортированы в других модулях. Пакеты позволяют организовывать код и структурировать большие проекты.

4) Каково назначение файла `__init__.py` ?

Файл `__init__.py` используется для инициализации пакета Python. Когда Python импортирует пакет, он также импортирует код из файла `__init__.py`, если он существует. Это позволяет определять переменные, функции и классы, которые будут доступны внутри пакета. Он также может использоваться для управления импортом модулей внутри пакета и указания того, что должно быть доступно для экспорта при использовании ключевого слова `from package import *`.

5) Каково назначение переменной `__all__` файла `__init__.py` ?

Переменная `__all__` определяет список имен, которые будут импортированы в модуль при импорте с помощью оператора `import *`.

Вывод: в ходе данной лабораторной работы я приобрел навыки по работе с модулями и пакетами языка программирования Python версии 3.x.