МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра

инфокоммуникаций

Институт цифрового

развития

ОТЧЁТ

по лабораторной работе №2.5

Дисциплина: «Основы программной инженерии»

Тема: «Работа с кортежами в языке Python»

Выполнил:

студент 2 курса

группы Пиж-б-о-21-1

Коныжев Максим

Викторович

Ставрополь 2022

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

1. Был создан репозиторий в Github в который были добавлены правила gitignore для работы IDE PyCharm, была выбрана лицензия MIT, сам репозиторий был клонирован на локальный сервер и был организован в соответствии с моделью ветвления git-flow.

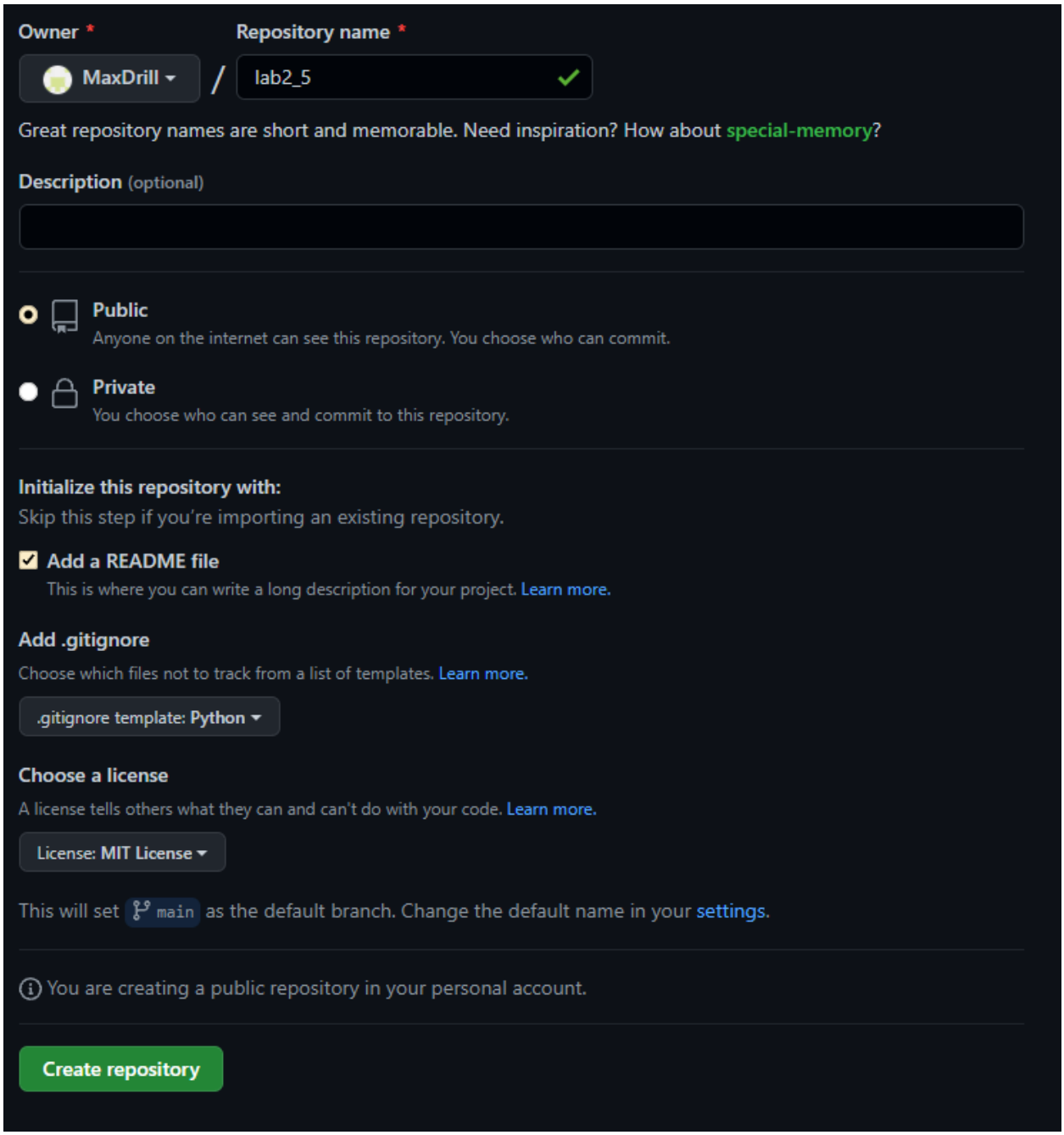

Рисунок 1.1 – Создание репозитория

```
C:\Users\UESR\gitproj>git clone https://github.com/MaxDrill/lab2_5.git
Cloning into 'lab2_5'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
Receiving objects:  60% (3/5)100% (4/4), done.
```

Рисунок 1.2 – Клонирование репозитория

```
C:\Users\UESR\gitproj\lab2_5>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? []
Hooks and filters directory? [C:/Users/UESR/gitproj/lab2_5/.git/hooks]

C:\Users\UESR\gitproj\lab2_5>
```

Рисунок 1.4 – Организация репозитория по модели ветвления git-flow

3. Было выполнено общее задание.

```
3 4 5 2 1 4 3 5 7 9
17

Process finished with exit code 0
```

Рисунок 2.2 – Результат работы первого примера

3. Было выполнено индивидуальное задание согласно 7 варианту.

Дан кортеж целых чисел. Если в нем есть хотя бы одна пара соседних четных чисел, то напечатать все элементы, предшествующие элементам последней из таких пар.

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import sys

if __name__ == '__main__':
    a = tuple(map(int, input().split()))
    k = -1
    for i in range(0, len(a) - 1):
        if a[i] % 2 == 0 and a[i + 1] % 2 == 0:
            k = i
    if k == -1:
        print("Нет таких пар")
    elif k == 0:
        print("До последней пары нет чисел")
    for i in range(0, k):
        print(a[i])
```

```
1 2 3 4 5 6 8 5 6 8
1
2
3
4
5
6
8
5

Process finished with exit code 0
```

Рисунок 3.1 – Результат работы программы

4. Было осуществлен коммит и слияние веток main и develop, также запушены изменения на удаленный сервер.

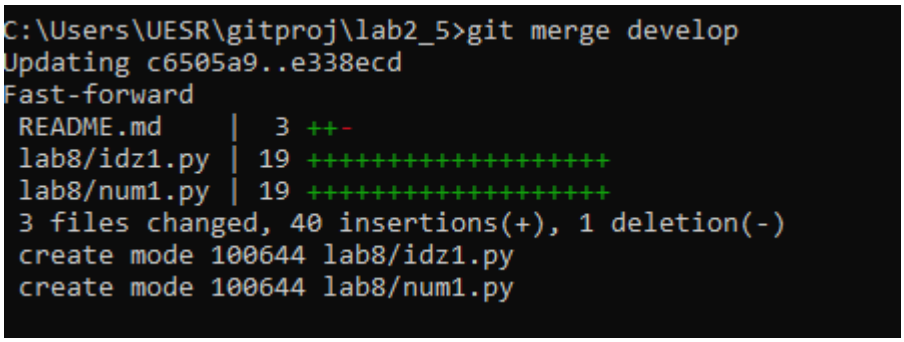

Рисунок 4.1 – Коммит изменений и слияние веток main и develop

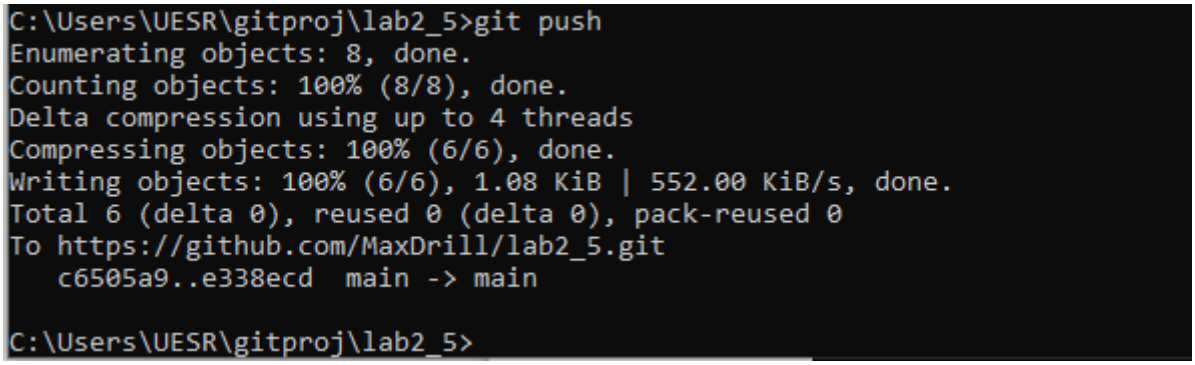

Рисунок 4.2 – Пуш на удаленный сервер

Вывод: были приобретены навыки по работе с кортежами при написании программ с помощью языка программирования Python версии 3.x.

Контрольные вопросы

1. Что такое кортежи в языке Python?

Кортеж (tuple) – это неизменяемая структура данных, которая по своему подобию очень похожа на список.

2. Каково назначение кортежей в языке Python?

Существует несколько причин, по которым стоит использовать

кортежи вместо списков. Одна из них – это обезопасить данные от случайного изменения. Если мы получили откуда-то массив данных, и у нас есть желание поработать с ним, но при этом непосредственно менять данные мы не собираемся, тогда, это как раз тот случай, когда кортежи придутся как нельзя кстати. Кортежи в памяти занимают меньший объем по сравнению со списками. Кортежи работают быстрее, чем списки

3. Как осуществляется создание кортежей?

a = ()

b = tuple()

4. Как осуществляется доступ к элементам кортежа?

Доступ к элементам кортежа осуществляется также как к элементам списка – через указание индекса.

5. Зачем нужна распаковка (деструктуризация) кортежа?

Обращение по индексу, это не самый удобный способ работы с кортежами. Дело в том, что кортежи часто содержат значения разных типов, и помнить, по какому индексу что лежит — очень непросто.

6. Какую роль играют кортежи в множественном присваивании?

Используя множественное присваивание, можно провернуть интересный трюк: обмен значениями между двумя переменными.

7. Как выбрать элементы кортежа с помощью среза?

С помощью операции взятия среза можно получить другой кортеж.

Общая форма операции взятия среза для кортежа следующая T2 = T1[i:j]

здесь

- T2 – новый кортеж, который получается из кортежа T1;

- T1 – исходный кортеж, для которого происходит срез;
- i, j – соответственно нижняя и верхняя границы среза. Фактически берутся ко вниманию элементы, лежащие на позициях i, i+1, …, j-1. Значение j определяет позицию за последним элементом среза.

8. Как выполняется конкатенация и повторение кортежей?

Для кортежей можно выполнять операцию конкатенации, которая обозначается символом +.

T3 = T1 + T2

9. Как выполняется обход элементов кортежа?

Элементы кортежа можно последовательно просмотреть с помощью операторов цикла while или for.

10. Как проверить принадлежность элемента кортежу?

Проверка вхождения элемента в кортеж - оператор in.

11. Какие методы работы с кортежами Вам известны?

index(), count().

12. Допустимо ли использование функций агрегации таких как len(), sum() и т. д. при работе с кортежами?

Доступно.

13. Как создать кортеж с помощью спискового включения.

Так же, как и список.