

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 №2.5
дисциплины «Основы кроссплатформенного программирования»

Выполнил:
Ермолович Иван Денисович
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: работа с кортежами в языке Python.

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

Порядок выполнения работы:

1. Создал новый репозиторий и клонировал его на свой компьютер

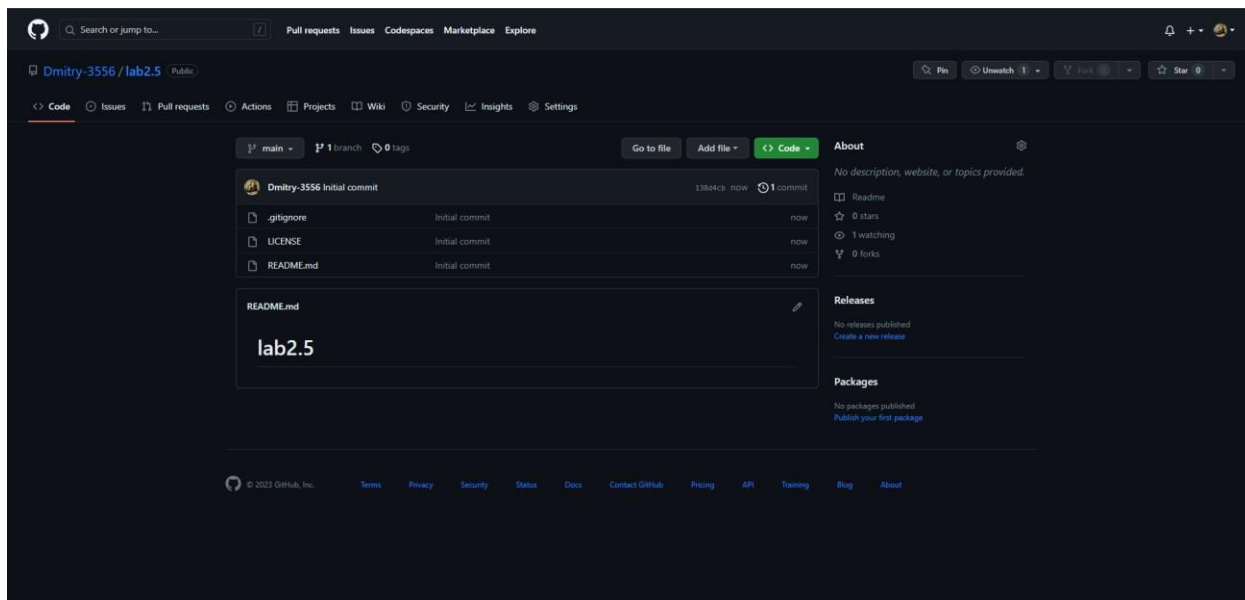


Рисунок 1 – Новый репозиторий

2. Проклонировал репозиторий на свой компьютер. В ходе данной лабораторной работы работал с моделью ветвления git-flow.

```
C:\Work>git clone https://github.com/Dmitry-3556/lab2.5.git
Cloning into 'lab2.5'...
remote: Enumerating objects: 5, done.
Receiving objects: 100% (5/5), 2.74 KiB | 2.74 MiB/s, done.s: 20% (1/5)

remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0

C:\Work\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:/Work/lab2.5/.git/hooks]

C:\Work\lab2.5>
C:\Work\lab2.5>git status
On branch develop
nothing to commit, working tree clean

C:\Work\lab2.5>
```

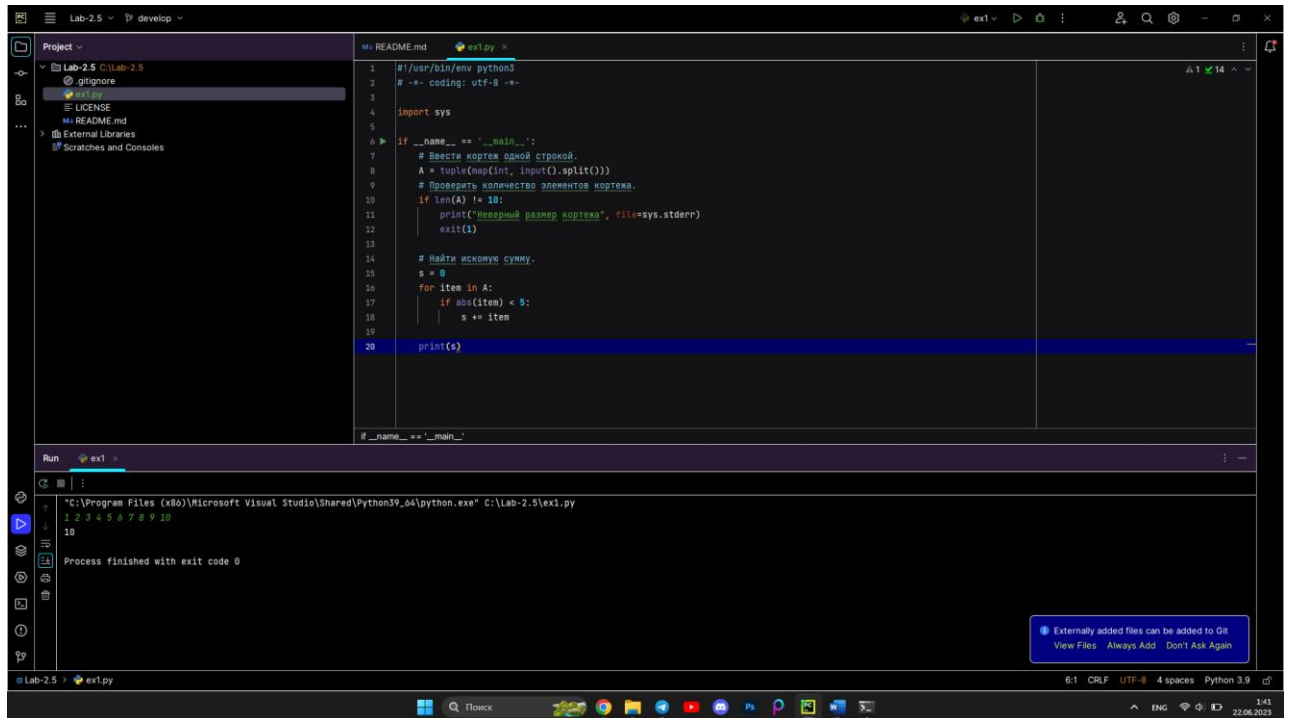
Рисунок 2 – Клонирование и модель ветвления git-flow

Реализовывал примеры и индивидуальные задания на основе ветки develop.

3. Пример №1.

Добавил новый файл

Условие примера: ввести кортеж A из 10 элементов, найти сумму элементов, меньших по модулю 5, и вывести ее на экран. Использовать в программе вместо списков кортежи.



The screenshot shows the Visual Studio Code interface with a Python file named `ex1.py` open. The code in the editor is as follows:

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5
6 if __name__ == '__main__':
7     # Ввести кортеж одной строкой.
8     A = tuple(map(int, input().split()))
9     # Проверить количество элементов кортежа.
10    if len(A) != 10:
11        print("Неверный размер кортежа", file=sys.stderr)
12        exit(1)
13
14    # Найти искомую сумму.
15    s = 0
16    for item in A:
17        if abs(item) < 5:
18            s += item
19
20    print(s)
21
22 if __name__ == '__main__':
```

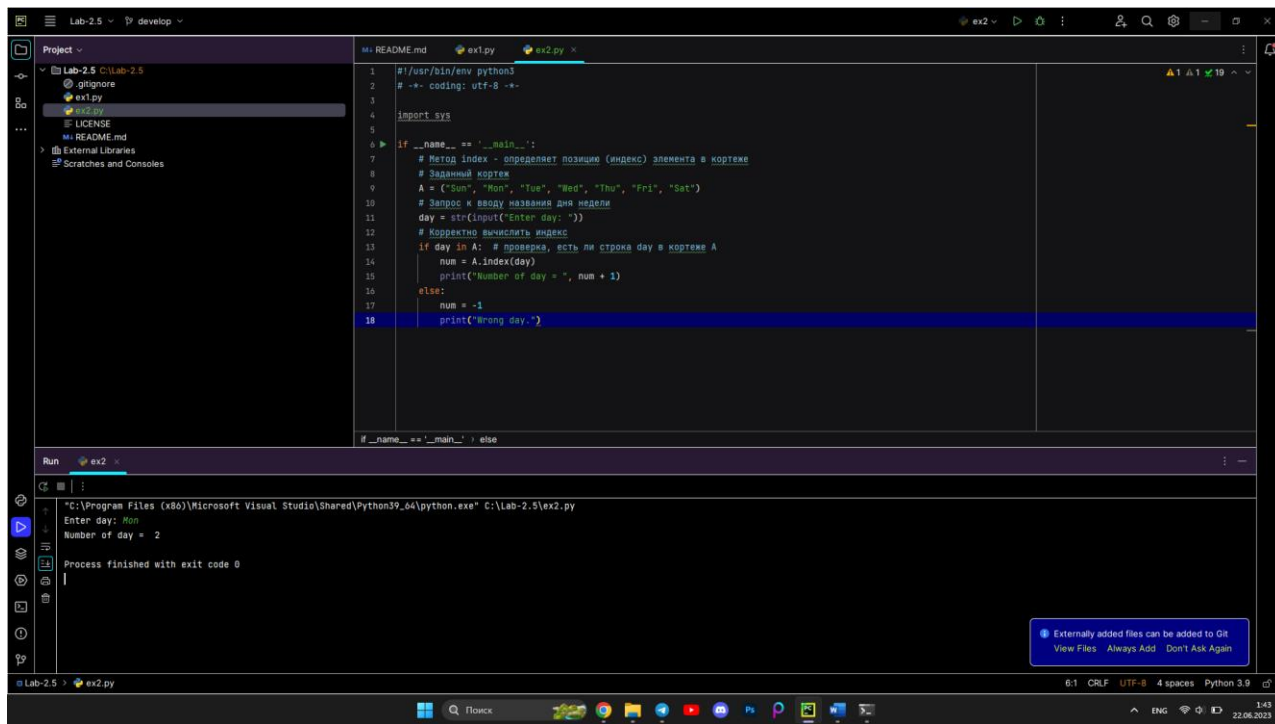
The Run and Debug console at the bottom shows the execution of the script. The input provided is `2 2 3 4 5 6 7 8 9 10`, and the output is `10`. The console also indicates that the process finished with exit code 0.

Рисунок 3 – Реализация первого примера

4. Создал новый файл под названием *ex2.py*

Пример№2.

Условие примера: в перечне названий дней недели вычисляется порядковый номер дня.



The screenshot shows the Visual Studio Code interface with a Python file named *ex2.py* open. The code defines a tuple of days of the week and uses the `index()` method to find the position of a user-input day. The output window shows the program running successfully with the input 'Mon' resulting in 'Number of day = 2'.

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5
6
7 if __name__ == '__main__':
8     # Метод index - определяет позицию (индекс) элемента в кортеже
9     # Заданный кортеж
10    A = ("Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat")
11    # Запрос к вводу названия дня недели
12    day = str(input("Enter day: "))
13    # Корректно вычислить индекс
14    if day in A: # проверка, есть ли строка day в кортеже A
15        num = A.index(day)
16        print("Number of day = ", num + 1)
17    else:
18        num = -1
19        print("Wrong day.")
20
21
22 if __name__ == '__main__': else
```

Run ex2

"C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python9_04\python.exe" C:\Lab-2.5\ex2.py

Enter day: Mon

Number of day = 2

Process finished with exit code 0

Рисунок 4 – Реализация второго примера

5. Индивидуальное задание №1.

Создал новый файл под названием

Имеются данные о сумме очков, набранных в чемпионате каждой из футбольных команд. Определить, перечислены ли команды в списке в соответствии с занятыми ими местами в чемпионате.

Вариант 11.

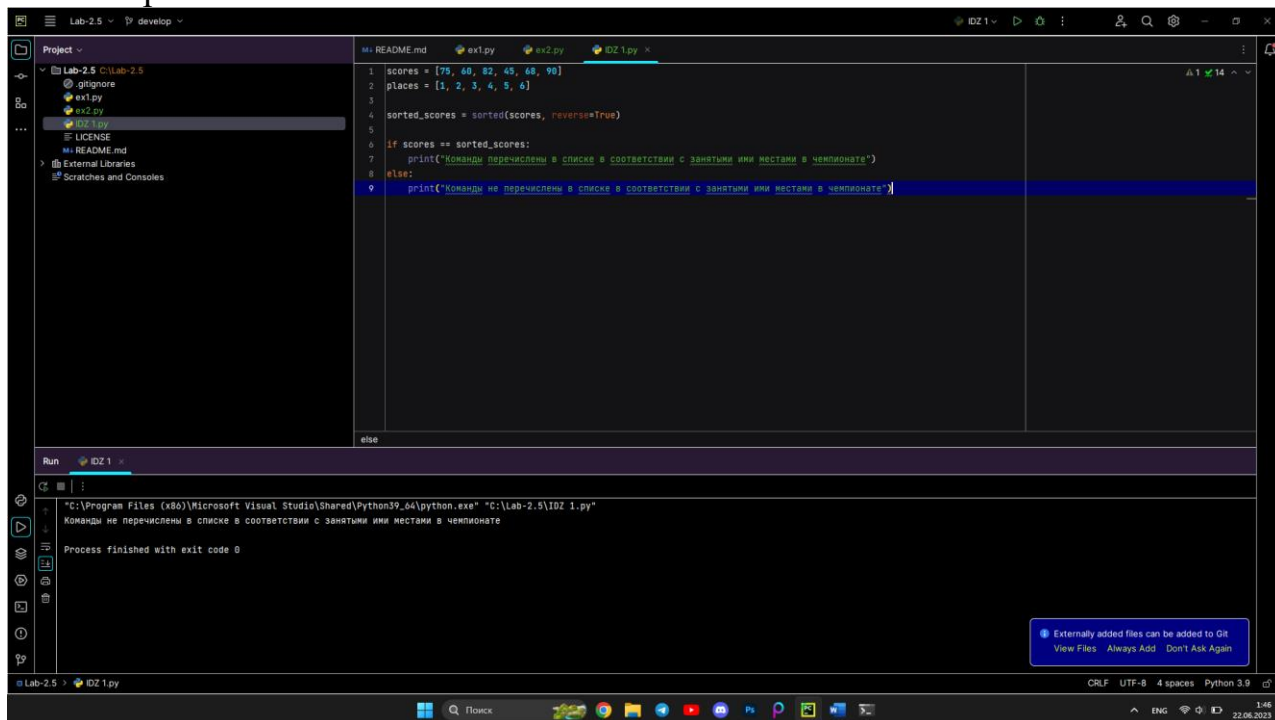
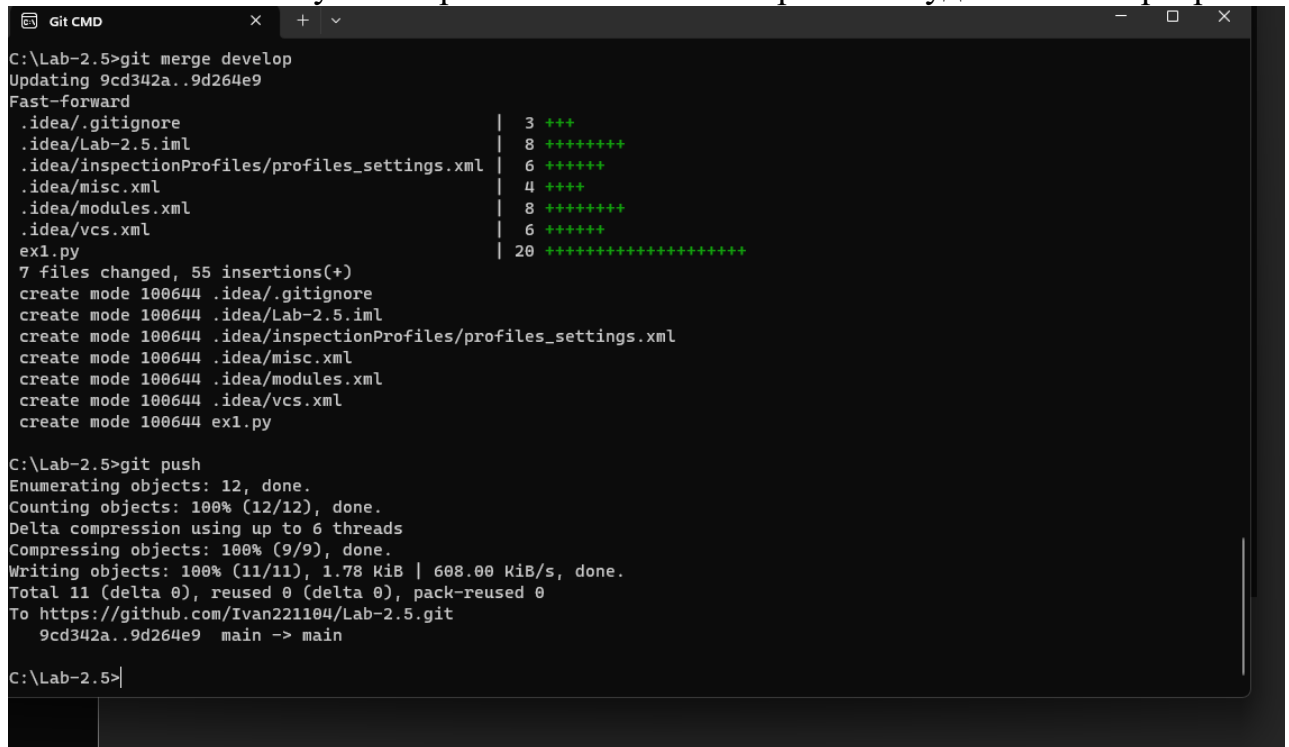


Рисунок 5 – Программа и ее результат

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



```
Git CMD
C:\Lab-2.5>git merge develop
Updating 9cd342a..9d264e9
Fast-forward
 .idea/.gitignore          | 3 +++
 .idea/Lab-2.5.iml         | 8 ++++++
 .idea/inspectionProfiles/profiles_settings.xml | 6 +++++
 .idea/misc.xml            | 4 ++++
 .idea/modules.xml         | 8 ++++++
 .idea/vcs.xml             | 6 +++++
 ex1.py                    | 20 ++++++
7 files changed, 55 insertions(+)
create mode 100644 .idea/.gitignore
create mode 100644 .idea/Lab-2.5.iml
create mode 100644 .idea/inspectionProfiles/profiles_settings.xml
create mode 100644 .idea/misc.xml
create mode 100644 .idea/modules.xml
create mode 100644 .idea/vcs.xml
create mode 100644 ex1.py
C:\Lab-2.5>git push
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 6 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (11/11), 1.78 KiB | 608.00 KiB/s, done.
Total 11 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Ivan221104/Lab-2.5.git
 9cd342a..9d264e9  main -> main
C:\Lab-2.5>
```

Рисунок 6 – Слияние веток develop в main

Ответы на контрольные вопросы:

1. Что такое списки в языке Python?

Списки - это коллекции указателей на элементы. В Python не обязательно, чтобы все элементы списка были одного типа

2. Каково назначение кортежей в языке Python?

Кортеж в Python — это более быстрый и неизменяемый аналог списка. Он очень часто используется для защиты хранимых данных приложения от незапланированных или непреднамеренных изменений.

3. Как осуществляется создание кортежей?

Кортеж создается путем помещения всех элементов (элементов) в круглые скобки (), разделенных запятыми.

4. Как осуществляется доступ к элементам кортежа?

Доступ к элементам кортежа осуществляется также как к элементам списка – через указание индекса. Но, как уже было сказано – изменять элементы кортежа нельзя!

5. Зачем нужна распаковка (деструктуризация) кортежа?

Часто кортежи содержат значения разных типов. Сложно запомнить, каким индексом обозначается каждое значение. Чтобы упростить работу, можно разобрать кортеж.

6. Какую роль играют кортежи в множественном присваивании?

Кортежи в Python играют важную роль в множественном присваивании. Множественное присваивание позволяет присваивать значения нескольким переменным за один раз. Кортежи могут быть использованы для присваивания нескольким переменным значений за один раз.

7. Как выбрать элементы кортежа с помощью среза?

Элементы кортежа можно выбрать с помощью среза, указав индексы начала и конца нужного диапазона через двоеточие в квадратных скобках.

8. Как выполняется конкатенация и повторение кортежей?

Конкатенация (объединение) кортежей в Python выполняется с помощью оператора "+". Для повторения кортежа используется оператор "*".

9. Как выполняется обход элементов кортежа?

Обход элементов кортежа может быть выполнен с помощью цикла for.

10. Как проверить принадлежность элемента кортежу?

Для проверки нахождения элемента в кортеже можно использовать оператор in.

11. Какие методы работы с кортежами Вам известны?

1. count(x) - возвращает количество вхождений элемента x в кортеж.
2. index(x) - возвращает индекс первого вхождения элемента x в кортеже.
3. len(t) - возвращает длину кортежа t.

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

Да, использование функций агрегации, таких как `len()`, `sum()`, `min()`, `max()`, и т.д. допустимо при работе с кортежами, так как они позволяют получать информацию о элементах кортежа и выполнять различные математические операции над ними.

13. Как создать кортеж с помощью спискового включения.

Кортеж можно создать с помощью спискового включения с использованием круглых скобок.

Вывод: в ходе данной лабораторной работы, я приобрел навыки по работе с кортежами при написании программ с помощью языка программирования Python версии 3.x.