

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:

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: 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:

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.

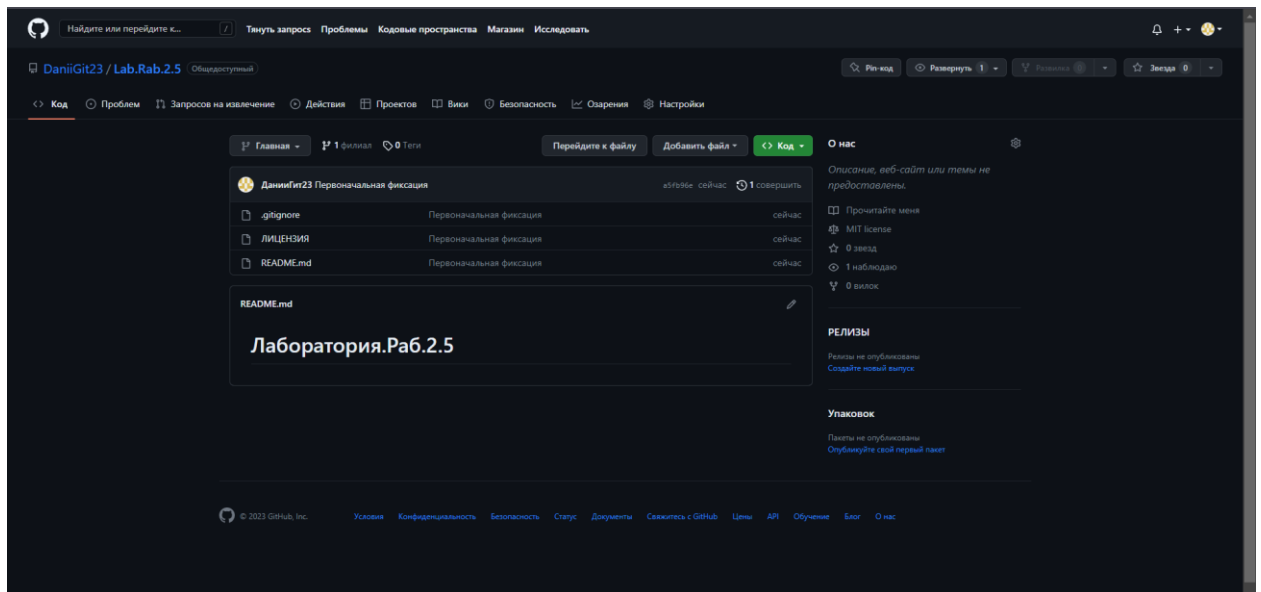


Рисунок 1 – Новый репозиторий

Задание 2.

Проклонировал свой репозиторий на свой компьютер.

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

```
C:\Users\HUAWEI>git config --global user.name "Daniil"
C:\Users\HUAWEI>git config --global user.email "kondratenko_danil.23@mail.ru"
C:\Users\HUAWEI>git clone https://github.com/DaniiGit23/Lab.Rab.2.5.git
Cloning into 'Lab.Rab.2.5'...
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.2.5
C:\Users\HUAWEI\Lab.Rab.2.5>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? [] t
Hooks and filters directory? [C:/Users/HUAWEI/Lab.Rab.2.5/.git/hooks]
C:\Users\HUAWEI\Lab.Rab.2.5>
```

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.

Работа с примером №1.

Добавил новый файл *primer1.py*.

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

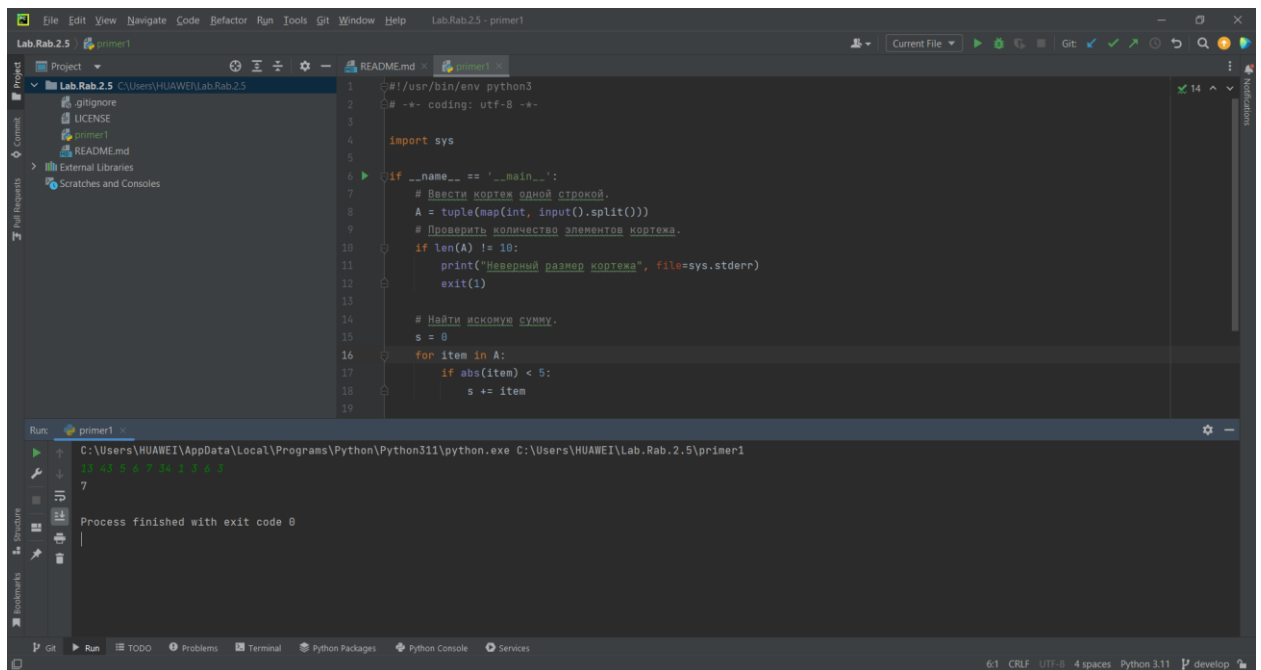


Рисунок 3 – Проект PyCharm и реализация первого примера

Задание 4.

Создал новый файл по названию *primer2.py*

Работа с примером №2.

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

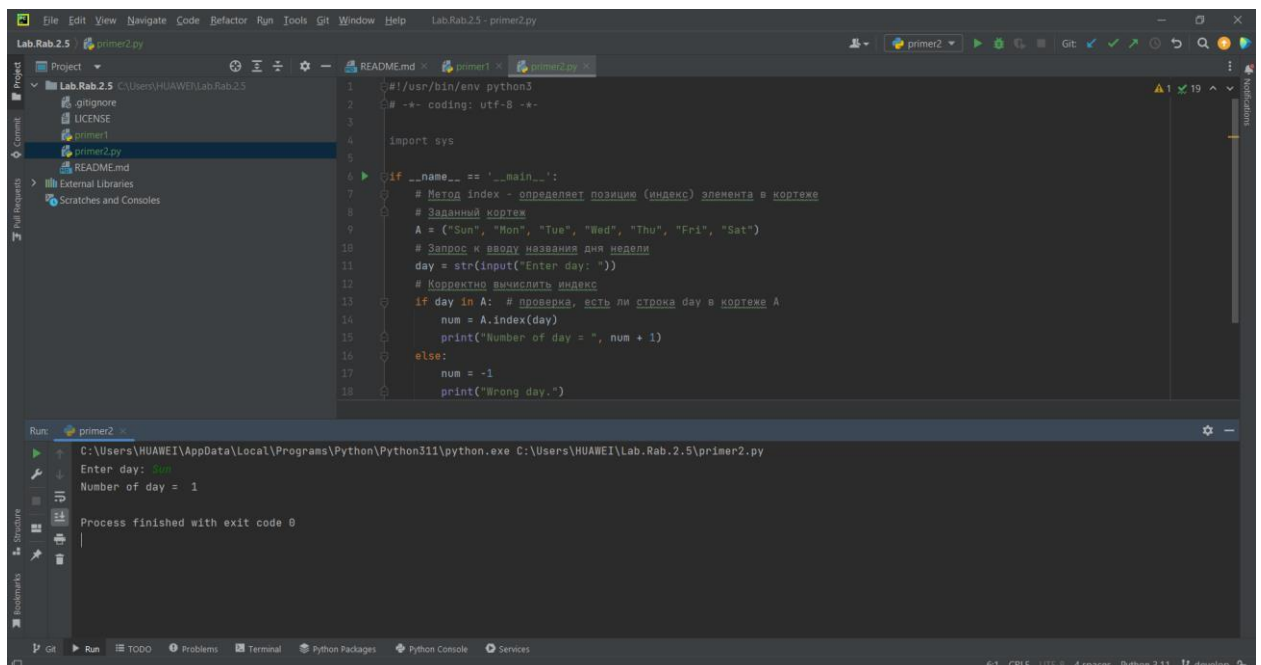


Рисунок 4 – Реализация второго примера

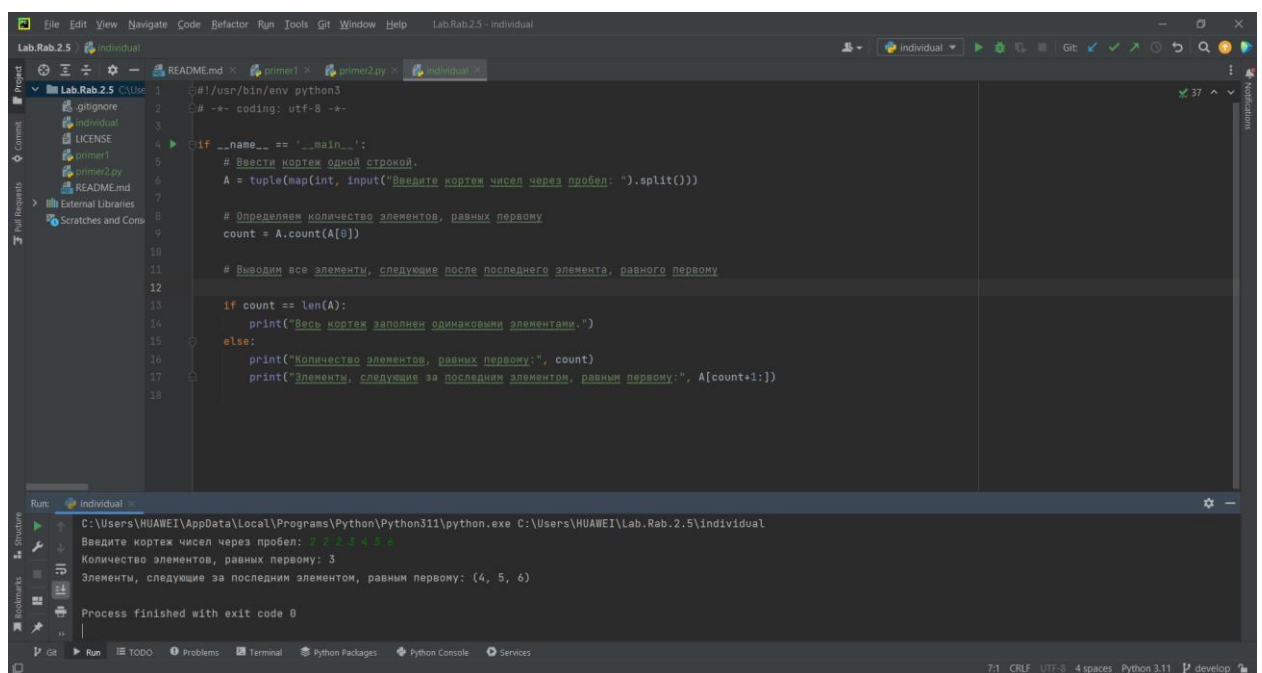
Задание 5.

Выполнение индивидуального задание №1.

Создал новый файл под названием *individual.py*

Вариант 12 (по списку группы).

Условие задания: в начале кортежа записано несколько равных между собой элементов. Определить количество таких элементов и вывести все элементы, следующие за последним из них. Рассмотреть возможность того, что весь массив заполнен одинаковыми элементами. Условный оператор не использовать.



The screenshot shows a code editor with a file named `individual.py`. The code is as follows:

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == '__main__':
5     # Ввести кортеж одной строкой.
6     A = tuple(map(int, input("Введите кортеж чисел через пробел: ").split()))
7
8     # Определяем количество элементов, равных первому
9     count = A.count(A[0])
10
11     # Выводим все элементы, следующие после последнего элемента, равного первому
12
13     if count == len(A):
14         print("Весь кортеж заполнен одинаковыми элементами.")
15     else:
16         print("Количество элементов, равных первому:", count)
17         print("Элементы, следующие за последним элементом, равным первому:", A[count+1:])
18
```

The Run window shows the following output:

```
C:\Users\HUAWEI\AppData\Local\Programs\Python\Python311\python.exe C:\Users\HUAWEI\Lab.Rab.2.5\individual
Введите кортеж чисел через пробел: 2 2 2 3 4 5 6
Количество элементов, равных первому: 3
Элементы, следующие за последним элементом, равным первому: (4, 5, 6)
Process finished with exit code 0
```

Рисунок 5 – Программа и ее результат

Задание 6.

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

```
C:\Users\HUAWEI\Lab.Rab.2.5>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

C:\Users\HUAWEI\Lab.Rab.2.5>git merge develop
Updating a5fb96e..6572ce5
Fast-forward
 .gitignore                | 4 ++++
 .idea/.gitignore           | 3 +++
 .idea/Lab.Rab.2.5.iml      | 8 ++++++++
 .idea/inspectionProfiles/profiles_settings.xml | 6 ++++++
 .idea/misc.xml             | 4 ++++
 .idea/modules.xml         | 8 ++++++++
 .idea/vcs.xml              | 6 ++++++
 README.md                 | 4 +++-
 individual                 | 17 +++++++++++++++++++++
 primer1                    | 20 +++++++++++++++++++++
 primer2.py                 | 18 +++++++++++++++++++++
11 files changed, 97 insertions(+), 1 deletion(-)
create mode 100644 .idea/.gitignore
create mode 100644 .idea/Lab.Rab.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 individual
create mode 100644 primer1
create mode 100644 primer2.py
C:\Users\HUAWEI\Lab.Rab.2.5>git |
```

Рисунок 5 – Слияние веток develop в main

Отправка изменений на удаленный сервер.

Ссылка на репозиторий: https://github.com/DaniiGit23/Lab.Rab.2_5.git

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.