

**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:
Сластёнов Андрей Сергеевич
1 курс, группа ИТС-б-о-21-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль) Сети связи и
системы коммутации,
очная форма обучения

(подпись)

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

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г

Тема: работа со строками в языке Python.

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

Порядок выполнения работы:

1. Создал общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT.

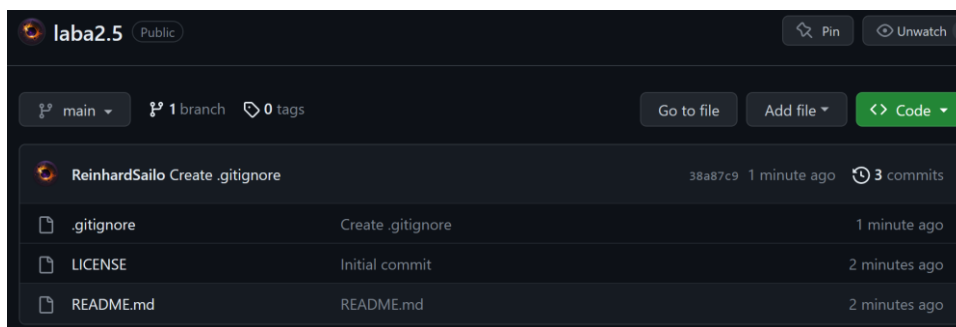


Рисунок 1. Новый репозиторий

2. В ходе данной лабораторной работы работал с моделью ветвления git flow

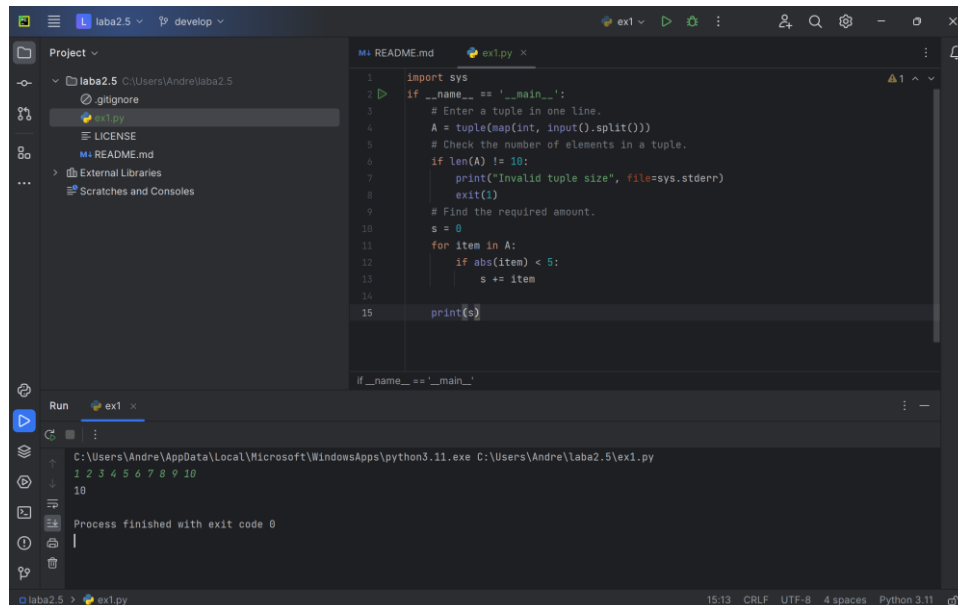
```
PS C:\Users\Andre> git clone https://github.com/ReinhardSailo/laba2.5.git
Cloning into 'laba2.5'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 10 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (10/10), done.
PS C:\Users\Andre> cd laba2.5
PS C:\Users\Andre\laba2.5> 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/Andre/laba2.5/.git/hooks]
PS C:\Users\Andre\laba2.5> git branch
* develop
main
PS C:\Users\Andre\laba2.5> |
```

Рисунок 2. Клонирование и модель ветвления git-flow

3. Пример №1. 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 a code editor with a project named 'laba2.5'. The file 'ex1.py' is open, containing the following Python code:

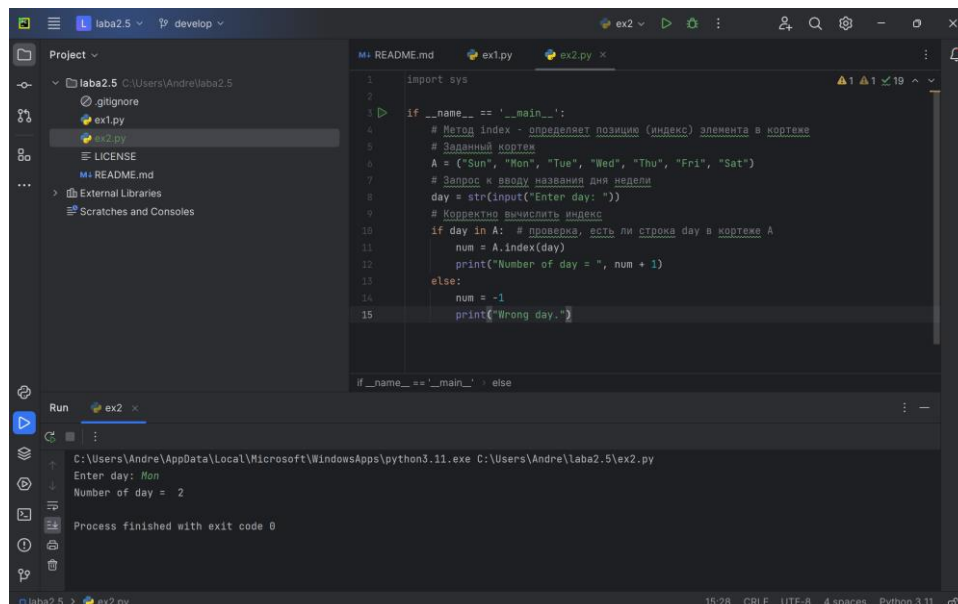
```
1 import sys
2 if __name__ == '__main__':
3     # Enter a tuple in one line.
4     A = tuple(map(int, input().split()))
5     # Check the number of elements in a tuple.
6     if len(A) != 10:
7         print("Invalid tuple size", file=sys.stderr)
8         exit(1)
9     # Find the required amount.
10    s = 0
11    for item in A:
12        if abs(item) < 5:
13            s += item
14
15    print(s)
```

The Run window shows the execution of 'ex1.py' with the following output:

```
1 2 3 4 5 6 7 8 9 10
10
Process finished with exit code 0
```

Рисунок 3. Программа и ее результат.

4. Пример №2. Условие примера: в перечне названий дней недели вычисляется порядковый номер дня.



The screenshot shows a code editor with a project named 'laba2.5'. The file 'ex2.py' is open, containing the following Python code:

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

The Run window shows the execution of 'ex2.py' with the following output:

```
Enter day: Mon
Number of day = 2
Process finished with exit code 0
```

Рисунок 4. Программа и ее результат.

5. Индивидуальное задание №1

Вариант 11. Условие задания: 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.

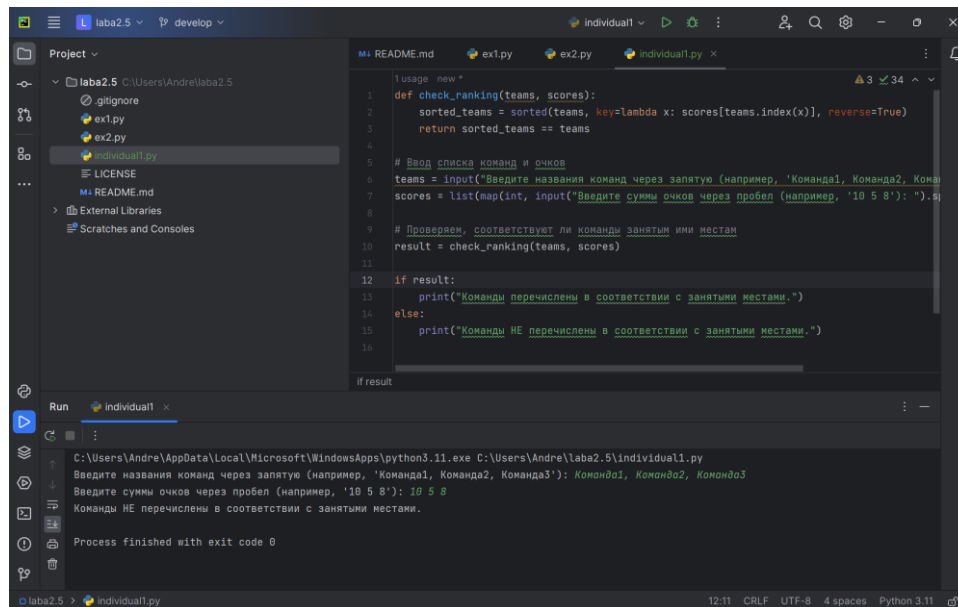


Рисунок 5. Программа и ее результат.

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

```
PS C:\Users\Andre\laba2.5> git check out main
git: 'check' is not a git command. See 'git --help'.

The most similar command is
  checkout
PS C:\Users\Andre\laba2.5> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\Andre\laba2.5> git merge develop
Updating 38a87c9..1ef3e14
Fast-forward
 ex1.py      | 15 ++++++
 ex2.py      | 15 ++++++
 individual1.py | 15 ++++++
 3 files changed, 45 insertions(+)
 create mode 100644 ex1.py
 create mode 100644 ex2.py
 create mode 100644 individual1.py
PS C:\Users\Andre\laba2.5> git push
To https://github.com/ReinhardSailo/laba2.5.git
```

Рисунок 6. Слил ветки.

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

нескольких 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.