

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение
высшего образования**

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра информационных систем и технологий

Отчет по лабораторной работе №2,5.

Дисциплина: «Основы программной инженерии»

Выполнил:

Студент группы ПИЖ-б-о-22-1,
направление подготовки: 09.03.04

«Программная инженерия»

ФИО: Рядская Мария Александровна

Проверил:

Воронкин Р. А.

Ставрополь 2023

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


Выполнение работы:

Создайте новый репозиторий

Репозиторий содержит все файлы проекта, включая историю изменений. У вас уже есть репозиторий проекта в другом месте? [Импортируйте репозиторий](#).

Обязательные поля отмечены звездочкой (*).



Владелец * Название репозитория *

 мрядская ▾ / lab8

✔️ доступен lab8.

Названия отличных репозиториях короткие и запоминающиеся. Вам нужно вдохновение? Как насчет [musical-octo-tribble?](#)

Описание (необязательно)

- ☒  **Общедоступный**
Этот репозиторий может видеть любой пользователь Интернета. Вы сами выбираете, кто может совершать коммиты.
- ☐  **Приватный**
Вы сами выбираете, кто может просматривать этот репозиторий и вносить в него изменения.

Инициализируйте этот репозиторий с помощью:

- ☒ **Добавьте файл README**
Здесь вы можете написать длинное описание вашего проекта. [Узнайте больше о READMEs.](#)

Добавить файл .gitignore

.шаблон gitignore: Отсутствует ▾

- Изучила теоретический материал работы.

Создала репозиторий на git.hub.

```
C:\Users>cd ..  
  
C:\>cd git1  
  
C:\git1>cd lab8  
Системе не удается найти указанный путь.  
  
C:\git1>git clone https://github.com/mryadskaya/lab8.git  
Cloning into 'lab8'...  
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:\git1>cd lab8  
  
C:\git1\lab8>
```

- Организовать свой репозиторий в соответствии с моделью ветвления git-flow.

```
C:\git1\lab8>git branch develop  
  
C:\git1\lab8>git push -u origin develop  
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0  
remote:  
remote: Create a pull request for 'develop' on GitHub by visiting:  
remote:      https://github.com/mryadskaya/lab8/pull/new/develop  
remote:  
To https://github.com/mryadskaya/lab8.git  
* [new branch]      develop -> develop  
branch 'develop' set up to track 'origin/develop'.  
  
C:\git1\lab8>git checkout develop  
Switched to branch 'develop'  
Your branch is up to date with 'origin/develop'.  
  
C:\git1\lab8>
```

Проработать примеры лабораторной работы

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

import sys

if __name__ == '__main__':
    # Ввести кортеж одной строкой.
    A = tuple(map(int, input().split()))
    # Проверить количество элементов кортежа.
    if len(A) != 10:
        print("Неверный размер кортежа", file=sys.stderr)
        exit(1)

    # Найти искомую сумму.
    s = 0
    for item in A:
        if abs(item) < 5:
            s += item

    print(s)
```

__name__ == '__main__'

пример1 x

:

C:\Users\ADMIN\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\git1\lab3\пример1.py

12 4 1 5 1 8 76 4 8 9

10

Process finished with exit code 0

|

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  if __name__ == '__main__':
7      # Ввести список одной строкой.
8      A = list(map(int, input().split()))
9      # Проверить количество элементов списка.
10     if len(A) != 10:
11         print("Неверный размер списка", file=sys.stderr)
12         exit(1)
13
14     # Найти искомую сумму.
15     s = sum(a for a in A if abs(a) < 5)
16     print(s)
```

__name__ == '__main__'

Python 3.10.4 Shell

пример1,2 x

C:\Users\ADMIN\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\git1\l
1 2 3 4 5 6 7 8 9 11
10

Process finished with exit code 0

15. Известно количество мячей, забитых футбольной командой за каждую игру в двух чемпионатах, которое хранится в двух кортежах. В каждом из чемпионатов команда сыграла 26 игр. Найти общее количество мячей, забитых командой в двух чемпионатах.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  if __name__ == '__main__':
7      championship_1 = tuple(map(int, input().split()))
8      championship_2 = tuple(map(int, input().split()))
9
10     total_goals = 0
11
12     for goals in championship_1:
13         total_goals += goals
14
15     for goals in championship_2:
16         total_goals += goals
17
18     print("Общее количество мячей, забитых командой в двух чемпионатах:",
19         total_goals)
20
```

Run индивидуальное ×

C:\Users\ADMIN\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\git1\lab3\индивидуальное.py

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 12 22 23 24 25 26
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 12 22 23 24 25 26

Общее количество мячей, забитых командой в двух чемпионатах: 684

Process finished with exit code 0

- Зафиксировал все изменения в github в ветке develop.
- Слила ветки.

```

C:\git1\lab8>git status
On branch develop
Your branch is up to date with 'origin/develop'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   "PyCharm/\320\270\320\275\320\264\320\270\320\262\320\270\320\264\32
1\203\320\260\320\273\321\214\320\275\320\276\320\265.py"
    new file:   "PyCharm/\320\277\321\200\320\270\320\274\320\265\321\2001,2.py"
    new file:   "PyCharm/\320\277\321\200\320\270\320\274\320\265\321\2001.py"

C:\git1\lab8>git commit -m "Сщхранение изменений"
[develop e25fa02] Сщхранение изменений
 3 files changed, 55 insertions(+)
 create mode 100644 "PyCharm/\320\270\320\275\320\264\320\270\320\262\320\270\320\264\32
1\203\320\260\320\273\321\214\320\275\320\276\320\265.py"
 create mode 100644 "PyCharm/\320\277\321\200\320\270\320\274\320\265\321\2001,2.py"
 create mode 100644 "PyCharm/\320\277\321\200\320\270\320\274\320\265\321\2001.py"

C:\git1\lab8>git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.20 KiB | 408.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/mryadskaya/lab8.git
   f9a1e65..e25fa02  develop -> develop

C:\git1\lab8>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

C:\git1\lab8>git mergge develop
git: 'mergge' is not a git command. See 'git --help'.

The most similar command is
    merge

C:\git1\lab8>git merge develop
Updating f9a1e65..e25fa02
Fast-forward
 ...0\260\320\273\321\214\320\275\320\276\320\265.py" | 19 ++++++
 ...77\321\200\320\270\320\274\320\265\321\2001,2.py" | 16 ++++++
 ...277\321\200\320\270\320\274\320\265\321\2001.py" | 20 ++++++
 3 files changed, 55 insertions(+)

```

Списки в Python - это упорядоченные изменяемые коллекции значений. Они могут содержать различные типы данных - числа, строки, словари, списки и т.д.

Кортежи в Python используются для хранения неизменяемых коллекций значений. Они похожи на списки, но отличаются тем, что их элементы нельзя изменять после создания кортежа.

Создание кортежей осуществляется с помощью функции `tuple()`. Например:

```
tup = tuple('abcdefgh')
```

Доступ к элементам кортежа осуществляется через индекс элемента в кортеже. Например:

```
my_tuple = ('a', 'b', 'c')
```

```
item = my_tuple[1]
```

Распаковка кортежа позволяет извлечь элементы из кортежа в отдельные переменные. Это может быть полезно, когда нужно работать с каждым элементом кортежа по отдельности.

В множественном присваивании кортежи используются для создания нескольких переменных одновременно. Например:

```
a, b, c = 1, 2, 3
```

Срезы кортежей позволяют выбрать подмножество элементов кортежа. Например:

```
my_slice = my_tuple[2:5]
```