

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 ЛАБОРАТОРНОЙ РАБОТЕ №2.5
дисциплины «Основы кроссплатформенного программирования»

Вариант ____

Выполнила:
Маньшина Дарья Алексеевна
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 кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

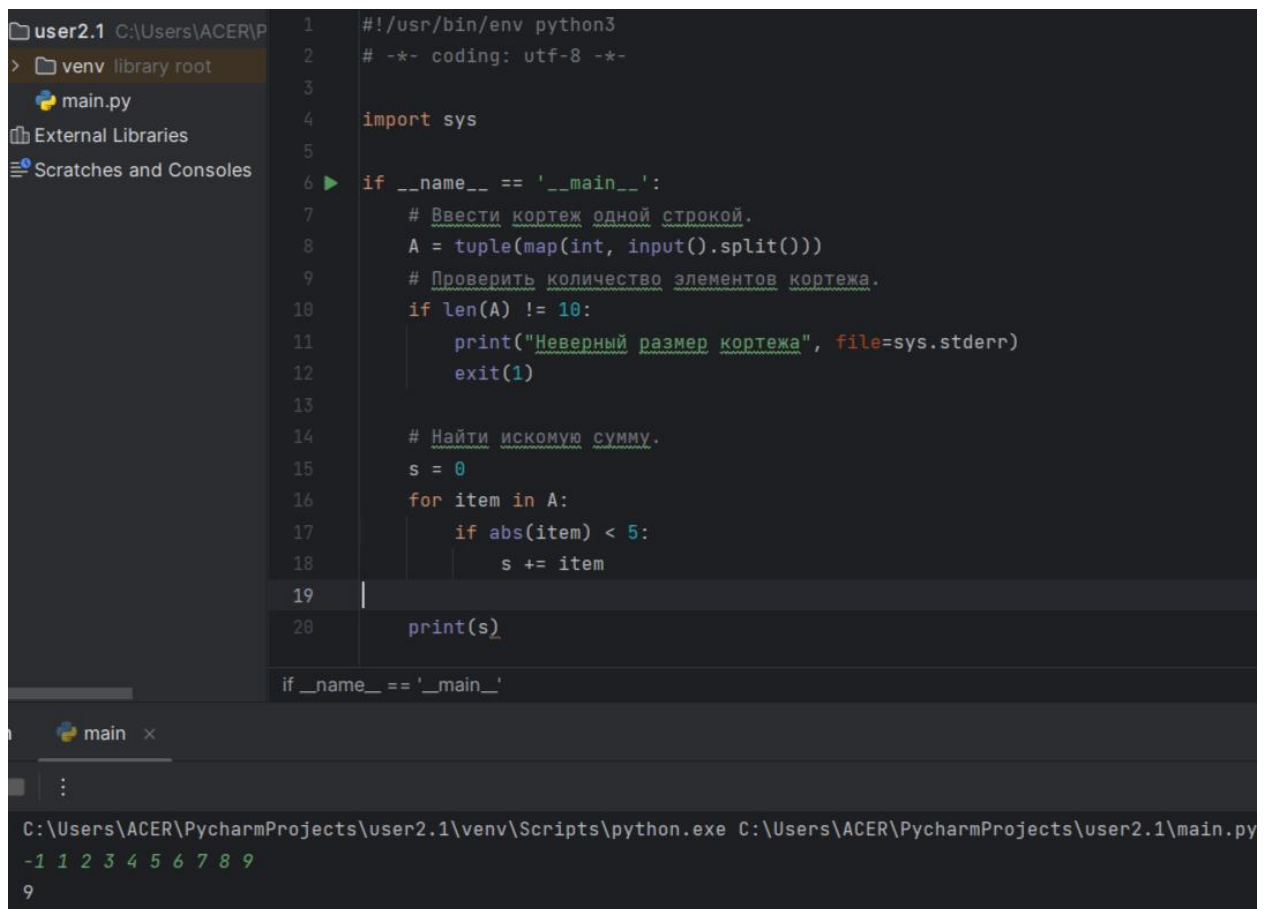
Тема: работа с кортежами в языке Python

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

Ход работы:

1. Выполним примеры из методички.

Пример 1. Ввести кортеж A из 10 элементов, найти сумму элементов, меньших по модулю 5, и вывести ее на экран. Использовать в программе вместо списков кортежи.



```
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)
```

main

C:\Users\ACER\PycharmProjects\user2.1\venv\Scripts\python.exe C:\Users\ACER\PycharmProjects\user2.1\main.py
-1 1 2 3 4 5 6 7 8 9
9

Рисунок 1 – Программа и решение примера №1

Пример 2. В примере, в перечне названий дней недели вычисляется порядковый номер дня.

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

main x

C:\Users\ACER\PycharmProjects\user2.1\venv\Scripts\python.exe C:\Users\ACER\PycharmProjects\user2.1\main.py

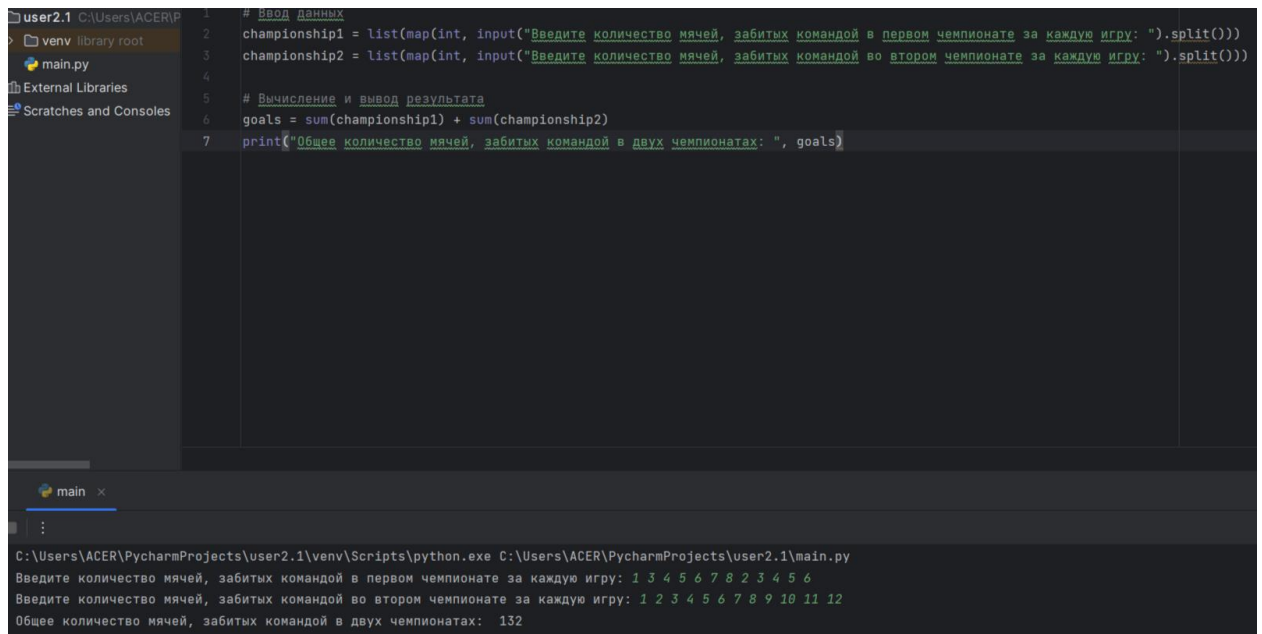
Enter day: Sun

Number of day = 1

Рисунок 2 – Программа и решение примера №2

2. Выполним индивидуальное задание. Вариант 15.

Задание. 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 # Ввод данных
2 championship1 = list(map(int, input("Введите количество мячей, забитых командой в первом чемпионате за каждую игру: ").split()))
3 championship2 = list(map(int, input("Введите количество мячей, забитых командой во втором чемпионате за каждую игру: ").split()))
4
5 # Вычисление и вывод результата
6 goals = sum(championship1) + sum(championship2)
7 print("Общее количество мячей, забитых командой в двух чемпионатах: ", goals)
```

main x

C:\Users\ACER\PycharmProjects\user2.1\venv\Scripts\python.exe C:\Users\ACER\PycharmProjects\user2.1\main.py

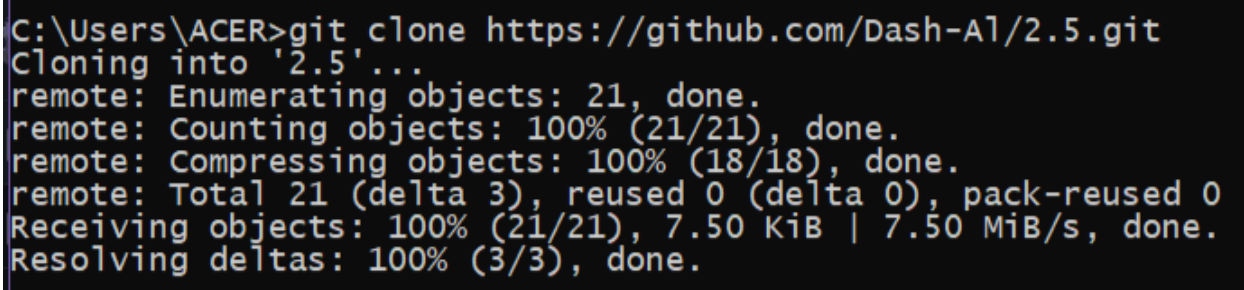
Введите количество мячей, забитых командой в первом чемпионате за каждую игру: 1 3 4 5 6 7 8 2 3 4 5 6

Введите количество мячей, забитых командой во втором чемпионате за каждую игру: 1 2 3 4 5 6 7 8 9 10 11 12

Общее количество мячей, забитых командой в двух чемпионатах: 132

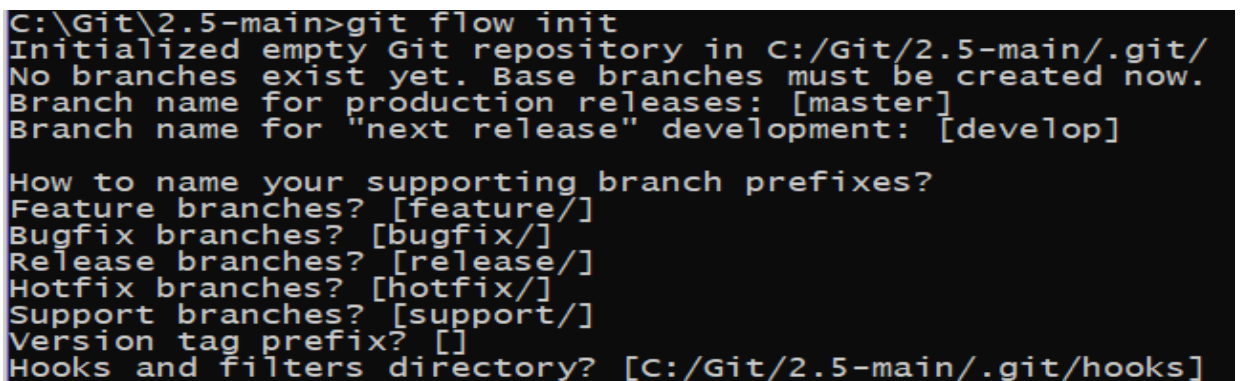
Рисунок 3 – Программа и решение индивидуального задания

3. Создала общедоступный репозиторий, в котором использована лицензия MIT и язык программирования Python. Был добавлен файл .gitignore. Клонирование репозитория. Организация репозитория с моделью ветвления git flow.



```
C:\Users\ACER>git clone https://github.com/Dash-A1/2.5.git
Cloning into '2.5'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 21 (delta 3), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (21/21), 7.50 KiB | 7.50 MiB/s, done.
Resolving deltas: 100% (3/3), done.
```

Рисунок 4 – Результат клонирования



```
C:\Git\2.5-main>git flow init
Initialized empty Git repository in C:/Git/2.5-main/.git/
No branches exist yet. Base branches must be created now.
Branch name for production releases: [master]
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:/Git/2.5-main/.git/hooks]
```

Рисунок 5 – Результат git flow init

Ответы на контрольные вопросы

1. Что такое списки в языке Python?

Списки в Python – упорядоченный изменяемый набор объектов произвольных типов, пронумерованных от 0. Они используются для хранения и работы с данными.

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. Кортеж также требует выделения значительно меньшего количества памяти.

3. Как осуществляется создание кортежей?

Для создания кортежа используются круглые скобки, в которые помещаются его значения, разделенные запятыми.

4. Как осуществляется доступ к элементам кортежа?

Чтобы обратиться к элементам кортежа необходимо написать имя кортежа, а затем в квадратных скобках указать индекс нужного (позицию) элемента. Важный момент: списки начинаются с 0, то есть первый элемент списка имеет индекс [0].

5. Зачем нужна распаковка (деструктуризация) кортежа?

Распаковка кортежа позволяет извлекать отдельные элементы кортежа и присваивать их значения переменным.

6. Какую роль играют кортежи в множественном присваивании?

У кортежей важная роль в множественном присваивании, потому как они могут быть использованы для одновременного присваивания нескольких переменных из кортежа.

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 кортежей в Python выполняется с помощью оператора «+». Для повторения кортежа – «*».

9. Как выполняется обход элементов кортежа?

Выполняется с помощью цикла for.

10. Как проверить принадлежность элемента кортежу?

Чтобы проверить, можно использовать оператор in.

11. Какие методы работы с кортежами Вам известны?

len () – возвращает длину кортежа.

count () – возвращает количество вхождений элемента в кортеж.

index () – возвращает индекс первого вхождения элемента в кортеже.

sorted () – возвращает новый отсортированный список из элементов кортежа.

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

Использование представленных функций агрегации является допустимым, так как они позволяют получать информацию о элементах кортежа и выполнять различные математические операции.

13. Как создать кортеж с помощью спискового включения.

При помощи функции tuple().

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