

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций
«Работа с кортежами в языке Python»**

**Отчет по лабораторной работе № 2.5
по дисциплине «Основы кроссплатформенного
программирования»**

Выполнил студент группы ИВТ-б-о-21-1

Шайдеров Дмитрий Викторович.

«20 » мая 2022г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

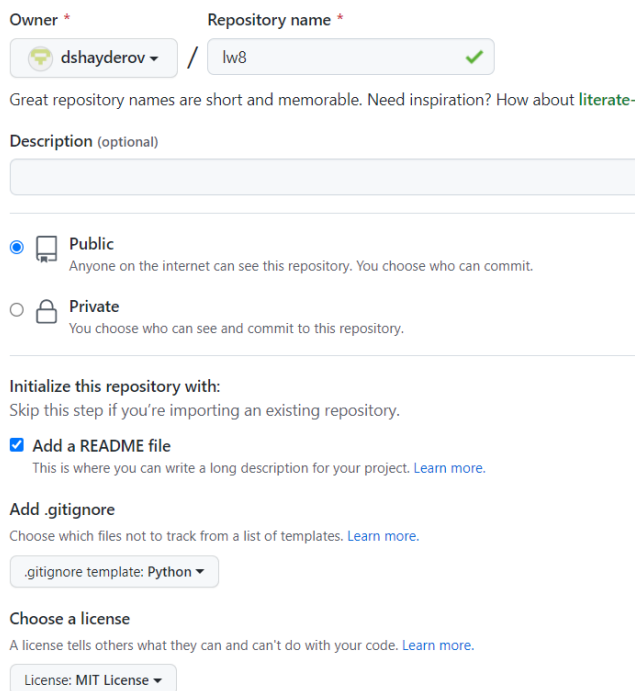
Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2022

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

Порядок выполнения работы:

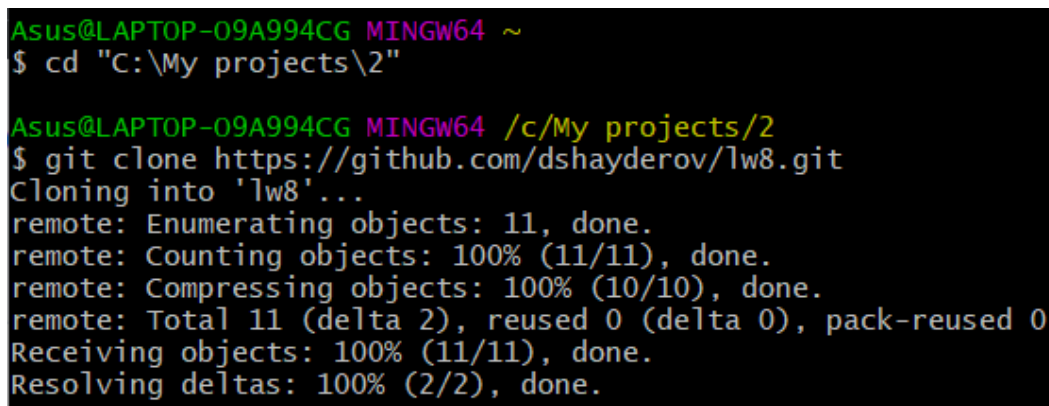
1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.



The screenshot shows the GitHub repository creation interface. At the top, the 'Owner' is set to 'dshayderov' and the 'Repository name' is 'lw8', which is marked as valid with a green checkmark. Below this, a message states: 'Great repository names are short and memorable. Need inspiration? How about [literate](#).' The 'Description (optional)' field is empty. Under the 'Visibility' section, the 'Public' option is selected with a radio button, accompanied by the text: 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option is unselected. The 'Initialize this repository with:' section includes a checkbox for 'Add a README file', which is checked, with a note: 'This is where you can write a long description for your project. [Learn more](#).' Below this is the 'Add .gitignore' section, where the '.gitignore template: Python' is selected. Finally, the 'Choose a license' section shows the 'License: MIT License' selected.

Рисунок 1 - Создание репозитория

2. Выполните клонирование созданного репозитория.



```
Asus@LAPTOP-09A994CG MINGW64 ~  
$ cd "C:\My projects\2"  
  
Asus@LAPTOP-09A994CG MINGW64 /c/My projects/2  
$ git clone https://github.com/dshayderov/lw8.git  
Cloning into 'lw8'...  
remote: Enumerating objects: 11, done.  
remote: Counting objects: 100% (11/11), done.  
remote: Compressing objects: 100% (10/10), done.  
remote: Total 11 (delta 2), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (11/11), done.  
Resolving deltas: 100% (2/2), done.
```

Рисунок 2 - Клонирование репозитория

3. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.

```

Asus@LAPTOP-09A994CG MINGW64 /c/My projects/2
$ cd "C:\My projects\2\lw8"

Asus@LAPTOP-09A994CG MINGW64 /c/My projects/2/lw8 (main)
$ git checkout -b develop
Switched to a new branch 'develop'

Asus@LAPTOP-09A994CG MINGW64 /c/My projects/2/lw8 (develop)
$

```

Рисунок 3 - Ветвление по модели git-flow

4. Создайте проект PyCharm в папке репозитория.

> Этот компьютер > OS (C:) > My projects > 2 > lw8 >				
Имя	Дата изменения	Тип	Размер	
Project	30.05.2022 14:54	Папка с файлами		
.gitignore	30.05.2022 14:52	Файл "GITIGNORE"	2 КБ	
LICENSE	30.05.2022 14:52	Файл	2 КБ	
README.md	30.05.2022 14:52	Файл "MD"	1 КБ	

Рисунок 4 - Проект PyCharm

5. Проработайте примеры лабораторной работы. Создайте для каждого примера отдельный модуль языка Python. Зафиксируйте изменения в репозитории.

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

```

example_1 x
"C:\My projects\2\lw8\Project\venv\Sc
0 1 2 3 4 5 6 7 8 9
10

Process finished with exit code 0

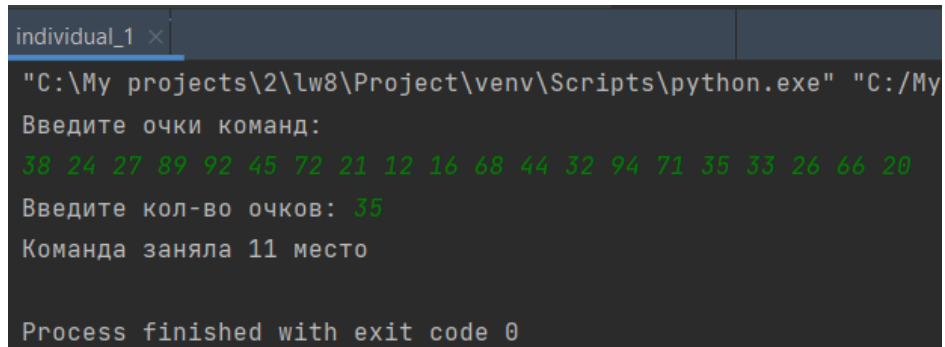
```

Рисунок 5 – Результат выполнения программы

6. Выполните индивидуальное задание (Вариант 27/1).

Известно количество очков, набранных каждой из 20 команд – участниц первенства по футболу. Перечень очков дан в порядке убывания (ни одна пара команд не набрала одинаковое количество очков). Определить,

какое место заняла команда, набравшая очков (естественно, что значение имеется в перечне). Условный оператор не использовать.



```
individual_1 x
"C:\My projects\2\lw8\Project\venv\Scripts\python.exe" "C:/My
Введите очки команд:
38 24 27 89 92 45 72 21 12 16 68 44 32 94 71 35 33 26 66 20
Введите кол-во очков: 35
Команда заняла 11 место
Process finished with exit code 0
```

Рисунок 6 - Результат выполнения программы

Контрольные вопросы:

1. Что такое кортежи в языке Python?

Кортеж (tuple) – это неизменяемая структура данных, которая по своему подобию очень похожа на список.

2. Каково назначение кортежей в языке Python?

Существует несколько причин, по которым стоит использовать кортежи вместо списков. Одна из них – это обезопасить данные от случайного изменения. Если мы получили откуда-то массив данных, и у нас есть желание поработать с ним, но при этом непосредственно менять данные мы не собираемся, тогда, это как раз тот случай, когда кортежи придутся как нельзя кстати. Кортежи в памяти занимают меньший объем по сравнению со списками. Кортежи работают быстрее, чем списки

3. Как осуществляется создание кортежей?

a = ()

b = tuple()

4. Как осуществляется доступ к элементам кортежа?

Доступ к элементам кортежа осуществляется также как к элементам списка – через указание индекса.

5. Зачем нужна распаковка (деструктуризация) кортежа?

Обращение по индексу, это не самый удобный способ работы с кортежами. Дело в том, что кортежи часто содержат значения разных типов, и помнить, по какому индексу что лежит — очень непросто.

6. Какую роль играют кортежи в множественном присваивании?

Используя множественное присваивание, можно провернуть интересный трюк: обмен значениями между двумя переменными.

7. Как выбрать элементы кортежа с помощью среза?

С помощью операции взятия среза можно получить другой кортеж. Общая форма операции взятия среза для кортежа следующая

$$T2 = T1[i:j]$$

здесь

$T2$ – новый кортеж, который получается из кортежа $T1$;

$T1$ – исходный кортеж, для которого происходит срез;

i, j – соответственно нижняя и верхняя границы среза. Фактически берутся ко вниманию элементы, лежащие на позициях $i, i+1, \dots, j-1$. Значение j определяет позицию за последним элементом среза.

8. Как выполняется конкатенация и повторение кортежей?

Для кортежей можно выполнять операцию конкатенации, которая обозначается символом $+$.

$$T3 = T1 + T2$$

9. Как выполняется обход элементов кортежа?

Элементы кортежа можно последовательно просмотреть с помощью операторов цикла `while` или `for`.

10. Как проверить принадлежность элемента кортежу?

Проверка вхождения элемента в кортеж - оператор `in`.

11. Какие методы работы с кортежами Вам известны?

`index()`, `count()`.

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

Доступно.

13. Как создать кортеж с помощью спискового включения.

Так же как и список

Вывод: приобрел навыки по работе с кортежами при написании программ с помощью языка программирования Python версии 3.x.