

**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 №8.

Дисциплина: «Основы программной инженерии»

Выполнил:

Студент группы ПИЖ-б-о-22-1,

направление подготовки: 09.03.04

«Программная инженерия»

ФИО: Гуртовой Ярослав Дмитриевич

Проверил:

Воронкин Р. А.

Ставрополь 2023

Тема: Лабораторная работа 2.5 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..


Выполнение работы:

1. Изучил теоретический материал работы.
2. Создал репозиторий на git.hub.
3. Клонировал репозиторий.

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk ().*


Owner * **Repository name ***


 KingItProgger /

✔ lr-2.5 is available.

Great repository names are short and memorable. Need inspiration? How about **fluffy-winner** ?

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:


☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)


Add .gitignore

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set  **main** as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

[Create repository](#)

Рисунок 1 – создание репозитория

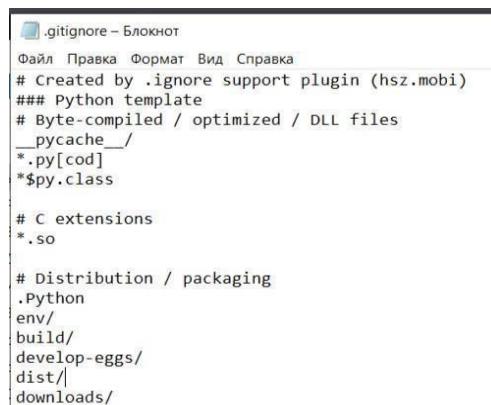
```

PS C:\Users\User\Desktop\учеба\Зсемак\змій\lr_2.5> git clone https:
Cloning into 'lr-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.
PS C:\Users\User\Desktop\учеба\Зсемак\змій\lr_2.5>

```

Рисунок 2 – клонирование репозитория 4.

Дополнить файл gitignore необходимыми правилами.



```

.gitignore – Блокнот
Файл Правка Формат Вид Справка
# Created by .ignore support plugin (hsz.mobi)
### Python template
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
env/
build/
develop-eggs/
dist/
downloads/

```

Рисунок 3 – .gitignore для IDE PyCharm

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

```

PS C:\Users\User\Desktop\учеба\Зсемак\змій\lr_2.5\lr-2.5> git checkout -b develop
Switched to a new branch 'develop'
PS C:\Users\User\Desktop\учеба\Зсемак\змій\lr_2.5\lr-2.5>

```

Рисунок 4 – создание ветки develop

6. Проработал примеры из методички.

```
#!/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)
```

Рисунок 5 – пример 1

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

import sys

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

    # Найти искомую сумму.
    s = sum(a for a in A if abs(a) < 5)
    print(s)
```

Рисунок 6 – пример 1(2)

```
C:\Users\User\AppData\Local\Program
1 2 3 4 5 6 7 8 9 0
10

Process finished with exit code 0
```

Рисунок 7 – пример выполнения примера 1

7. Если в кортеже есть хотя бы одна пара одинаковых соседних элементов, то напечатать все элементы, следующие за элементами первой из таких пар.

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

import sys

if __name__ == '__main__':

    a = tuple(map(int, input().split()))

    l = 0
    for i, item in enumerate(a):
        if i < len(a)-1:
            if item == a[i+1]:
                l = i + 2
                break
        else:
            break

    print(a[l:])
```

Рисунок 8 – индивидуальное задание

```
C:\Users\User\AppData\Local\Program
1 2 3 4 5 6
(1, 2, 3, 4, 5, 6)

Process finished with exit code 0
|
```

Рисунок 9 – пример выполнения индивидуального задания

9.Зафиксировал все изменения в github в ветке develop.

```
PS C:\Users\User\Desktop\учеба\Зсемак\змій\lr_2.5\lr-2.5> git add .
warning: in the working copy of 'pycharm/.idea/inspectionProfiles/profiles_setti
e next time Git touches it
PS C:\Users\User\Desktop\учеба\Зсемак\змій\lr_2.5\lr-2.5> git status
On branch develop
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   pycharm/.idea/.gitignore
    new file:   pycharm/.idea/inspectionProfiles/profiles_settings.xml
    new file:   pycharm/.idea/lr 2.5.iml
    new file:   pycharm/.idea/misc.xml
    new file:   pycharm/.idea/modules.xml
    new file:   pycharm/ex 1.2.py
    new file:   pycharm/ex1.py
    new file:   pycharm/task1.py

PS C:\Users\User\Desktop\учеба\Зсемак\змій\lr_2.5\lr-2.5> git commit -m"laba"
[develop 9b70608] laba
 8 files changed, 84 insertions(+)
 create mode 100644 pycharm/.idea/.gitignore
 create mode 100644 pycharm/.idea/inspectionProfiles/profiles_settings.xml
 create mode 100644 pycharm/.idea/lr 2.5.iml
 create mode 100644 pycharm/.idea/misc.xml
 create mode 100644 pycharm/.idea/modules.xml
 create mode 100644 pycharm/ex 1.2.py
 create mode 100644 pycharm/ex1.py
 create mode 100644 pycharm/task1.py
PS C:\Users\User\Desktop\учеба\Зсемак\змій\lr_2.5\lr-2.5>
```

Рисунок 10 – фиксация изменений в ветку develop

10.Слил ветки.


```

PS C:\Users\User\Desktop\учеба\Зсемак\змії\lr_2.5\lr-2.5> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\User\Desktop\учеба\Зсемак\змії\lr_2.5\lr-2.5> git merge develop
Updating 7a16905..9b70608
Fast-forward
 pycharm/.idea/.gitignore           | 3 +++
 .../.idea/inspectionProfiles/profiles_settings.xml | 6 ++++++
 pycharm/.idea/lr 2.5.iml           | 8 ++++++++
 pycharm/.idea/misc.xml             | 4 ++++
 pycharm/.idea/modules.xml          | 8 ++++++++
 pycharm/ex 1.2.py                  | 16 +++++++++++++++++++++
 pycharm/ex1.py                     | 20 +++++++++++++++++++++
 pycharm/task1.py                   | 19 +++++++++++++++++++++
8 files changed, 84 insertions(+)
create mode 100644 pycharm/.idea/.gitignore
create mode 100644 pycharm/.idea/inspectionProfiles/profiles_settings.xml
create mode 100644 pycharm/.idea/lr 2.5.iml
create mode 100644 pycharm/.idea/misc.xml
create mode 100644 pycharm/.idea/modules.xml
create mode 100644 pycharm/ex 1.2.py
create mode 100644 pycharm/ex1.py
create mode 100644 pycharm/task1.py
PS C:\Users\User\Desktop\учеба\Зсемак\змії\lr_2.5\lr-2.5>

```

Рисунок 11 – сливание ветки develop в ветку main

1. Что такое списки в языке Python?

Список (list) – это структура данных для хранения объектов различных типов. Список очень похож на массив, только, как было уже сказано выше, в нем можно хранить объекты различных типов. Размер списка не статичен, его можно изменять. Список по своей природе является изменяемым типом данных. Переменная, определяемая как список, содержит ссылку на структуру в памяти, которая в свою очередь хранит ссылки на какие-либо другие объекты или структуры

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. Используя их в данной задаче, мы дополнительно получаем сразу несколько бонусов – во-первых, это экономия места

3. Как осуществляется создание кортежей?

```
>>> a = ()  
>>> b = tuple()  
>>> a = (1, 2, 3, 4, 5)  
>>> a = tuple([1, 2, 3, 4])  
>>> tuple = (42,)
```

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?

Распаковка (или деструктуризация) кортежа в программировании позволяет извлечь отдельные элементы кортежа и присвоить их переменным. Это полезная операция, потому что она позволяет удобно работать с данными в кортеже, делая их доступными для дальнейшего использования.

6. Какую роль играют кортежи в множественном присваивании?

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

7. Как выбрать элементы кортежа с помощью среза?

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

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, ..., j-1. Значение j определяет позицию за последним элементом среза.

8. Как выполняется конкатенация и повторение кортежей?

Во время конкатенации складываются два кортежа, а во время повторения, они повторяются n раз.

9. Как выполняется обход элементов кортежа?

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

10. Как проверить принадлежность элемента кортежу?

С помощью оператора `in`.

11. Какие методы работы с кортежами Вам известны?

Метод `index()` - поиск позиции элемента в кортеже

Метод `count()` - количество вхождений элемента в кортеж

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

Допустимо использование.

13. Как создать кортеж с помощью спискового включения.

`my_tuple = tuple(i for i in A)`