

**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 программной инженерии»**

Выполнил студент группы ПИЖ-б-о-21-1

Халимендик Я. Д. « » 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.

Ход работы:

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.

Create a new repository

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

Owner * **Repository name ***

Yana-Kh / Lab-8-OPJ ✓

Great repository names are short and memorable. Need inspiration? How about [legendary-train](#)?

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:
Skip this step if you're importing an existing repository.

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

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

.gitignore template: Python ▼

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

License: MIT License ▼

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

3. Выполните клонирование созданного репозитория.

```
C:\Users\ynakh\OneDrive\Рабочий стол\Git> git clone https://github.com/Yana-Kh/Lab-8-OPJ.git
Cloning into 'Lab-8-OPJ'...
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:\Users\ynakh\OneDrive\Рабочий стол\Git>
```

Рисунок 2 – Клонирование репозитория

4. Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm.

```
C:\Users\ynakh\OneDrive\Рабочий стол\Git\Lab-8-OPJ>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   .gitignore

C:\Users\ynakh\OneDrive\Рабочий стол\Git\Lab-8-OPJ>git commit -m"ignore"
[main 424e046] ignore
 1 file changed, 22 insertions(+), 2 deletions(-)

C:\Users\ynakh\OneDrive\Рабочий стол\Git\Lab-8-OPJ>
```

Рисунок 3 – Дополнение файла .gitignore

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

```
Командная строка
1 file changed, 22 insertions(+), 2 deletions(-)

C:\Users\ynakh\OneDrive\Рабочий стол\Git\Lab-8-OPJ>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/ynakh/OneDrive/Рабочий стол/Git/Lab-8-OPJ/.git/hooks]

C:\Users\ynakh\OneDrive\Рабочий стол\Git\Lab-8-OPJ>
```

Рисунок 4 – Организация репозитория в соответствии с моделью git-flow

6. Создайте проект PyCharm в папке репозитория.

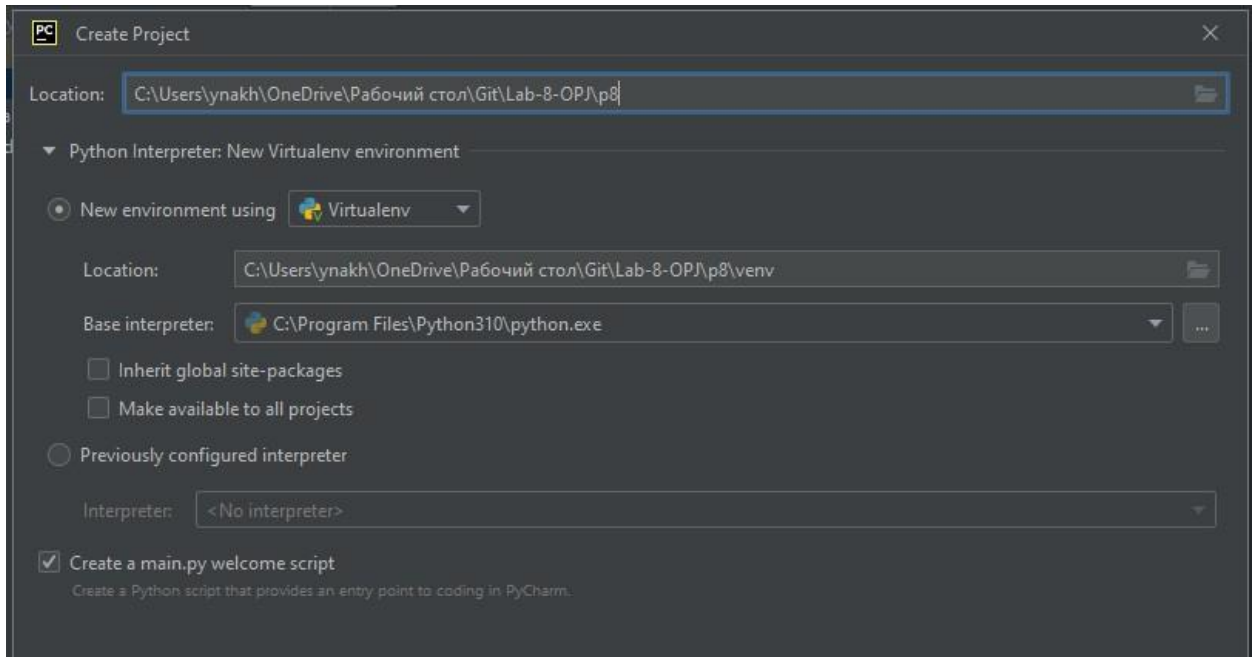


Рисунок 5 – Создание проекта PyCharm в папке репозитория

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

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.

Код:

```
#!/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)
```

```
4 6 3 9 6 3 1 9 6 12
11
Process finished with exit code 0
```

Рисунок 6 – Результат работы программы

```
3 5
Неверный размер кортежа
Process finished with exit code 1
```

Рисунок 7 – Результат работы программы

Решение через списковые включения:

Код:

```
#!/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 = sum(a for a in A if abs(a) < 5)
    print(s)
```

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

Рисунок 8 – Результат работы программы

8. Выполните индивидуальные задания, согласно своему варианту.

Вариант 6(32). Дан кортеж целых чисел. Определить, есть ли в нем хотя бы одна пара соседних нечетных чисел. В случае положительного ответа определить номера элементов первой из таких пар.

Код:

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

if __name__ == '__main__':
    # Ввести кортеж одной строкой.
    A = tuple(map(int, input().split()))

    # Найти искомые числа
    f = 0
    for i, value in enumerate(A):
        if value % 2 != 1 and A[i+1] % 2 != 1:
            print(f"Найдена пара, начиная с индекса: {i}")
            f += 1
            break
    if f == 0:
        print("Пары не найдены")
```

```
7 8 5 7 3 2 9
Найдена пара, начиная с индекса: 2
Process finished with exit code 0
```

Рисунок 9 – Результат работы программы

```
58 6 2 9 0 2 7 4
Пары не найдены(
Process finished with exit code 0
```

Рисунок 10 – Результат работы программы

Вопросы для защиты работы

1. Что такое списки в языке Python?

2. Каково назначение кортежей в языке Python?

Т.к. кортеж – неизменяемый тип данных, это защищает данные от случайного изменения. К тому же, из-за отсутствия операций изменения делают работу кортежей быстрее и место они занимают меньше, чем списки.

3. Как осуществляется создание кортежей?

Создание кортежей осуществляется следующим образом:

```
a = ()
```

```
b = tuple()
```

В скобках через запятую перечисляются элементы кортежа

Если кортеж состоит из одного элемента, то после него нужно поставить «,»:

```
tuple = (42,)
```

4. Как осуществляется доступ к элементам кортежа?

Доступ к элементам кортежа осуществляется аналогично доступу к элементам списка – через индекс.

```
>>> a = (1, 2, 3, 4, 5)
```

```
>>> print(a[0])
```

```
1
```

5. Зачем нужна распаковка (деструктуризация) кортежа?

Это облегчает доступ к элементам, осуществляя его не по индексу, а с помощью переменных

```
name_and_age = ('Bob', 42)
```

```
(name, age) = name_and_age
```

```
name # 'Bob'
```

```
age # 42
```

6. Какую роль играют кортежи в множественном присваивании?

Благодаря тому, что кортежи легко собирать и разбирать, в Python удобно делать такие вещи, как множественное присваивание. Смотрите:

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

```
a = 100
```

```
b = 'foo'
```

```
(a, b) = (b, a)
```

```
a # 'foo'
```

```
b # 100
```

Строку `(a, b) = (b, a)` нужно понимать как "присвоить в `a` и `b` значения из кортежа, состоящего из значений переменных `b` и `a`".

7. Как выбрать элементы кортежа с помощью среза?

Аналогично со списком:

```
>>> a = (1, 2, 3, 4, 5)
```

```
>>> print(a[1:3])
```

```
(2, 3)
```

8. Как выполняется конкатенация и повторение кортежей?

Конкатенация обозначается знаком «+», по сути, эта операция объединяет несколько кортежей в один новый. форма: $T3 = T1 + T2$, где $T1, T2$ – кортежи, над которыми выполняется операция, а $T3$ – новый кортеж.

9. Как выполняется обход элементов кортежа?

Обход элементов кортежа можно осуществить с помощью циклов `for` или `while`:

```
for i in A:
```

```
while i < len(A):
```

```
    i = i + 1
```


10. Как проверить принадлежность элемента кортежу?

Чтобы проверить принадлежность элемента кортежу необходимо воспользоваться операцией `in`, форма: `a = i in A`, где `a = true/false`, `i` – искомый элемент, `A` – кортеж, в котором осуществляется поиск.

11. Какие методы работы с кортежами Вам известны?

Метод `index()` осуществляет поиск позиции элемента в кортеже, форма: `pos = T.index(i)`, где `pos` – переменная, в которую будет записан индекс, `T` – кортеж в котором осуществляется поиск, `i` – искомый элемент.

Метод `count()` осуществляет подсчет количества вхождений элемента в кортеж, форма: `k = T.count(i)`, где `k` – искомое количество, `T` – кортеж в котором осуществляется поиск, `i` – искомый элемент.

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

Да

13. Как создать кортеж с помощью спискового включения.

Аналогично списку, только скобки не квадратные, а круглые, и выражение дает на выходе специальный объект генератора, а не кортеж, для преобразования которого необходимо воспользоваться вызовом `tuple()`.