

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего
образования
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»**

**Кафедра
инфокоммуникаций
Институт цифрового
развития**

ОТЧЁТ

по лабораторной работе №2.5

Дисциплина: «Основы программной инженерии»

Тема: «Работа с кортежами в языке Python»

Выполнил:
студент 2 курса
группы Пиж-б-о-21-1
Коныжев Максим
Викторович

Ставрополь 2022

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

1. Был создан репозиторий в Github в который были добавлены правила gitignore для работы IDE PyCharm, была выбрана лицензия MIT, сам репозиторий был клонирован на локальный сервер и был организован в соответствии с моделью ветвления git-flow.

The screenshot shows the GitHub 'Create repository' interface. At the top, the 'Owner' is set to 'MaxDrill' and the 'Repository name' is 'lab2_5', which is marked as valid with a green checkmark. Below this, a hint suggests that great repository names are short and memorable, with a link to 'special-memory?'. The 'Description' field is optional and currently empty. Under the 'Visibility' section, 'Public' is selected, indicating that anyone on the internet can see the repository. The 'Private' option is also visible. The 'Initialize this repository with:' section includes a checkbox for 'Add a README file', which is checked, and a link to 'Learn more'. The 'Add .gitignore' section has a dropdown menu set to '.gitignore template: Python'. The 'Choose a license' section has a dropdown menu set to 'License: MIT License'. A note at the bottom states that the default branch will be set to 'main'. A final note indicates that a public repository is being created in a personal account. A green 'Create repository' button is at the bottom.

Owner * Repository name *

MaxDrill / lab2_5 ✓

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

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](#).

i You are creating a public repository in your personal account.

Create repository

Рисунок 1.1 – Создание репозитория

```
C:\Users\UESR\gitproj>git clone https://github.com/MaxDrill/lab2_5.git
Cloning into 'lab2_5'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
Receiving objects: 60% (3/5)100% (4/4), done.
```

Рисунок 1.2 – Клонирование репозитория

```
C:\Users\UESR\gitproj\lab2_5>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/UESR/gitproj/lab2_5/.git/hooks]

C:\Users\UESR\gitproj\lab2_5>
```

Рисунок 1.4 – Организация репозитория по модели ветвления git-flow

3. Было выполнено общее задание.

```
C:\Users\UESR\gitproj\lab2_5>git flow init
3 4 5 2 1 4 3 5 7 9
17
Process finished with exit code 0
```

Рисунок 2.2 – Результат работы первого примера

3. Было выполнено индивидуальное задание согласно 7 варианту.

Дан кортеж целых чисел. Если в нем есть хотя бы одна пара соседних четных чисел, то напечатать все элементы, предшествующие элементам последней из таких пар.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import sys
if __name__ == '__main__':
    a = tuple(map(int, input().split()))
    k = -1
    if not a:
        print(
            "заданный кортеж пуст",
            file=sys.stderr
        )
        exit(1)
    for i in a:
        if i < len(a) - 1:
            if a[i] % 2 == 0 and a[i + 1] % 2 == 0:
                k = i
        else:
            break
    if k == -1:
        print("Нет таких пар")
    elif k == 0:
        print("До последней пары нет чисел")
    for i in range(0, k):
        print(a[i])
```

```
1 2 3 4 5 6 8 5 6 8
1
2
3
4
5
6
8
5
Process finished with exit code 0
```

Рисунок 3.1 – Результат работы программы

4. Было осуществлен коммит и слияние веток main и develop, также запущены изменения на удаленный сервер.

```
C:\Users\UESR\gitproj\lab2_5>git merge develop
Updating c6505a9..e338ecd
Fast-forward
 README.md      | 3 ++-
 lab8/idz1.py   | 19 ++++++
 lab8/num1.py   | 19 ++++++
 3 files changed, 40 insertions(+), 1 deletion(-)
 create mode 100644 lab8/idz1.py
 create mode 100644 lab8/num1.py
```

Рисунок 4.1 – Коммит изменений и слияние веток main и develop

```
C:\Users\UESR\gitproj\lab2_5>git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.08 KiB | 552.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MaxDrill/lab2_5.git
 c6505a9..e338ecd  main -> main
C:\Users\UESR\gitproj\lab2_5>
```

Рисунок 4.2 – Пуш на удаленный сервер

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

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 проверить 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

- T_2 – новый кортеж, который получается из кортежа T_1 ;
- T_1 – 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 $+$.

$$T_3 = T_1 + T_2$$

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.