

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

**Кафедра
инфокоммуникаций
Институт цифрового
развития**

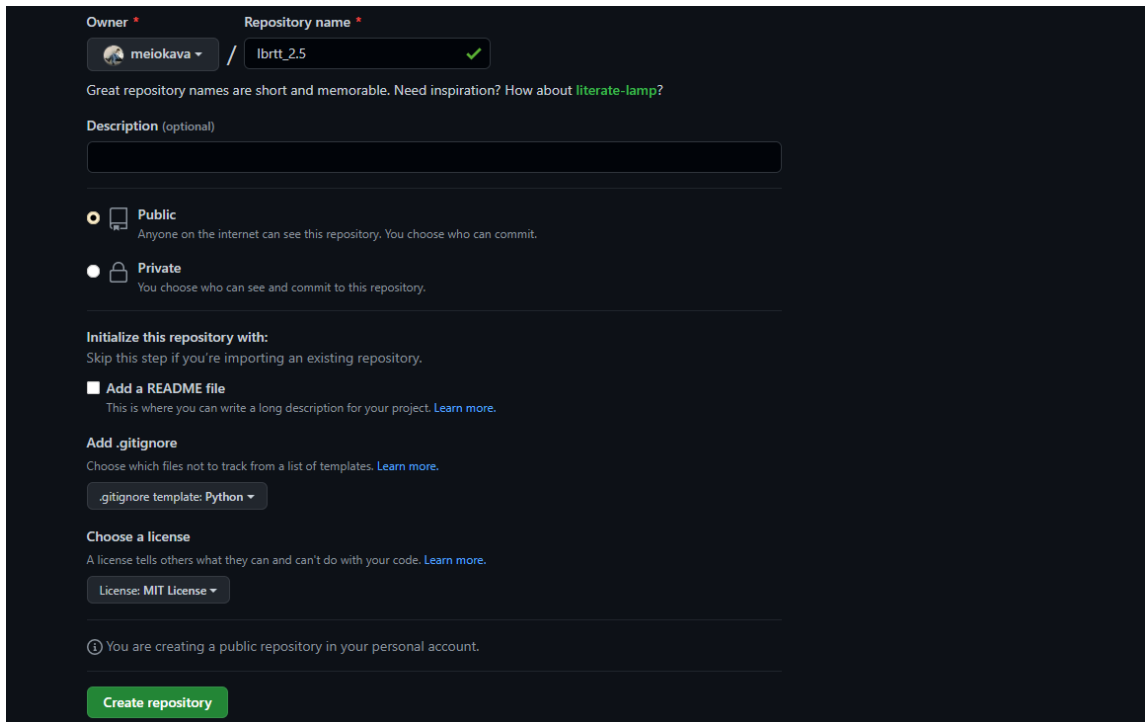
**ОТЧЁТ
по лабораторной работе №2.5
Дисциплина: «Основы программной инженерии»
Тема: «Работа с кортежами в языке Python»**

Выполнила:
студентка 2 курса
группы Пиж-б-о-21-1
Джолдошова Мээрим
Бекболотовна

Ставрополь 2022

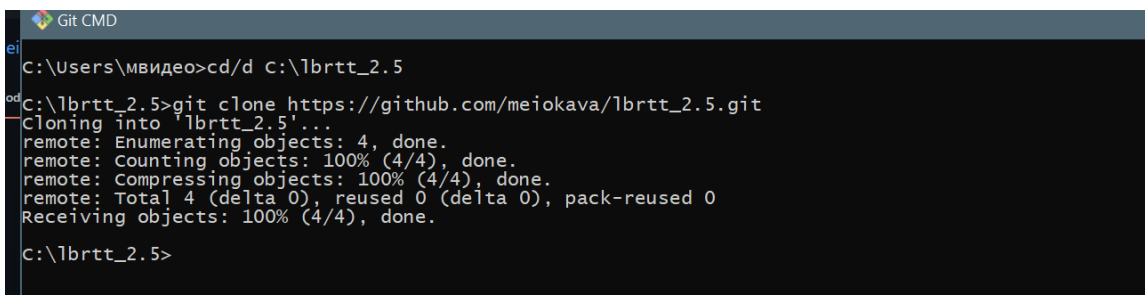
Цель работы: приобретение навыков по работе с кортежами при написании программ с помощью языка программирования Python версии 3.x.

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' form. The 'Owner' is 'meiokava' and the 'Repository name' is 'lbrtt_2.5'. The 'Description' field is empty. The 'Public' option is selected under 'Visibility'. Under 'Initialize this repository with:', the 'Add a README file' checkbox is checked. The '.gitignore' template is set to 'Python'. The 'License' is set to 'MIT License'. A green 'Create repository' button is at the bottom.

Рисунок 1.1 – Создание репозитория



```
Git CMD
C:\Users\мвидео>cd/d C:\lbrtt_2.5
C:\lbrtt_2.5>git clone https://github.com/meiokava/lbrtt_2.5.git
Cloning into 'lbrtt_2.5'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
C:\lbrtt_2.5>
```

Рисунок 1.2 – Клонирование репозитория

```
149 lines (121 sloc) | 2.12 KB
1  ## Example user template template
2  ## Example user template
3
4  # IntelliJ project files
5  .idea
6  *.iml
7  out
8  gen
9  ## Python template
10 # Byte-compiled / optimized / DLL files
11 __pycache__/
12 *.py[cod]
13 *.py.class
14
15 # C extensions
16 *.so
17
18 # Distribution / packaging
19 .Python
20 build/
21 develop-eggs/
22 dist/
23 downloads/
24 eggs/
```

Рисунок 1.3 – Дополнение файла gitignor

```
Git CMD
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
receiving objects: 100% (4/4), done.

c:\lbrtt_2.5>cd/d c:\lbrtt_2.5\lbrtt_2.5

c:\lbrtt_2.5\lbrtt_2.5>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

c:\lbrtt_2.5\lbrtt_2.5>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:\lbrtt_2.5\lbrtt_2.5\.git\hooks]

c:\lbrtt_2.5\lbrtt_2.5>
```

Рисунок 1.4 – Организация репозитория по модели ветвления git-flow

2. Был создана папка PyCharm в которой хранятся примеры из лабораторной работы.

...
Папка с файлами	19.11.2022 10:54	.git
Папка с файлами	19.11.2022 10:55	PyCharm
Исходный файл Git I...	19.11.2022 10:40	.gitignore
Файл	19.11.2022 10:40	LICENSE

Рисунок 2.1 – Папка PyCharm для примеров

```
C:\Users\мвидео\PycharmProjects\pythonProject41\venv\Scripts\python.exe
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 5 варианту.

Задание для 5 варианта

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()))
    k = 0
    for ind, val in enumerate(a):
        if k == 1:
            print(val, " ")
        if a[ind - 1] == val:
            k = 1
```

```
indiv1 x
"C:\Program Files\Python310\python.exe" C:\lbrtt_2.5\lbrtt_2.5\indiv\indiv1.py
1 2 5 3 2 2 5 6 2 2 1 8
5
6
2
2
1
8
```

Рисунок 3.1 – Результат работы программы

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

```

c:\lbrtt_2.5\lbrtt_2.5>git add .
c:\lbrtt_2.5\lbrtt_2.5>git commit -m "new folders"
[develop 1493dc6] new folders
 2 files changed, 30 insertions(+)
 create mode 100644 Pycharm/examp1.py
 create mode 100644 indiv/indiv1.py
c:\lbrtt_2.5\lbrtt_2.5>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
c:\lbrtt_2.5\lbrtt_2.5>git merge develop
Updating f9721d0..1493dc6
Fast-forward
 Pycharm/examp1.py | 18 ++++++
 indiv/indiv1.py   | 12 ++++++
 2 files changed, 30 insertions(+)
 create mode 100644 Pycharm/examp1.py
 create mode 100644 indiv/indiv1.py
c:\lbrtt_2.5\lbrtt_2.5>git push

```

Рисунок 4.1 – Коммит изменений и слияние веток main и develop

```

:\lbrtt_2.5\lbrtt_2.5>git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 1.28 KiB | 1.28 MiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/meiokava/lbrtt_2.5.git
 1900400..d6e3d62 main -> main
:\lbrtt_2.5\lbrtt_2.5>

```

1. Что такое кортежи в языке Python?

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

- $T2$ – новый кортеж, который получается из кортежа $T1$;
- $T1$ – исходный кортеж, для которого происходит срез;
- i, j – соответственно нижняя и верхняя границы среза. Фактически

берутся во внимание 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.