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 №8

Дисциплина: «Основы кроссплатформенного программирования»

Тема: «Работа со списками в языке Python»

Выполнил: студент 2 курса

Группы ППЖ-б-о-21-1

Шарипов Сафа

Выполнение работы:

1. Создала репозиторий в GitHub «rep 2.5» в который добавила .gitignore, который дополнила правила для работы с IDE PyCharm с ЯП Python, выбрала лицензию МІТ, клонировала его на лок. сервер и организовала в соответствии с моделью ветвления git-flow.

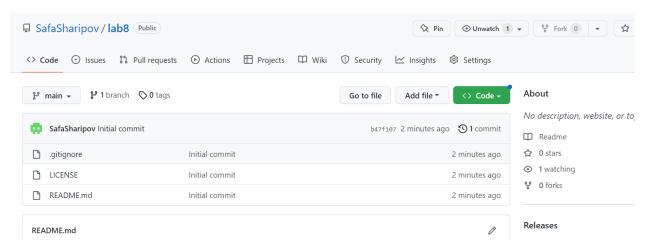


Рисунок 1.1 Создание репозитория

```
C:\Users\User\Desktop\safa\lab8>git clone https://github.com/SafaSharipov/lab8.git
Cloning into 'lab8'...
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.
```

Рисунок 1.2 Клонирование репозитория

```
X
🗐 .gitignore – Блокнот
Файл Правка Формат Вид Справка
.idea/
# Created by https://www.toptal.com/developers/gitignore/api/python,pycharm
# Edit at https://www.toptal.com/developers/gitignore?templates=python,pycharm
### PyCharm ###
# Covers JetBrains IDEs: IntelliJ, RubyMine, PhpStorm, AppCode, PyCharm, CLion, Androi
# Reference: https://intellij-support.jetbrains.com/hc/en-us/articles/206544839
# User-specific stuff
.idea/**/workspace.xml
.idea/**/tasks.xml
.idea/**/usage.statistics.xml
.idea/**/dictionaries
.idea/**/shelf
# AWS User-specific
.idea/**/aws.xml
```

Рисунок 1.4 Изменение .gitignore

2. Создала проект РуCharm в папке репозитория, проработала примеры ЛР.

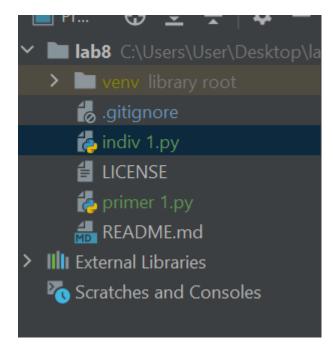


Рисунок 2.1 Создание проекта в PyCharm



Рисунок 2.2 Рез-т выполнения программы

3. Выполнил индивидуальное задание. Вариант 23

```
Введите количество элементов кортежа: 6
Ведите элементы списка:

2
1
5
9
-8
-4
а = (2, 1, 5, 9, -8, -4)
b = (4, 1, 10, 81, -16, 16)
```

Рисунок 3.1 Вывод программы индивидуального задания

4. Сделала коммит, выполнила слияние с веткой main, и запушила изменения в уд. репозиторий.

```
C:\Users\User\Desktop\lab8\lab8>git add .
C:\Users\User\Desktop\lab8\lab8>git commit -m "d"
[develop 1049896] d
3 files changed, 190 insertions(+), 3 deletions(-)
create mode 100644 "\320\230\320\275\320\264\320\270\320\265\indiv 1.py"
create mode 100644 "\320\237\321\200\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\2
```

Рисунок 4.1 коммит изменений и переход на ветку main

Рисунок 4.2 Слияние ветки main c develop

Рисунок 4.3 Пуш изменений на удаленный сервер

Контр. вопросы и ответы на них:

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;
- Т1 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?

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.