

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №8
дисциплины
«Основы кроссплатформенного программирования»

Выполнил:
Бакулин Вадим Романович
2 курс, группа ИТС-б-о-23-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи», очная
форма обучения

(подпись)

Проверил:
Доцент департамента цифровых,
робототехнических систем и
электроники Воронкин Р.А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Тема: Работа с кортежами в языке Python

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

Ссылка на GitHub: <https://github.com/zepteloid/LR8>

Порядок выполнения работы:

1. Изучил теоретический материал.
2. Приступил к выполнению заданий.
3. Создание репозитория

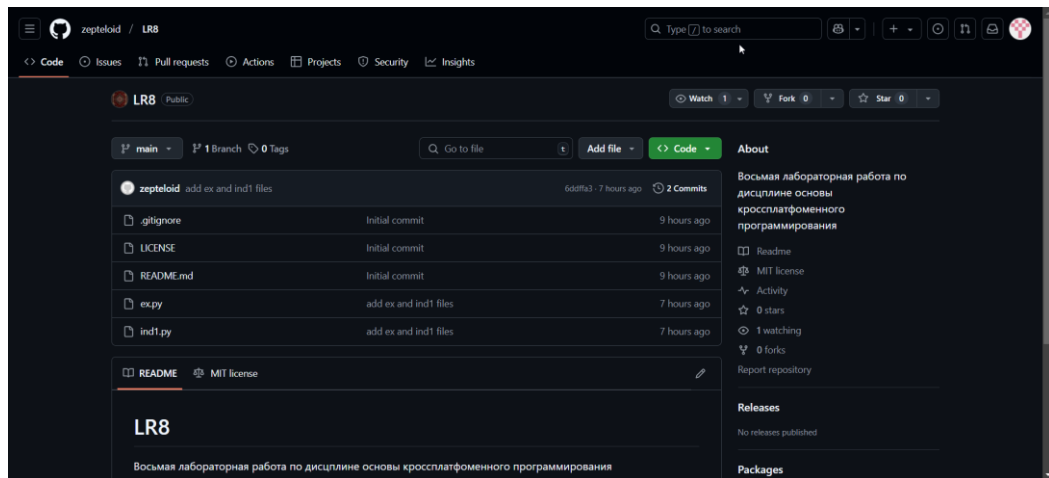


Рисунок 1. Репозиторий

4. Проработал примеры

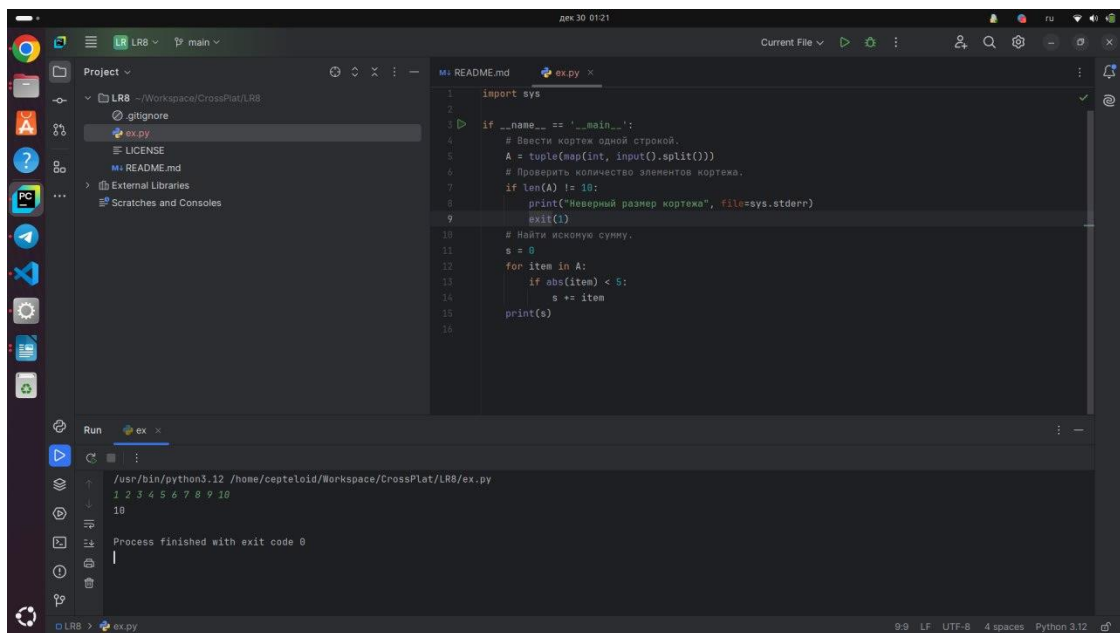


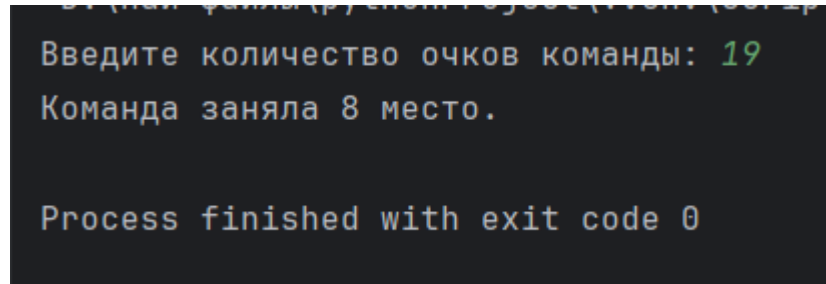
Рисунок 2. Код и вывод программы первого примера

points = (30, 28, 25, 24, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7) # Ввод количества очков команды

```

score = int(input("Введите количество очков команды: ")) # Определяем место
команды position = points.index(score) + 1 # Индекс + 1 = место
print(f"Команда заняла {position} место.")

```



```

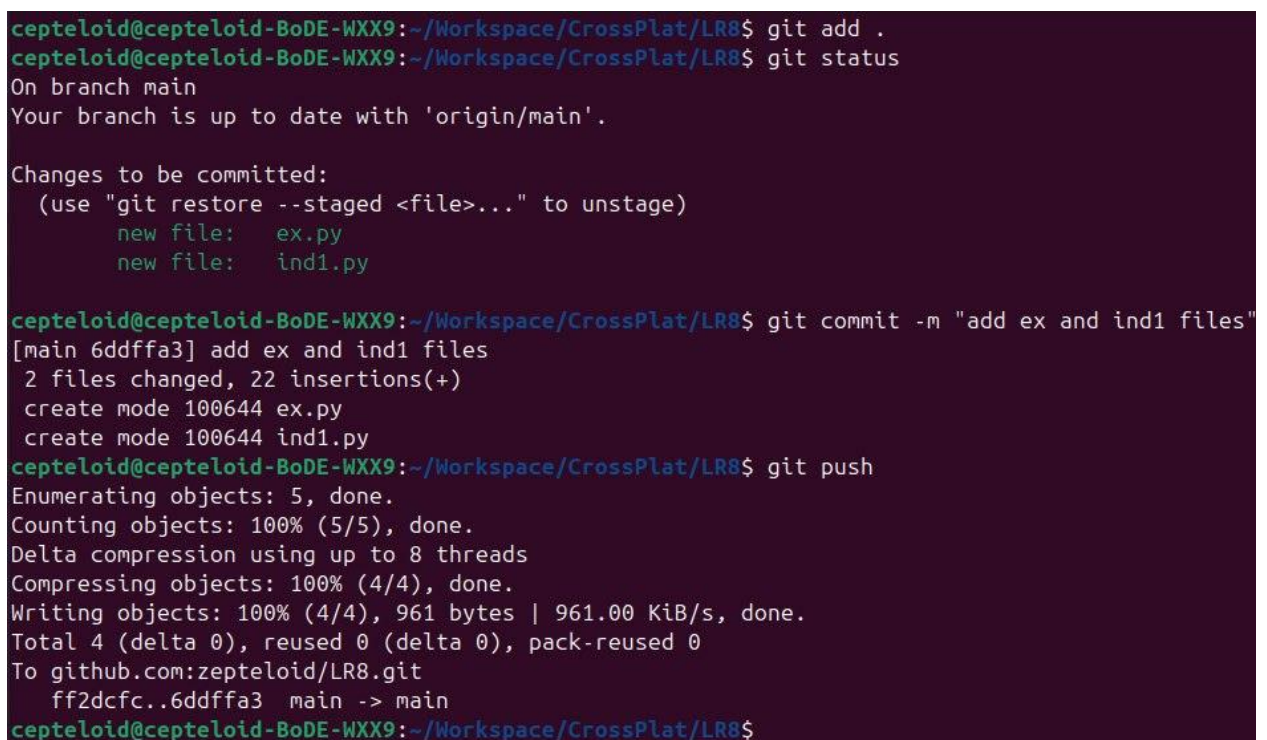
Введите количество очков команды: 19
Команда заняла 8 место.

Process finished with exit code 0

```

Рисунок 3. Вывод задания в консоль

5. Зафиксировал изменения и отправил на удаленный сервер



```

zepteloid@zepteloid-BoDE-WXX9:~/Workspace/CrossPlat/LR8$ git add .
zepteloid@zepteloid-BoDE-WXX9:~/Workspace/CrossPlat/LR8$ 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)
    new file:   ex.py
    new file:   ind1.py

zepteloid@zepteloid-BoDE-WXX9:~/Workspace/CrossPlat/LR8$ git commit -m "add ex and ind1 files"
[main 6ddffa3] add ex and ind1 files
 2 files changed, 22 insertions(+)
 create mode 100644 ex.py
 create mode 100644 ind1.py
zepteloid@zepteloid-BoDE-WXX9:~/Workspace/CrossPlat/LR8$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 961 bytes | 961.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:zepteloid/LR8.git
 ff2dcfc..6ddffa3  main -> main
zepteloid@zepteloid-BoDE-WXX9:~/Workspace/CrossPlat/LR8$

```

Рисунок 4. Фиксация и отправка изменений

Ответы на контрольные вопросы:

1. Что такое кортежи в языке Python?

Кортежи — это неизменяемые последовательности, которые могут содержать элементы различных типов. Они используются для хранения коллекций данных, аналогично спискам, но с тем отличием, что их содержимое нельзя изменить после создания.

2. Каково назначение кортежей в языке Python?

Кортежи используются для хранения фиксированных наборов данных, которые не должны изменяться. Они могут быть полезны для группировки связанных данных и передачи их в функции.

3. Как осуществляется создание кортежей?

Кортежи создаются с помощью круглых скобок (), также можно создать кортеж без скобок, просто перечислив элементы через запятую. Для создания кортежа с одним элементом необходимо добавить запятую.

4. Как осуществляется доступ к элементам кортежа?

Доступ к элементам кортежа осуществляется с помощью индексов, аналогично спискам

5. Зачем нужна распаковка (деструктуризация) кортежа?

Распаковка кортежа позволяет присвоить значения его элементам переменным в одном выражении, что делает код более читаемым и удобным.

6. Какую роль играют кортежи в множественном присваивании?

Кортежи позволяют удобно присваивать несколько значений нескольким переменным одновременно, что упрощает код и делает его более понятным.

7. Как выбрать элементы кортежа с помощью среза?

Срезы для кортежей работают так же, как и для списков: `my tuple[start:end]` вернет элементы с индексами от `start` до `end-1`.

8. Как выполняется конкатенация и повторение кортежей?

Конкатенация осуществляется с помощью оператора `+`, а повторение — с помощью оператора `*`

9. Как выполняется обход элементов кортежа?

Обход элементов кортежа можно осуществить с помощью цикла `for`.

10. Как проверить принадлежность элемента кортежу?

Используя оператор `in` `if element in my tuple`

11. Какие методы работы с кортежами Вам известны?

Кортежи имеют ограниченное количество методов, `count` — подсчитывает количество вхождений элемента. `index` — возвращает индекс первого вхождения элемента.

12. Допустимо ли использование функций агрегации таких как `len`, `sum` и тд при работе с кортежами?

Да, функции агрегации, такие как `len`, `sum`, `min`, `max`, могут использоваться с кортежами так же, как и со списками.

13. Как создать кортеж с помощью спискового включения?

Кортеж можно создать с помощью генератора кортежей, используя круглые скобки

Вывод: в ходе выполнения лабораторной работы я изучил основные особенности использования кортежей в языке программирования Python. Было рассмотрено их применение для хранения неизменяемых последовательностей данных.