

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

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.5
дисциплины
«Основы кроссплатформенного программирования»

Выполнил:
Ашуров Абдуллобек
Одилджонович
1 курс, группа ИТС-б-о-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», заочная форма обучения

(подпись)

Проверил:
Воронкин Р. А., доцент кафедры
инфокоммуникаций

(подпись)

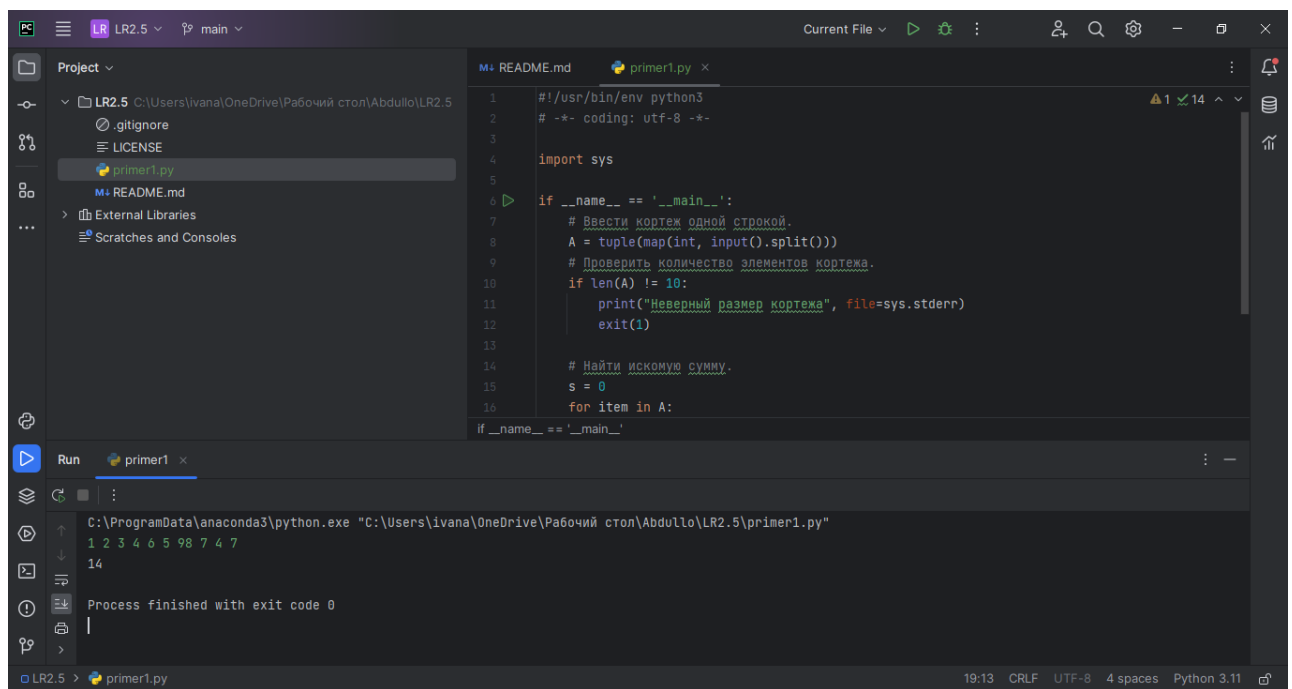
Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 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.

Ход работы:



The screenshot shows a code editor with a project named 'LR2.5'. The file explorer on the left shows the project structure, including 'primer1.py'. The main editor window displays the code for 'primer1.py'. The code is a Python script that takes a string of numbers as input, converts it to a tuple, checks its length, and calculates the sum of its elements. The console output shows the input '1 2 3 4 6 5 98 7 4 7' and the output '14'. The status bar at the bottom indicates the file is 'primer1.py' and the Python version is 3.11.

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5
6 if __name__ == '__main__':
7     # Ввести кортеж одной строкой.
8     A = tuple(map(int, input().split()))
9     # Проверить количество элементов кортежа.
10    if len(A) != 10:
11        print("Неверный размер кортежа", file=sys.stderr)
12        exit(1)
13
14    # Найти искомую сумму.
15    s = 0
16    for item in A:
17
18 if __name__ == '__main__':
```

Run primer1 x

C:\ProgramData\anaconda3\python.exe "C:\Users\ivana\OneDrive\Рабочий стол\Abdullo\LR2.5\primer1.py"

1 2 3 4 6 5 98 7 4 7

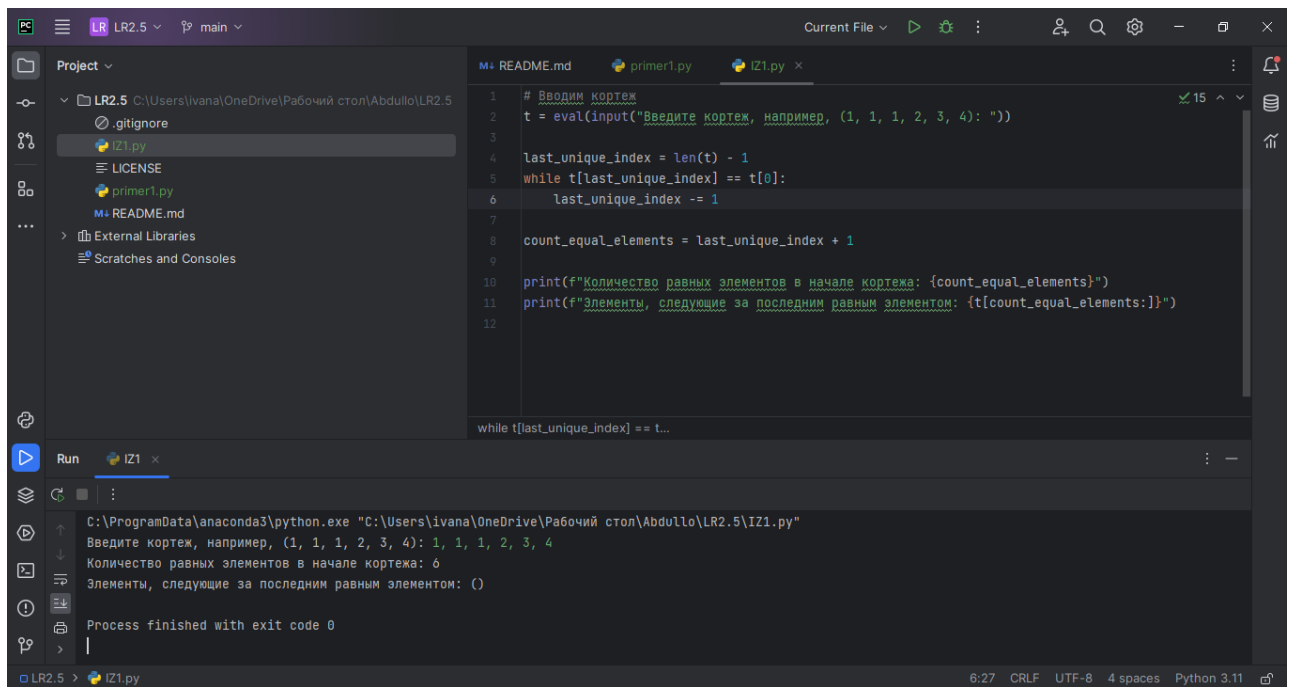
14

Process finished with exit code 0

LR2.5 > primer1.py

Рисунок 1 – Пример 1.

Индивидуальное задание:



The screenshot shows a code editor with a project named 'LR2.5'. The file explorer on the left shows files: .gitignore, LR2.5, IZ1.py, LICENSE, primer1.py, README.md, External Libraries, and Scratches and Consoles. The main editor window displays the code in 'IZ1.py':

```
1 # Вводим кортеж
2 t = eval(input("Введите кортеж, например, (1, 1, 1, 2, 3, 4): "))
3
4 last_unique_index = len(t) - 1
5 while t[last_unique_index] == t[0]:
6     last_unique_index -= 1
7
8 count_equal_elements = last_unique_index + 1
9
10 print(f"Количество равных элементов в начале кортежа: {count_equal_elements}")
11 print(f"Элементы, следующие за последним равным элементом: {t[count_equal_elements:]}")
12
```

The Run console at the bottom shows the execution of the script:

```
C:\ProgramData\anaconda3\python.exe "C:\Users\ivana\OneDrive\Рабочий стол\Abdullo\LR2.5\IZ1.py"
Введите кортеж, например, (1, 1, 1, 2, 3, 4): 1, 1, 1, 2, 3, 4
Количество равных элементов в начале кортежа: 6
Элементы, следующие за последним равным элементом: ()
Process finished with exit code 0
```

Рисунок 2 – ИДЗ 1.

Контрольные вопросы:

1. Что такое кортежи в языке Python?

Список (list) – это структура данных для хранения объектов различных типов. В нем можно хранить объекты различных типов. Размер списка не статичен, его можно изменять. Список по своей природе является изменяемым типом данных. Переменная, определяемая как список, содержит ссылку на

структуру в памяти, которая в свою очередь хранит ссылки на какие-либо другие объекты или структуры.

2. Каково назначение кортежей в языке Python?

Для создания списка нужно заключить элементы в квадратные скобки.

3. Как осуществляется создание кортежей?

При создании списка в памяти резервируется область, которую можно условно назвать некоторым “контейнером”, в котором хранятся ссылки на другие элементы данных в памяти. В отличие от таких типов данных, как число или строка, содержимое “контейнера” списка можно менять.

4. Как осуществляется доступ к элементам кортежа?

Доступ к элементам кортежа осуществляется также как к элементам списка – через указание индекса. Но, как уже было сказано – изменять элементы кортежа нельзя!

5. Зачем нужна распаковка (деструктуризация) кортежа?

Для объединения списков можно использовать оператор сложения (+).

Список можно повторить с помощью оператора умножения (*).

6. Какую роль играют кортежи в множественном присваивании?

Для того, чтобы проверить, есть ли заданный элемент в списке Python необходимо использовать оператор `in`.

7. Как определить число вхождений заданного элемента в списке?

Метод `count` можно использовать для определения числа, сколько раз данный элемент встречается в списке.

8. Как осуществляется добавление (вставка) элемента в список?

Метод `append` можно использовать для добавления элемента в список.

Метод `insert` можно использовать, чтобы вставить элемент в список.

9. Как выполнить сортировку списка?

Для сортировки списка нужно использовать метод `sort`. Для сортировки списка в порядке убывания необходимо вызвать метод `sort` с аргументом `reverse=True`.

10. Как удалить один или несколько элементов из списка?

Удалить элемент можно, написав его индекс в методе `pop`. Если не указывать индекс, то функция удалит последний элемент. Элемент можно удалить с помощью метода `remove`. Оператор `del` можно использовать для тех же целей. Можно удалить несколько элементов с помощью оператора среза. Можно удалить все элементы из списка с помощью метода `clear`.

11. Что такое списковое включение и как с его помощью осуществлять обработку списков?

List Comprehensions чаще всего на русский язык переводят как абстракция списков или списковое включение, является частью синтаксиса языка, которая предоставляет простой способ построения списков. В языке Python есть две очень мощные функции для работы с коллекциями: `map` и `filter`. Они позволяют использовать функциональный стиль программирования, не прибегая к помощи циклов, для работы с такими типами, как `list`, `tuple`, `set`, `dict` и т.п. Списковое включение позволяет обойтись без этих функций.

12. Какие существуют функции агрегации для работы со списками?

Для работы со списками Python предоставляет следующие функции:

1. `len(L)` - получить число элементов в списке `L`.
2. `min(L)` - получить минимальный элемент списка `L`.
3. `max(L)` - получить максимальный элемент списка `L`.
4. `sum(L)` - получить сумму элементов списка `L`, если список `L` содержит только числовые значения.

13. Как создать копию списка?

`copy.copy(x)`

14. Самостоятельно изучите функцию `sorted` языка Python. В чем ее отличие от метода `sort` списков?

Функция `sorted()` в Python возвращает отсортированный список из элементов в итерируемом объекте. `list.sort()` на 13% быстрее, чем `sorted()`.

Вывод: В ходе выполнения лабораторной работы приобретены навыки по работе с кортежами при написании программ с помощью языка программирования Python версии 3.x