

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 №8
дисциплины «Программирование на Python»
Вариант 9

Выполнил:
Дудкин Константин Александрович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»
направление «Программное
обеспечение средств вычислительной
техники и автоматизированных
систем»,
очная форма обучения

(подпись)

Руководитель практики:
Кандидат технических наук, доцент
кафедры инфокоммуникаций, доцент
Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

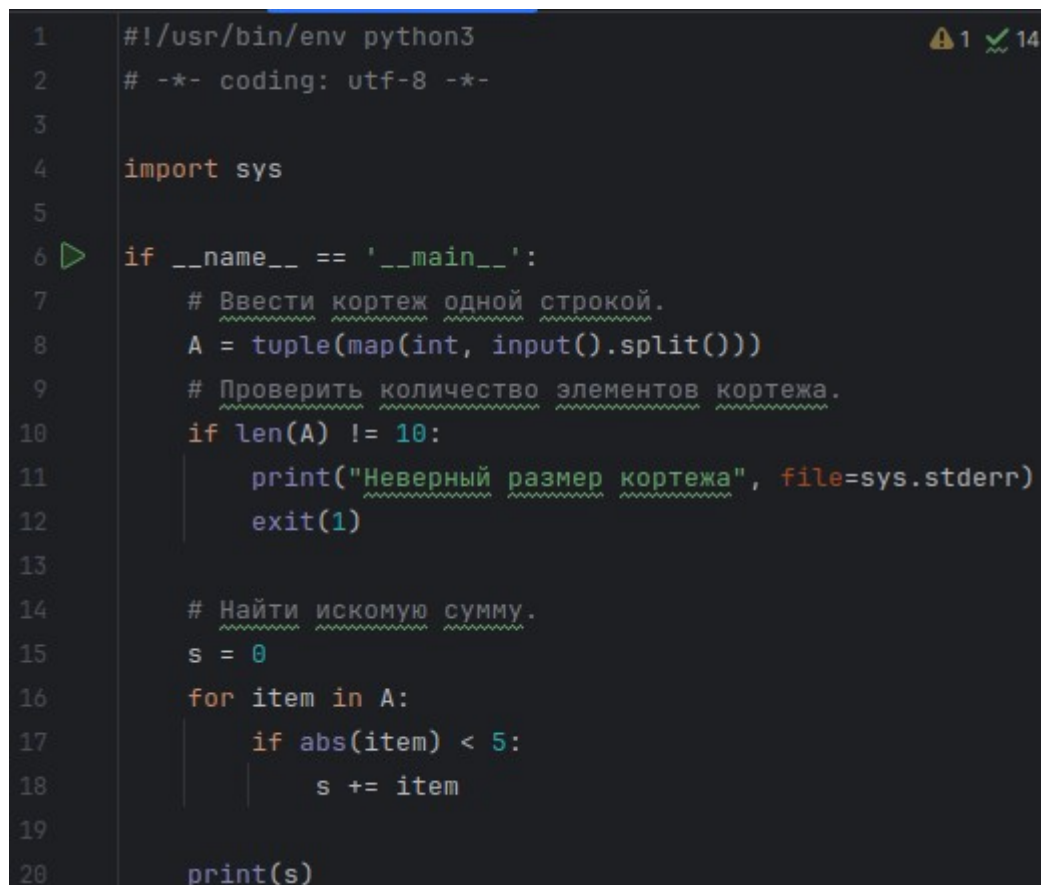
Ставрополь, 2023 г.

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


Порядок выполнения работы

1. Проработал пример: Ввести кортеж A из 10 элементов, найти сумму элементов, меньших по модулю 5, и вывести ее на экран. Использовать в программе вместо списков кортежи



```
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         if abs(item) < 5:
18             s += item
19
20     print(s)
```

Рисунок 1. Код программы примера



```
/usr/bin/python3.11 /home/code_raider/git/Python_LW8/Example.py
5 3 6 8 3 1 7 4 3 8
14
Process finished with exit code 0
```

Рисунок 2. Результат программы

2. Выполнил индивидуальное задание: Если в кортеже есть хотя бы одна тройка соседних чисел, в которой средний элемент больше своих

«соседей» т.е. предшествующего и последующего, то напечатать все элементы, предшествующие элементам последней из таких троек

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5      nums = tuple(map(int, input('Введите ряд из чисел через пробел:').split()))
6      pos = 0
7
8      for i in range(len(nums)):
9          if nums[i] > nums[i - 1] and nums[i] > nums[i + 1]:
10             pos = i
11             if nums[i] == nums[-1]:
12                 for j in range(len(nums)):
13                     if j == pos - 1:
14                         break
15                     else:
16                         print(nums[j])
```

Рисунок 3. Код программы задания

```
/usr/bin/python3.11 /home/code_raider/git/Python_LW8/Individual.py
Введите ряд из чисел через пробел:4 1 5 7 43 0 2 56 7 7 57 2 34 5
4
1
5
7
43
0
2
56
7
7
57
2
34
5
4
1
5
7
43
0
2
56
7
7
57

Process finished with exit code 0
```

Рисунок 4. Результат программы

Ответы на вопросы

1. Список в языке Python представляет собой структуру данных, которая может включать несколько объектов различных типов. Списки позволяют их изменять, добавлять, удалять и проводить другие операции

2. Кортеж в Python — это неизменяемая упорядоченная последовательность элементов. Их используют, когда требуется неизменяемый список данных

3. Для создания кортежей можно использовать круглые скобки (`tuple = (1, 2, 3)`) или метод `tuple()` (`tuple = tuple(1, 2, 3)`)
4. Доступ к элементам кортежа не отличается от доступа к элементам списка — все так же применяются индексы, указанные в квадратных скобках
5. Распаковка кортежа позволяет присваивать значения элементов кортежа переменным одновременно: `a, b, c = tuple`
6. Кортежи позволяют эффективно и удобно присваивать значения нескольким элементам одновременно
7. Выбор нескольких элементов с помощью среза ничем не отличается от выбора нескольких элементов в списке
8. Кортежи могут быть объединены с помощью оператора `+` для конкатенации или оператора `*` для повторения: `concatenated_tuple = tuple1 + tuple2`, `repeated_tuple = tuple1 * tuple2`
9. Обход элементов кортежа, как в списках, можно выполнить посредством цикла `for`: `for i in tuple`
10. Для этого можно использовать метод `in`
11. В кортежах можно выполнить подсчет вхождений элемента с помощью `count()` или узнать индекс первого вхождения элемента с помощью метода `index()`
12. Да, функции агрегации применимы для работы с кортежами, ведь в кортежах нельзя только изменять сами элементы, а не выполнять операции с ними
13. Можно использовать следующую функцию: `my_tuple = tuple(x for x in my_list)`

Вывод. В ходе выполнения лабораторной работы были изучены кортежи, а также методы работы с ними посредством программирования на языке программирования Python версии 3.x