

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 7

Выполнил:
Кулешов Олег Иванович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023

Тема: Работа с кортежами в языке Python

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

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

Листинг к примеру №1:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    # Ввести кортеж одной строкой.
    A = tuple(map(int, input().split()))
    # Проверить количество элементов кортежа.
    if len(A) != 10:
        print('Неверный размер кортежа', file=sys.stderr)
        exit(1)

    # Найти искомую сумму.
    s = 0
    for item in A:
        if abs(item) < 5:
            s += item

    print(s)
```

```
C:\Users\User\PycharmProjects\Python_laba_8\venv\Scripts\python.exe
1 2 3 4 5 6 7 8 9 10
10
Process finished with exit code 0
```

Рисунок 1. Тест №1

```
C:\Users\User\PycharmProjects\Python_laba_8\venv\Scripts\python.exe
-1 3 4 -5 -6 -10 2 3 4 1
16
Process finished with exit code 0
```

Рисунок 2. Тест №2

```
C:\Users\User\PycharmProjects\Python_laba_8\venv\Scripts\python.exe
1 1 1 -1 6 7 8
Неверный размер кортежа
Process finished with exit code 1
```

Рисунок 3. Тест №3

Данная задача может быть также решена с помощью списковых включений следующим образом:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == "__main__":
    # Ввести список одной строкой.
    A = list(map(int, input().split()))
    # Проверить количество элементов списка.
    if len(A) != 10:
        print('Неверный размер списка', file=sys.stderr)
        exit(1)

    # Найти искомую сумму.
    s = sum(a for a in A if abs(a) < 5)
    print(s)
```

Индивидуальное задание. Дан кортеж целых чисел. Если в нем есть хотя бы одна пара соседних четных чисел, то напечатать все элементы, предшествующие элементам последней из таких пар.

Листинг к индивидуальному заданию:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == "__main__":
    my_tuple = tuple(map(int, input("Enter the elements of the tuple
separated by a space: ").split()))

    last_even_index = -1

    for i in range(len(my_tuple) - 1):
        if my_tuple[i] % 2 == 0 and my_tuple[i + 1] % 2 == 0:
            last_even_index = i

    if last_even_index != -1:
        temp_result = my_tuple[:last_even_index]
        result = (' '.join(map(str, temp_result)))

        if result == '':
            print('There are no elements before the last pair of adjacent
even numbers.')
        else:
            print(result)
    else:
        print("There are no pairs of adjacent even numbers in the tuple!",
file=sys.stderr)
```

```
C:\Users\User\PycharmProjects\Python_laba_8\venv\Scripts\python.exe C
Enter the elements of the tuple separated by a space: 2 2 2 2
2 2
Process finished with exit code 0
```

Рисунок 4. Тест №1

```
C:\Users\User\PycharmProjects\Python_laba_8\venv\Scripts\python.exe C
Enter the elements of the tuple separated by a space: 1 1 1 1
There are no pairs of adjacent even numbers in the tuple!
Process finished with exit code 0
```

Рисунок 5. Тест №2

```
C:\Users\User\PycharmProjects\Python_laba_8\venv\Scripts\python.exe C
Enter the elements of the tuple separated by a space: 2 2 1
There are no elements before the last pair of adjacent even numbers.
Process finished with exit code 0
```

Рисунок 6. Тест №3

Вывод: в ходе выполнения данной лабораторной работы были приобретены навыки взаимодействия с кортежами при написании программ с помощью языка программирования Python версии 3.x.

Ответы на контрольные вопросы

1. Списки в языке Python представляют собой упорядоченные коллекции элементов, которые могут быть изменены.
2. Кортежи в языке Python используются для хранения неизменяемых коллекций элементов.
3. Для создания кортежа в Python используются круглые скобки и элементы, разделенные запятыми.
4. Доступ к элементам кортежа осуществляется по их индексам, начиная с 0.
5. Распаковка (деструктуризация) кортежа позволяет присвоить значения элементов кортежа отдельным переменным.

6. Кортежи играют важную роль в множественном присваивании, когда значения нескольких переменных могут быть присвоены из кортежа одновременно.

7. Элементы кортежа могут быть выбраны с помощью срезов, указывающих начальный и конечный индексы.

8. Конкатенация кортежей выполняется с помощью оператора "+", а повторение - с помощью оператора "*".

9. Обход элементов кортежа может быть выполнен с помощью цикла `for`.

10. Принадлежность элемента кортежу можно проверить с помощью оператора `"in"`.

11. Некоторые методы работы с кортежами включают `count()` для подсчета количества вхождений элемента и `index()` для поиска индекса элемента.

12. Да, функции агрегации такие как `len()` и `sum()` могут быть использованы при работе с кортежами.

13. Кортеж можно создать с помощью спискового включения, заключив выражение в круглые скобки и добавив цикл для генерации элементов.