

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 ПРАКТИЧЕСКОЙ РАБОТЕ №8
дисциплины «Программирование на Python»

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

(подпись)

Руководитель практики:
Воронкин Р.А., канд. техн. наук,
доцент, доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

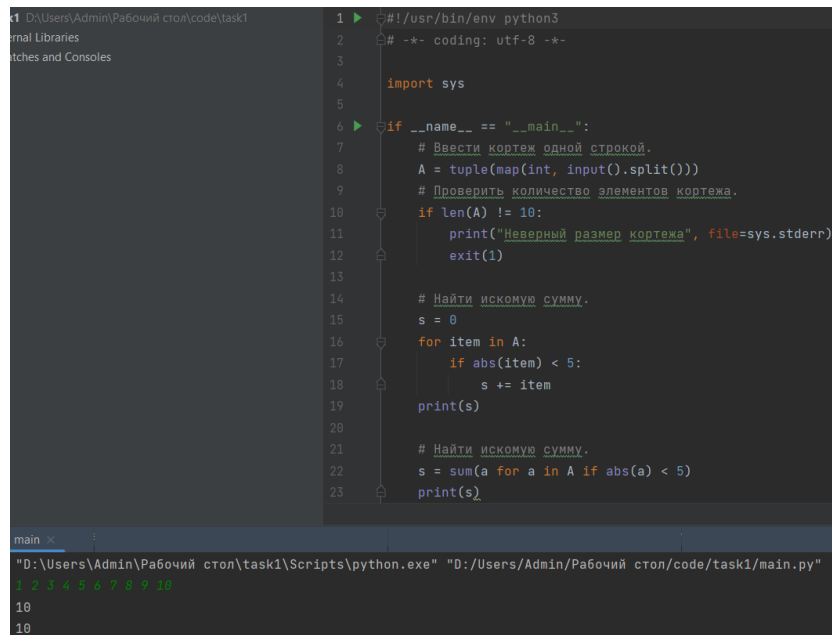
Ставрополь, 2023 г.

Тема: Работа с кортежами в языке Python

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

Порядок выполнения работы:

1. Проработал пример 1, в нем ищется сумма элементов кортежа меньше пяти двумя способами: с помощью перебора и при помощи спискового включения.



```
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    print(s)
20
21    # Найти искомую сумму.
22    s = sum(a for a in A if abs(a) < 5)
23    print(s)
```

main x

"D:\Users\Admin\Рабочий стол\task1\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/code/task1/main.py"

1 2 3 4 5 6 7 8 9 10

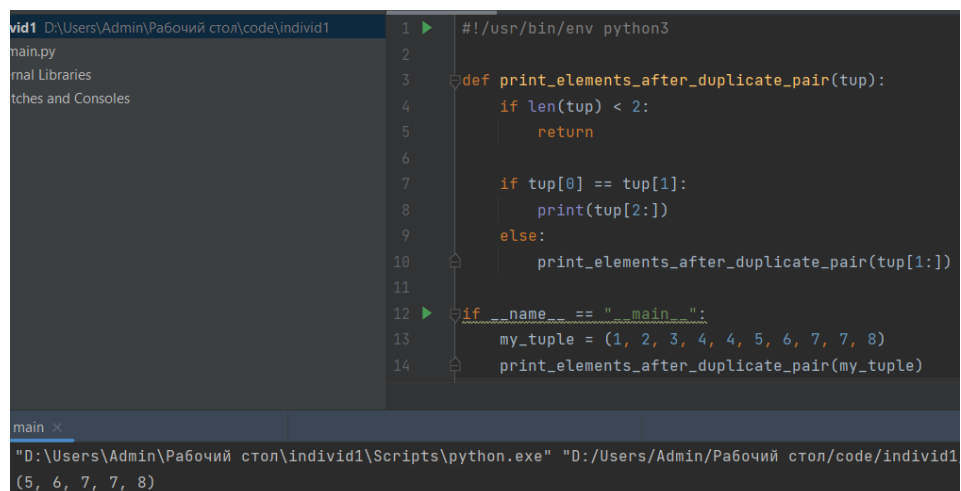
10

10

Рисунок 1 – Результат выполнения программы

Индивидуальное задание (5 Вариант):

Если в кортеже есть хотя бы одна пара одинаковых соседних элементов, то напечатать все элементы, следующие за элементами первой из таких пар.



```
1 #!/usr/bin/env python3
2
3 def print_elements_after_duplicate_pair(tup):
4     if len(tup) < 2:
5         return
6
7     if tup[0] == tup[1]:
8         print(tup[2:])
9     else:
10        print_elements_after_duplicate_pair(tup[1:])
11
12 if __name__ == "__main__":
13     my_tuple = (1, 2, 3, 4, 4, 5, 6, 7, 7, 8)
14     print_elements_after_duplicate_pair(my_tuple)
```

main x

"D:\Users\Admin\Рабочий стол\individ1\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/code/individ1/main.py"

(5, 6, 7, 7, 8)

Рисунок 2 – Результат выполнения программы

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

Контрольные вопросы:

1. Что такое списки в языке Python?

Список – это структура данных для хранения объектов 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?

Кортеж (tuple) – это неизменяемая структура данных, которая по своему подобию очень похожа на список. Как вы знаете, список – это изменяемый тип данных. Т.е. в списке можно изменять его элементы. А в кортеже нельзя изменять элементы. Первая причина существования кортежей – обезопасить данные от случайного изменения, вторая – эконом места, они в памяти занимают меньший объём чем списки.

3. Как осуществляется создание кортежей?

Создание кортежей может осуществляться так: 1) a = (), 2) b = tuple(), 3) c = (1, 2, 3, 4, 5)

4. Как осуществляется доступ к элементам кортежа?

Доступ к элементам кортежа осуществляется также, как и к элементам списка – через указания индекса.

5. Зачем нужна распаковка (деструктуризация) кортежа?

Обращение по индексу, это не самый удобный способ работы с кортежами. Можно забыть, что в каком индексе лежит, но есть способ другой: деструктуризация – кортеж можно собрать и разобрать, пример:

```
name_and_age = ("Bob", 42)
(name, age) = name_and_age
name # "Bob"
```

age # 42

Какую роль играют кортежи в множественном присваивании?

Так как легко собирать и разбирать кортежи, в Python удобно делать множественное присваивание, при помощи этого можно обменивать значения между двумя переменными.

7. Как выбрать элементы кортежа с помощью среза?

Для выбора элементов кортежа с помощью среза в Python можно использовать следующий синтаксис: кортеж[начало:конец:шаг].

8. Как выполняется конкатенация и повторение кортежей?

Конкатенация кортежей в Python выполняется с помощью оператора +. Он объединяет два кортежа в один, создавая новый кортеж, содержащий все элементы обоих кортежей.

9. Как выполняется обход элементов кортежа?

Обход элементов кортежа в Python можно выполнить с помощью цикла for. В каждой итерации цикла переменная принимает значение одного из элементов кортежа.

10. Как проверить принадлежность элемента кортежу?

Для проверки принадлежности элемента кортежу в Python можно использовать оператор in. Он возвращает значение True, если элемент присутствует в кортеже, и False в противном случае.

11. Какие методы работы с кортежами Вам известны?

В Python кортежи являются неизменяемыми объектами, поэтому у них нет множества методов, как у списков. Однако, есть несколько полезных функций, которые можно использовать с кортежами:

12. Допустимо ли использование функций агрегации таких как len(), sum() и т. д. при работе с кортежами?

Да, использование функций агрегации, таких как len(), sum(), min(), max() и т.д., полностью допустимо при работе с кортежами в Python.

13. Как создать кортеж с помощью спискового включения?

В Python можно создать кортеж с помощью спискового включения, а затем преобразовать его в кортеж с помощью функции `tuple()`.