

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 «Программирование на Python»

Выполнил:
Магдаев Даламбек Магомедович
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., доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

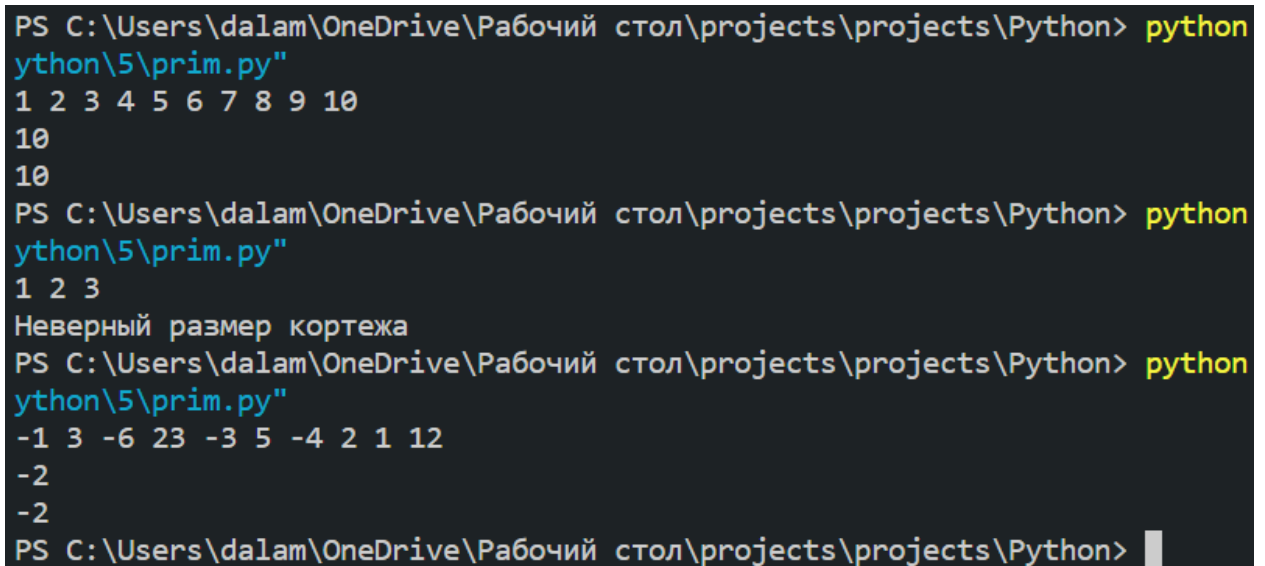
Ставрополь, 2023 г.

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.

Порядок выполнения работы:

1. Создал новый репозиторий, клонировал его, в нем создал ветку developer и перешел на нее.
2. Проработал пример лабораторной работы:



```
PS C:\Users\dalam\OneDrive\Рабочий стол\projects\projects\Python> python  
ython\5\prim.py"  
1 2 3 4 5 6 7 8 9 10  
10  
10  
PS C:\Users\dalam\OneDrive\Рабочий стол\projects\projects\Python> python  
ython\5\prim.py"  
1 2 3  
Неверный размер кортежа  
PS C:\Users\dalam\OneDrive\Рабочий стол\projects\projects\Python> python  
ython\5\prim.py"  
-1 3 -6 23 -3 5 -4 2 1 12  
-2  
-2  
PS C:\Users\dalam\OneDrive\Рабочий стол\projects\projects\Python>
```

Рисунок 1. Несколько запусков программы примера

3. Выполнил индивидуальное задание вариант 9: 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.

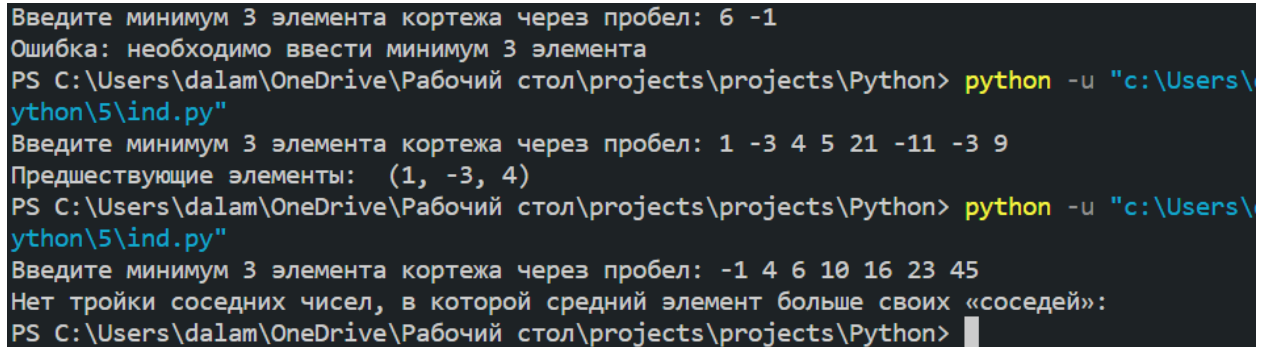
Код программы:

```
#!/usr/bin/env  
python3# -*- coding:  
utf-8 -*-  
import sys  
if __name__ == "__main__":  
    input_tuple = tuple(map(int, input("Введите минимум 3 элемента кортежа через  
    пробел: ").split()))  
    if len(input_tuple) < 3:  
        print("Ошибка: необходимо ввести минимум 3 элемента", file=sys.stderr)  
        exit(1)
```

```

found_triplet = False
for i in range(len(input_tuple) - 2):
    if input_tuple[i] < input_tuple[i+1] > input_tuple[i+2]:
        result = input_tuple[:i] if i > 0 else input_tuple[:i+1]
        print("Предшествующие элементы: ", result)
        foundt = True
        break
if not found:
    print("Нет тройки соседних чисел, в которой средний элемент больше своих «соседей»")

```



```

Введите минимум 3 элемента кортежа через пробел: 6 -1
Ошибка: необходимо ввести минимум 3 элемента
PS C:\Users\dalam\OneDrive\Рабочий стол\projects\projects\Python> python -u "c:\Users\dalam\OneDrive\Рабочий стол\projects\projects\Python\5\ind.py"
Введите минимум 3 элемента кортежа через пробел: 1 -3 4 5 21 -11 -3 9
Предшествующие элементы: (1, -3, 4)
PS C:\Users\dalam\OneDrive\Рабочий стол\projects\projects\Python> python -u "c:\Users\dalam\OneDrive\Рабочий стол\projects\projects\Python\5\ind.py"
Введите минимум 3 элемента кортежа через пробел: -1 4 6 10 16 23 45
Нет тройки соседних чисел, в которой средний элемент больше своих «соседей»:
PS C:\Users\dalam\OneDrive\Рабочий стол\projects\projects\Python>

```

Рисунок 4. Несколько запусков программы индивидуального задания

Ответы на контрольные вопросы:

1. Что такое кортежи в языке Python?

Кортеж (tuple) – это неизменяемая структура данных, которая по своему подобию очень похожа на список.

2. Каково назначение кортежей в языке Python?

Существует несколько причин, по которым стоит использовать кортежи вместо списков. Одна из них – это обезопасить данные от случайного изменения. Если мы получили откуда-то массив данных, и у нас есть желание поработать с ним, но при этом непосредственно менять данные мы не собираемся, тогда, это как раз тот случай, когда кортежи придутся как нельзя кстати.

Используя их в данной задаче, мы дополнительно получаем сразу несколько бонусов – во-первых, это экономия места. Дело в том, что кортежи в памяти занимают меньший объем по сравнению со списками.

Во-вторых – прирост производительности, который связан с тем, что

кортежи работают быстрее, чем списки (т. е. на операции перебора элементов и т. п. будет тратиться меньше времени). Важно также отметить, что кортежи можно использовать в качестве ключа у словаря.

3. Как осуществляется создание кортежей?

Для создания пустого кортежа можно воспользоваться следующими командами: `a = ()`, `b = tuple()`. При желании можно воспользоваться функцией `tuple()`: `a = tuple([1, 2, 3, 4])`.

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?

Распаковка (или деструктуризация) кортежа позволяет удобно извлекать значения из кортежа и присваивать их переменным. Этот метод обеспечивает более чистый и понятный способ работы с кортежами, чем явное обращение к индексам. Распаковка кортежа оказывается полезной, когда функция возвращает кортеж значений, и необходимо назначить эти значения отдельным переменным. Также это удобно для обмена значениями переменных без необходимости использования дополнительной временной переменной. Распаковка кортежей делает код более читаемым и удобным для работы.

6. Какую роль играют кортежи в множественном присваивании?

Используя множественное присваивание, можно провернуть интересный трюк: обмен значениями между двумя переменными.

`(a, b) = (b, a)`

7. Как выбрать элементы кортежа с помощью среза?

С помощью операции взятия среза можно получить другой кортеж.

Общая форма операции взятия среза для кортежа следующая: $T2 = T1[i:j]$.

8. Как выполняется конкатенация и повторение кортежей?

Для кортежей можно выполнять операцию конкатенации, которая обозначается символом $+$. В простейшем случае для конкатенации двух кортежей общая форма операции, следующая: $T3 = T1 + T2$. Кортеж может быть образован путем операции повторения, обозначаемой символом $*$. При использовании в выражении общая форма операции, следующая: $T2 = T1 * n$.

9. Как выполняется обход элементов кортежа?

Элементы кортежа можно последовательно просмотреть с помощью операторов цикла `while` или `for`.

10. Как проверить принадлежность элемента кортежу?

С помощью операции `in`.

11. Какие методы работы с кортежами Вам известны?

Метод `index()` – поиск позиции элемента в кортеже. Метод `count()` – количество вхождений элемента в кортеж.

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

Да.

13. Как создать кортеж с помощью спискового включения.

$B = \text{tuple}(i \text{ for } i \text{ in } A)$

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