

МИНЕСТЕРСТВО НАУКИ ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЁТ
по лабораторной работе №3
Дисциплина: «Языки программирования»

Выполнил: студент 2 курса
группы ИТС-б-о-20-1

Попов Данила Владимирович

Проверил:

к.ф.-м.н., доцент

кафедры инфокоммуникаций

Воронкин Роман Александрович

Работа защищена с оценкой: _____

Ставрополь, 2021

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

Порядок выполнения работы:

Ссылка на репозиторий: <https://github.com/12W300/Three.git>

Создан репозиторий Three.

Проработан пример из методического указания.

```
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)
```

Рисунок 1 – код примера

```
7 6 3 4 2 9 8 1 7 5
10
Process finished with exit code 0
```

Рисунок 2 – работа кода

Решена задача 1.

7. Ввести список *A* из 10 элементов, найти произведение отрицательных элементов и вывести его на экран.

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5
6
7 ▶ if __name__ == '__main__':
8     A = tuple(map(int, input().split()))
9     if len(A) != 10:
10         print("Неверный размер списка", file=sys.stderr)
11         exit(1)
12     x = 1
13     for item in A:
14         if item < 0:
15             x *= item
16     print('Произведение отрицательных элементов: ', x)
17
```

Рисунок 3 – код задачи 1

```
-1 2 -3 4 -5 6 -7 8 -9 1
Произведение отрицательных элементов:  -945
Process finished with exit code 0
```

Рисунок 4 – работа кода

Решена задача 2

6. В списке, состоящем из целых элементов, вычислить:

- 1) номер максимального элемента списка;
- 2) произведение элементов списка, расположенных между первым и вторым нулевыми элементами.

```

1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4      import sys
5
6
7  ▶  if __name__ == '__main__':
8      A = tuple(map(int, input().split()))
9      print('максимальный элемент имеет номер ', A.index(max(A)))
10     zero_1 = zero_2 = -1
11     for i, item in enumerate(A):
12         if (item == 0) and (zero_1 != -1) and (zero_2 == -1):
13             zero_2 = i
14         if (item == 0) and (zero_1 == -1):
15             zero_1 = i
16     print('первый нулевой элемент в позиции ', zero_1, ' второй нулевой элемент в позиции ', zero_2)
17     mult = 1
18     for item in A[zero_1 + 1:zero_2]:
19         mult *= item
20     print('произведение элементов между нулевыми элементами ', mult)
21

```

Рисунок 3 – код задачи 2

```

0 3 6 2 0 4
максимальный элемент имеет номер  2
первый нулевой элемент в позиции  0  второй нулевой элемент в позиции  4
произведение элементов между нулевыми элементами  36

Process finished with exit code 0

```

Рисунок 4 – работа кода

Вопросы для защиты работы:

1. Что такое списки 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?

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). Во-третьих – это безопасность данных от случайного изменения.

3. Как осуществляется создание кортежей?

Для создания пустого кортежа можно воспользоваться одной из следующих команд. `>>> a = () >>> b = tuple()` Кортеж с заданным содержанием создается также как список, только вместо квадратных скобок используются круглые.

4. Как осуществляется доступ к элементам кортежа?

Доступ к элементам кортежа осуществляется также как к элементам списка – через указание индекса.

5. Зачем нужна распаковка (деструктуризация) кортежа?

Обращение по индексу, это не самый удобный способ работы с кортежами. Дело в том, что кортежи часто содержат значения разных типов, и помнить, по какому индексу что лежит — очень непросто. Но есть способ лучше! Как мы кортеж собираем, так его можно и разобрать: `name_and_age = ('Bob', 42) (name, age) = name_and_age name # 'Bob' age # 42`

6. Какую роль играют кортежи во множественном присваивании?

Благодаря тому, что кортежи легко собирать и разбирать, в Python удобно делать такие вещи, как множественное присваивание: `(a, b, c) = (1, 2, 3) a # 1 b # 2 c # 3`

7. Как выбрать элементы кортежа с помощью среза?

С помощью операции взятия среза можно получить другой кортеж. Общая форма операции взятия среза для кортежа:

`T2 = T1[i:j]`, где `T2` – новый кортеж, который получается из кортежа `T1`; `T1` – исходный кортеж, для которого происходит срез; `i`, `j` – соответственно нижняя и верхняя границы среза. Фактически берутся ко вниманию элементы, лежащие на позициях `i`, `i+1`, ..., `j-1`. Значение `j` определяет позицию за последним элементом среза.

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?

Для того, чтобы проверить, есть ли заданный элемент в кортеже Python необходимо использовать оператор `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?

Допустимо.

13. Как создать кортеж с помощью спискового включения.

В отличие от выражения `[a for a in A ...]`, которое на выходе дает нам список, выражение `(a for a in A ...)` дает на выходе специальный объект генератора, а не кортеж. Для преобразования генератора в кортеж необходимо воспользоваться вызовом `tuple()`.

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