

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

**Основы кроссплатформенного программирования
Отчет по лабораторной работе №3**

Работа с множествами в языке Python

Выполнила студентка группы
ИТС-б-о-20-1 (2)

Скачедубова А.В « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил к.т.н., доцент

Кафедры инфокоммуникаций

Воронкин Р.А.

(подпись)

г.Ставрополь 2021

Цель работы: приобрести навыки по работе с кортежами при написании программ с помощью языка программирования Python версии 3х.

Ссылка на репозиторий: https://github.com/Any3002/lab_3

Порядок выполнения работы:

1. Изучила теоретический материал.
2. Создала общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.
3. Выполнила клонирование созданного репозитория.
4. Проработала примеры лабораторной работы:
 - обход кортежа в цикле;

```
# Обход кортежа в цикле
# 1. Цикл for
# Заданный кортеж
A = ("abc", "abcd", "bcd", "cde")

# Вывести все элементы кортежа
for item in A:
    print(item)

# 2. Цикл while
# Исходный кортеж - целые числа
A = (-1, 3, -8, 12, -20)

# Вычислить количество положительных чисел
i = 0
k = 0 # количество положительных чисел

while i < len(A):
    if A[i] < 0:
        k = k + 1
    i = i + 1

# Вывести результат
print("k = ", k)

# 3. Обход в цикле for
# Заданный кортеж, содержащий строки
A = ("abc", "ad", "bcd")

# Сформировать новый список из элементов кортежа A,
```

Рисунок 1— Код проработанного примера №1

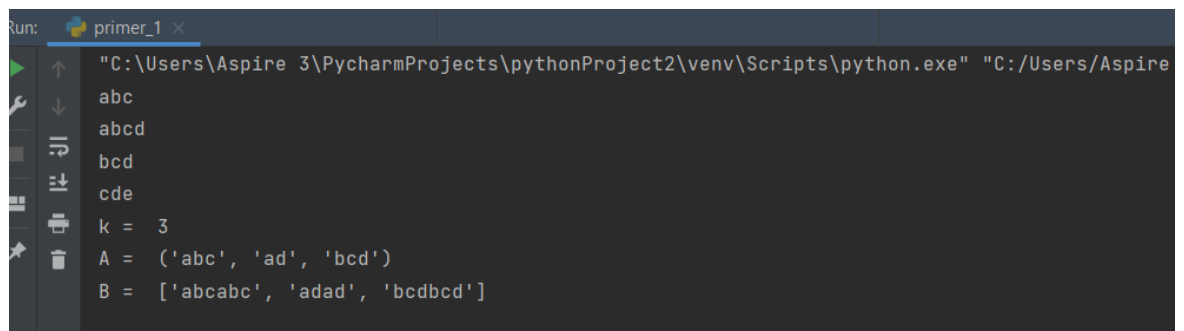
```

# Сформировать новый список из элементов кортежа A,
# в новом списке B, каждый элемент удваивается
B = [item * 2 for item in A]

print("A = ", A)
print("B = ", B)

```

Рисунок 2– Код проработанного примера №1



```

Run: primer_1 x
"C:\Users\Aspire 3\PycharmProjects\pythonProject2\venv\Scripts\python.exe" "C:/Users/Aspire
abc
abcd
bcd
cde
k = 3
A = ('abc', 'ad', 'bcd')
B = ['abcbc', 'adad', 'bcdbcd']

```

Рисунок 3– Результат проработанного примера №1

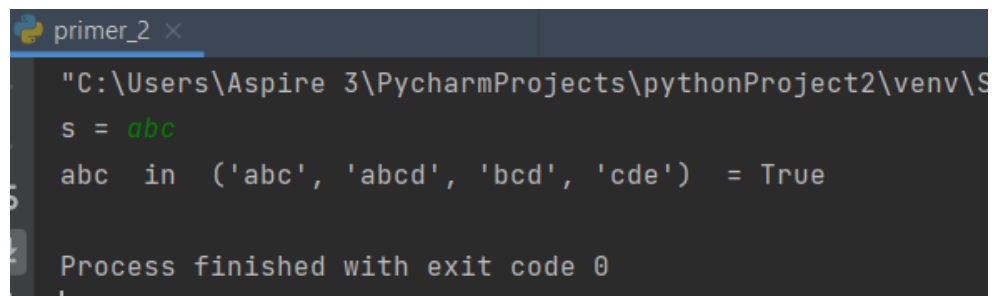
Операция In. Проверка вхождения элемента в кортеж

```

1  # Проверка вхождения элемента в кортеж
2  # Оператор in
3  # Заданный кортеж, который содержит строки
4  A = ("abc", "abcd", "bcd", "cde")
5
6  # Ввести элемент
7  item = str(input("s = "))
8
9  if (item in A):
10     print(item, " in ", A, " = True")
11 else:
12     print(item, " in ", A, " = False")

```

Рисунок 4– Код проработанного примера №2



```

primer_2 x
"C:\Users\Aspire 3\PycharmProjects\pythonProject2\venv\Scripts\python.exe" "C:/Users/Aspire
s = abc
abc in ('abc', 'abcd', 'bcd', 'cde') = True
Process finished with exit code 0

```

Рисунок 5 – Результат проработанного примера №2

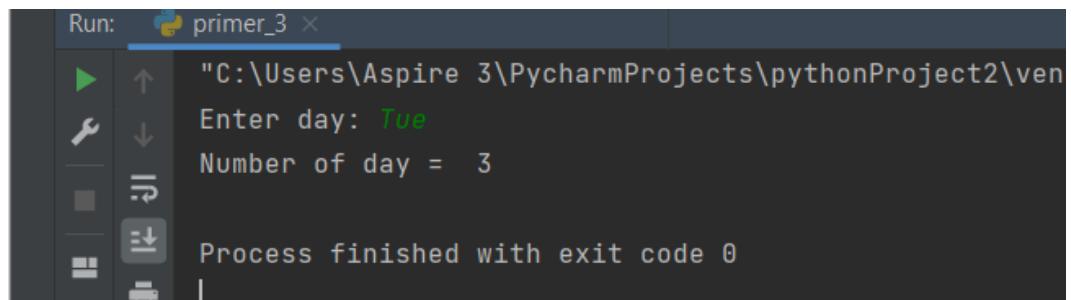
Методы работы с кортежами:

```
# Метод index - определяет позицию (индекс) элемента в кортеже
# Заданный кортеж
A = ("Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat")

# Запрос к вводу названия дня недели
day = str(input("Enter day: "))

# Корректно вычислить индекс
if day in A: # проверка, есть ли строка day в кортеже A
    num = A.index(day)
    print("Number of day = ", num + 1)
else:
    num = -1
    print("Wrong day.")
```

Рисунок 6— Код проработанного примера №3



```
Run: primer_3 x
"C:\Users\Aspire 3\PycharmProjects\pythonProject2\ven
Enter day: Tue
Number of day = 3
Process finished with exit code 0
```

Рисунок 7—Результат проработанного примера №3

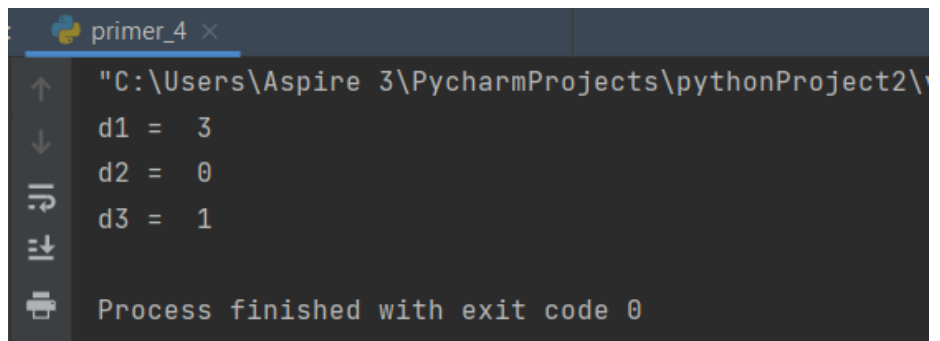
Метод count(). Количество вхождений элемента в кортеж.

```
# Метод count - подсчет количества вхождений элемента в кортеж
# Заданный кортеж
A = ("ab", "ac", "ab", "ab", "ca", "ad", "jklmn")

d1 = A.count("ab") # d1 = 3
d2 = A.count("jprst") # d2 = 0
d3 = A.count("ca") # d3 = 1

print("d1 = ", d1)
print("d2 = ", d2)
print("d3 = ", d3)
```

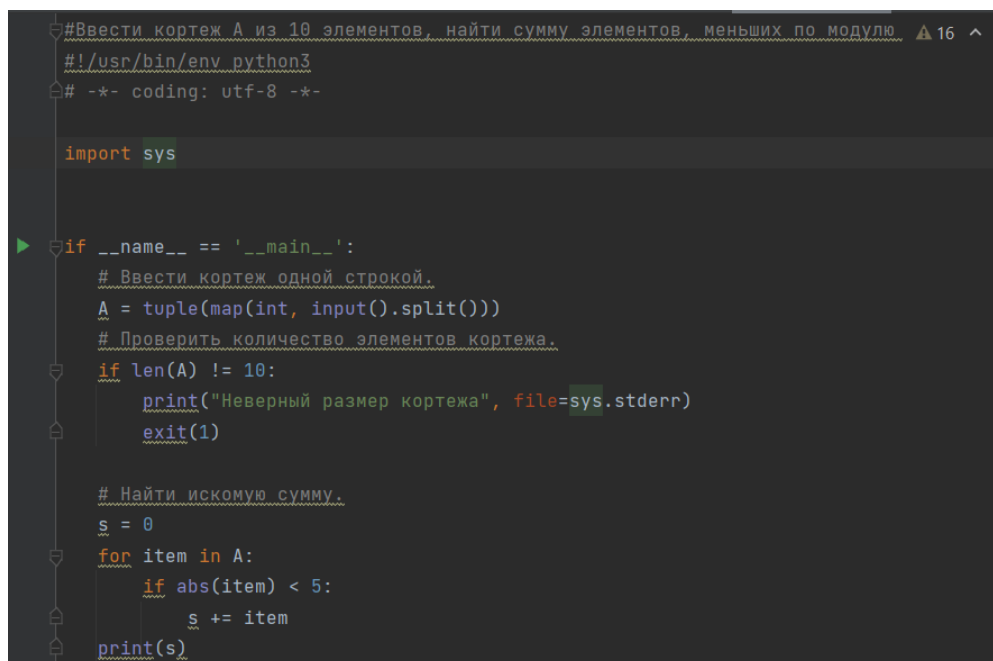
Рисунок 8— Код проработанного примера №4



```
primer_4 x
"C:\Users\Aspire 3\PycharmProjects\pythonProject2\
d1 = 3
d2 = 0
d3 = 1
Process finished with exit code 0
```

Рисунок 9– Результат проработанного примера №4

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



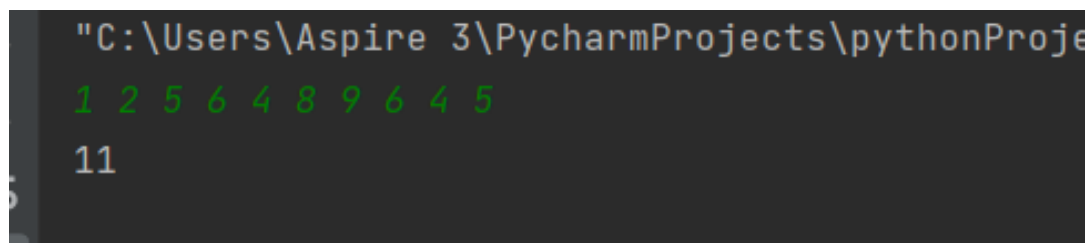
```
#Ввести кортеж A из 10 элементов, найти сумму элементов, меньших по модулю
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

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)
```

Рисунок 10– Код проработанного примера №5



```
"C:\Users\Aspire 3\PycharmProjects\pythonProje
1 2 5 6 4 8 9 6 4 5
11
```

Рисунок 11–Результат проработанного примера №5.1

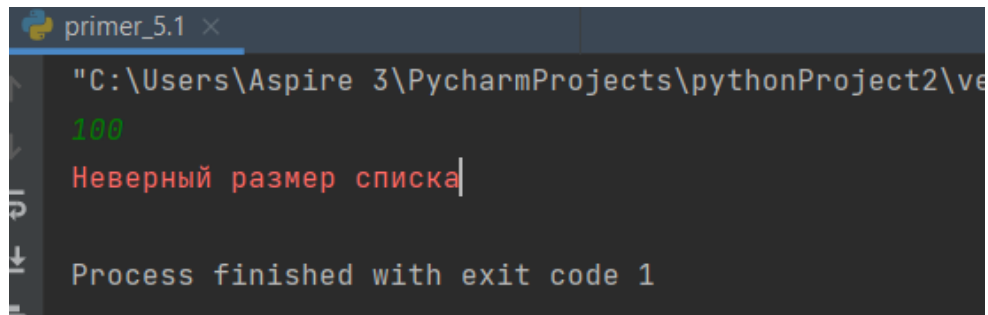
Пример 5.1 Другой способ решения задачи примера 1.1

```

1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4      import sys
5
6
7  ▶  ● if __name__ == '__main__':
8      # Ввести список одной строкой.
9      A = list(map(int, input().split()))
10     # Проверить количество элементов списка.
11     if len(A) != 10:
12         print("Неверный размер списка", file=sys.stderr)
13         exit(1)
14
15     # Найти искомую сумму.
16     s = sum(a for a in A if abs(a) < 5)
17     print(s)
18

```

Рисунок 12–Код проработанного примера №1.2



```

primer_5.1 x
"C:\Users\Aspire 3\PycharmProjects\pythonProject2\ve
100
Неверный размер списка
Process finished with exit code 1

```

Рисунок 13–Результат проработанного примера №1.2

Индивидуальное задание. Вариант 14.

Задание 1. Ввести список A из 10 элементов, найти разность положительных элементов кратных 11, их количество и вывести результаты на экран.

Запишем код программы:

```
primer_1.py x primer_2.py x primer_3.py x primer_4.py x primer_5.py x zadanie_1.py x
#Ввести список A из 10 элементов, найти разность положительных элементов кр
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    # Ввести список 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)

    k = s = 0
    for item in A:
        if item % 11 == 0 and item > 0:
            k -= item
            s += 1

    print(k, '\n', s)
```

Рисунок 14—Код задания №1

```
Run: zadanie_1 x
"C:\Users\Aspire 3\PycharmProjects\pythonPro
1 22 13 4 6 8 44 7 9 6
-66
2
Process finished with exit code 0
```

Рисунок 15—Результат задания №1

Построим диаграмму:

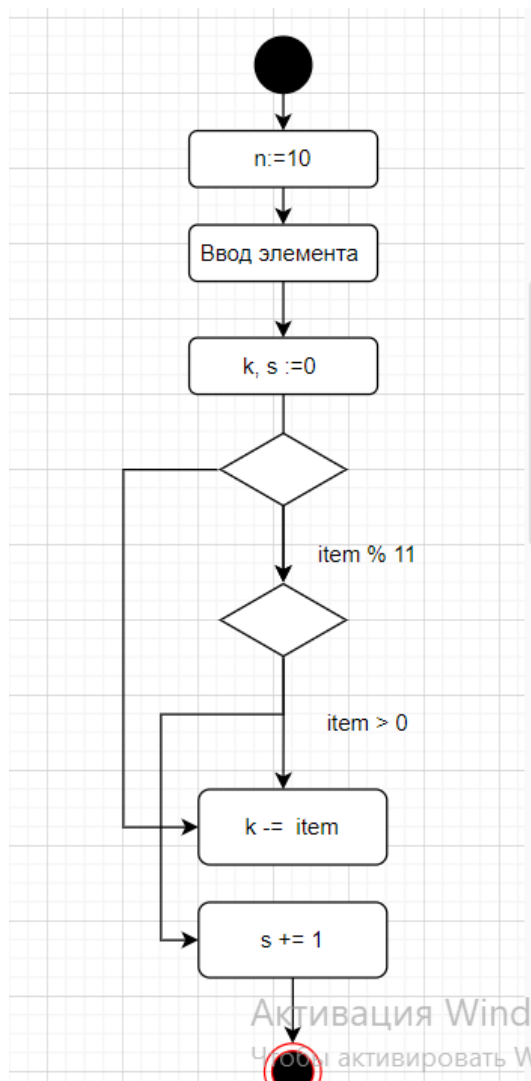


Рисунок 16—UML—диаграмма задания №1

Задание №2 В списке, состоящем из вещественных элементов, вычислить:

- 1) количество элементов списка, больших C ;
- 2) произведение элементов списка, расположенных после максимального по модулю элемента. Преобразовать список таким образом, чтобы сначала располагались все отрицательные элементы, а потом - все положительные (элементы, равные 0, считать положительными).

Запишем код задания:


```
main.py x zadanie_2.py x
# В списке, состоящем из вещественных элементов, вычислить:
#1) количество элементов списка, больших C;
#2) произведение элементов списка, расположенных после максимального по модулю э
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    # Ввести список одной строкой.
    A = tuple(map(int, input().split()))
    C = int(input('Введите C '))

    # Проверить количество элементов кортежа.
    if len(A) != 10:
        print("Неверный размер кортежа", file=sys.stderr)
        exit(1)

    # Количество элементов больших C
    s = 0
    for item in A:
        if item > C:
            s += 1
    print(s)
```

Рисунок 17–Код задания №2

```
# Произведение элементов списка, расположенных после максимального по модулю элемента
pos = 0
mult = 1
if abs(max(A)) > abs(min(A)): pos = A.index(max(A))
else: pos = A.index(min(A))
for i in range(pos + 1, len(A)):
    mult *= A[i]
print(mult)

# Сортированный список
print(tuple(sorted(A)))
```

Рисунок 18–Код задания №2

```
in: zadanie_2 x
"C:\Users\Aspire 3\PycharmProjects\pythonProject2\venv\Scripts\python.exe"
1 2 3 4 5 6 7 8 9 4
Введите C 15
0
4
(1, 2, 3, 4, 4, 5, 6, 7, 8, 9)
```

Рисунок 19–Код задания №2

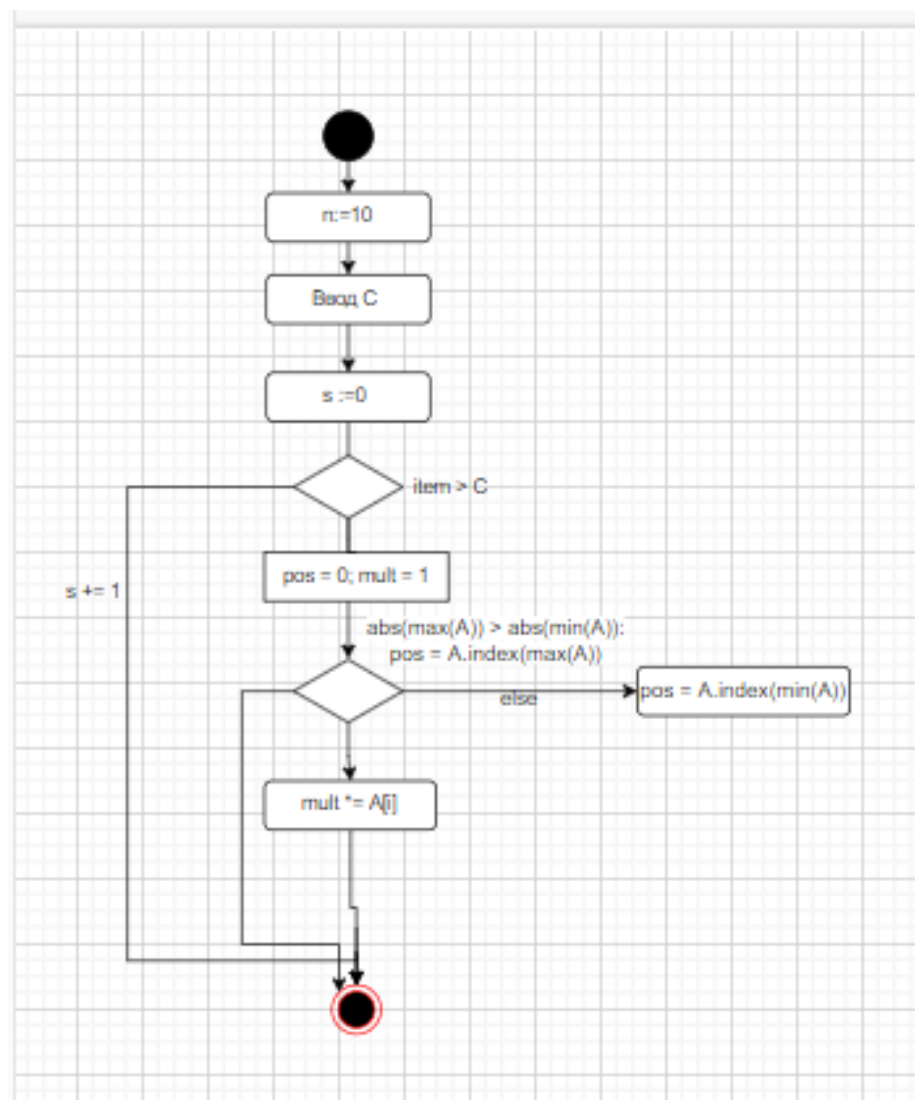


Рисунок 20–UML–диаграмма задания №2

Вопросы для защиты работы

1. Что такое списки в языке Python?

Список –это изменяемый тип данных.

2. Каково назначение кортежей в языке Python?

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

Назначение:

- защищают данные от случайного изменения;
- занимают в памяти меньший объем по сравнению со списками;
- работают быстрее, чем списки;
- можно использовать в качестве ключа у словаря.

3. Как осуществляется создание кортежей?

Для создания кортежей можно воспользоваться функцией `tuple()`, также нужно поставить после элемента кортежа запятую.

4. Как осуществляется доступ к элементам кортежа?

Доступ к элементам кортежа осуществляется через указание индекса.

5. Зачем нужна распаковка (деструктуризация) кортежа?

Распаковка является более удобным способом работы с кортежами, так как кортежи часто содержат значения разных типов, и запомнить, по какому индексу что лежит не просто.

6. Какую роль играют кортежи в множественном присваивании?

При использовании множественного присваивания, можно провернуть обмен значениями между двумя переменными.

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 $T2 = T1[i:j]$, 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 $i, i+1, \dots, j-1$. Значение j определяет позицию за последним элементом среза.

8. Как выполняется конкатенация и повторение кортежей?

Конкатенация обозначается символом $+$.

Общая форма операции: $T3 = T1 + T2$, где

– T1, T2 – кортежи, для которых нужно выполнить операцию конкатенации. Операнды T1, T2 обязательно должны быть кортежами. При выполнении операции конкатенации для кортежей, использовать в качестве операндов любые другие типы (строки, списки) запрещено;

– T3 – кортеж, который есть результатом

Повторение обозначается символом *.

Общая форма операции: $T2 = T1 * n$, где

–T2 – результирующий кортеж;

–T1 – исходный кортеж, который нужно повторить n раз;

–n – количество повторений кортежа T1.

9. Как выполняется обход элементов кортежа?

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

10. Как проверить принадлежность элемента кортежу?

При помощи оператора in.

11. Какие методы работы с кортежами Вам известны?

Метод index()– для получения индекса(позиции) элемента в кортеже.

Метод count()– для того, чтобы определить количество вхождений заданного элемента в кортеж

Метод

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

При помощи команды (a for a in A), которая дает на выходе специальный объект генератора. Для преобразования генератора в кортеж необходимо воспользоваться вызовом tuple() .

Вывод по работе: в ходе лабораторной работы были приобретены навыки по работе с кортежами при написании программ. А также, были

изучены методы работы с кортежами, элементы кортежа с помощью среза, выполнение обхода элементов кортежа и т.д.