

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»

Вариант 10

Выполнил:
Кенесбаев Хилол Куат улы
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 _____

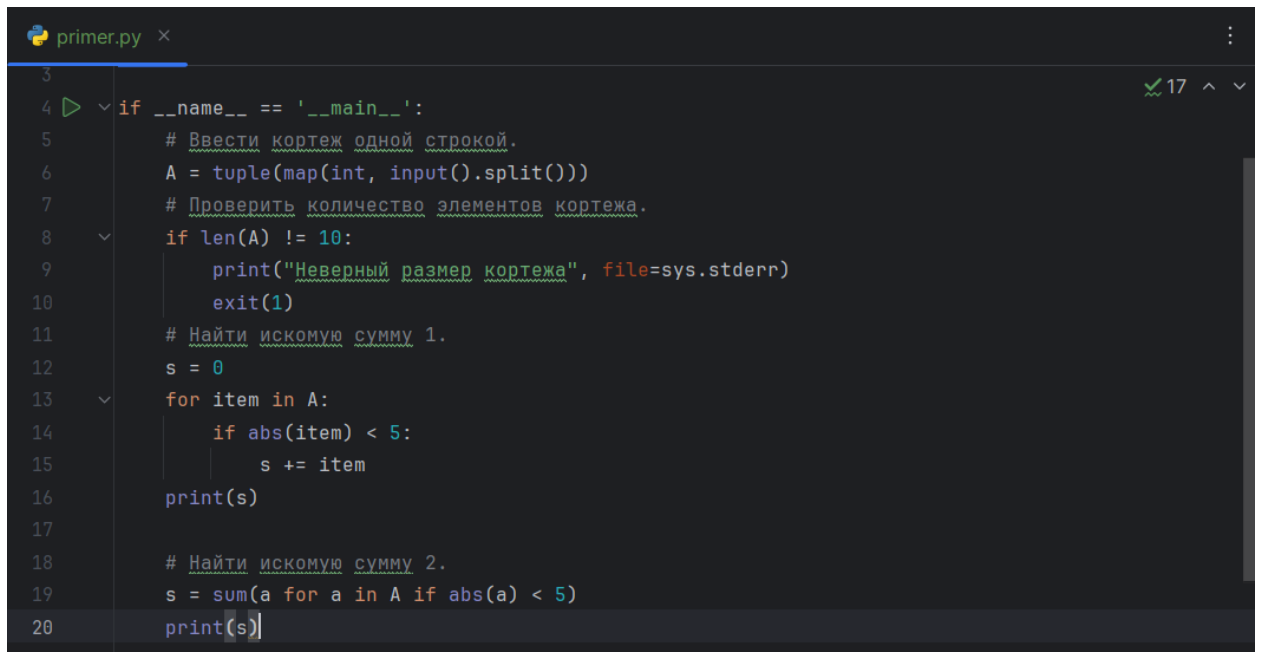
Ставрополь, 2023 г.

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.

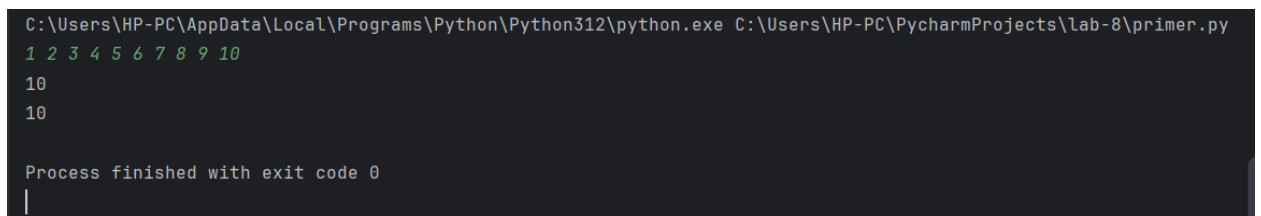
Порядок выполнения работы:

1. Проработал примеры лабораторной работы:



```
3
4 if __name__ == '__main__':
5     # Ввести кортеж одной строкой.
6     A = tuple(map(int, input().split()))
7     # Проверить количество элементов кортежа.
8     if len(A) != 10:
9         print("Неверный размер кортежа", file=sys.stderr)
10        exit(1)
11    # Найти искомую сумму 1.
12    s = 0
13    for item in A:
14        if abs(item) < 5:
15            s += item
16    print(s)
17
18    # Найти искомую сумму 2.
19    s = sum(a for a in A if abs(a) < 5)
20    print(s)
```

Рисунок 1 primer.py



```
C:\Users\HP-PC\AppData\Local\Programs\Python\Python312\python.exe C:\Users\HP-PC\PycharmProjects\lab-8\primer.py
1 2 3 4 5 6 7 8 9 10
10
10

Process finished with exit code 0
|
```

Рисунок 2 Результат работы программы primer.py

Выполнил индивидуальное задание, номер варианта 10:

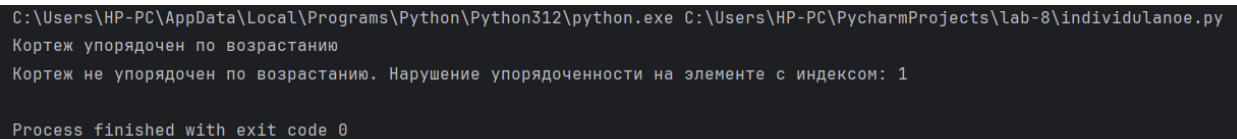
Определить, является ли кортеж упорядоченным по возрастанию. В случае отрицательного ответа определить номер первого элемента, нарушающего такую упорядоченность.

Код программы:

```
def check_ordered_tuple(t):
    # Проверка на упорядоченность по возрастанию
    if all(t[i] <= t[i + 1] for i in range(len(t) - 1)):
        print("Кортеж упорядочен по возрастанию")
    else:
        # Находим номер первого элемента, нарушающего упорядоченность
        index_of_first_unordered = next((i for i in range(len(t) - 1) if t[i] > t[i + 1]),
        None)
        print("Кортеж не упорядочен по возрастанию. Нарушение
        упорядоченности на элементе с индексом:", index_of_first_unordered)
```

```
# Пример кортежей для проверки
tuple1 = (1, 2, 3, 4, 5)
tuple2 = (1, 3, 2, 4, 5)

# Проверяем упорядоченность кортежей
check_ordered_tuple(tuple1)
check_ordered_tuple(tuple2)
```



```
C:\Users\HP-PC\AppData\Local\Programs\Python\Python312\python.exe C:\Users\HP-PC\PycharmProjects\lab-8\individualnoe.py
Кортеж упорядочен по возрастанию
Кортеж не упорядочен по возрастанию. Нарушение упорядоченности на элементе с индексом: 1
Process finished with exit code 0
```

Рисунок 3 Результат работы программы individualnoe.py

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 проверить 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.