

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ
ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ» ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ

Отчет о лабораторной работе №7
по дисциплине основы программной инженерии

Выполнил: Духно Михаил
Александрович,

2 курс, группа ПИЖ-б-о-20-1,

Проверил: Доцент кафедры
инфокоммуникаций, Воронкин Р.А.

Ставрополь, 2021 г

```

a = [1, 2, 3]
print(a)
# [1, 2, 3]
a[1] = 15
print(a)
# [1, 15, 3]

b = (1, 2, 3)
print(b)
# (1, 2, 3)
b[1] = 15
# Traceback (most recent call last):...

```

Рисунок 8.1 – Что такое кортеж в python

```

lst = [10, 20, 30]
tpl = (10, 20, 30)
print(lst.__sizeof__())
# 32
print(tpl.__sizeof__())
# 24

```

Рисунок 8.2 – Зачем нужны кортежи

```

#CREATE TUPLES
a = ()
print(type(a))
#<class 'tuple'>
b = tuple()
print(type(b))
#<class 'tuple'>

a = (1, 2, 3, 4, 5)
print(type(a))
#<class 'tuple'>
print(a)
# (1, 2, 3, 4, 5)

a = tuple([1, 2, 3, 4])
print(a)
# (1, 2, 3, 4)

not_a_tuple = (42)

tuple = (42,)

```

Рисунок 8.3 – Создание кортежа

```

#GET ELEMENTS
a = (1, 2, 3, 4, 5)
print(a[0])
#1
print(a[1:3])
#(2, 3)
a[1] = 3
#Traceback (most recent call last)...

```

Рисунок 8.4 – Доступ к элементам кортежа

```

#DELET TUPLES
a = (1, 2, 3, 4, 5)
del a[0]
#Traceback (most recent call last)...
del a
print(a)
#Traceback (most recent call last)...

```

Рисунок 8.5 – Удаление кортежей

```

lst = [1, 2, 3, 4, 5]
print(type(lst))
#<class 'list'>
print(lst)
#[1, 2, 3, 4, 5]
tpl = tuple(lst)
print(type(tpl))
#<class 'tuple'>
print(tpl)
#(1, 2, 3, 4, 5)

tpl = (2, 4, 6, 8, 10)
print(type(tpl))
#<class 'tuple'>
print(tpl)
#(2, 4, 6, 8, 10)
lst = list(tpl)
print(type(lst))
#<class 'list'>
print(lst)
#[2, 4, 6, 8, 10]

```

Рисунок 8.6 – Преобразование кортежа в список и обратно

```

name_and_age = ('Bob', 42)
(name, age) = name_and_age
name # 'Bob'
age # 42

(a,) = (42,)
a # 42

```

Рисунок 8.7 – Деструктуризация

```

(a, b, c) = (1, 2, 3)
a # 1
b # 2
c # 3

a = 100
b = 'foo'
(a, b) = (b, a)
a # 'foo'
b # 100

```

Рисунок 8.8 – Множественное присваивание и обмен значениями

```

d = tuple('Hello') # d = ('H', 'e', 'l', 'l', 'o')

lst = [2, 'abc', 3.88]
e = tuple(lst) # e = (2, 'abc', 3.88)
f = tuple((3, 2, 0, -5)) # f = (3, 2, 0, -5)

```

Рисунок 8.9 – Создание кортежа из итерированного объекта

```

A = (0, 1, 2, 3)
item = A[0:2] # item = (0, 1)

A = (2.5, ['abcd', True, 3.1415], 8, False, 'z')
item = A[1:3] # item = (['abcd', True, 3.1415], 8)

A = (3, 8, -11, 'program')
B = ('Python', A, True)
item = B[:3] # item = ('Python', (3, 8, -11, 'program'), True)
item = B[1:] # item = ((3, 8, -11, 'program'), True)

```

Рисунок 8.10 – Взятие среза в кортеже

```

A = (1, 2, 3)
B = (4, 5, 6)
C = A + B #C = (1, 2, 3, 4, 5, 6)

D = (3, 'abc') + (-7.22, ['a', 5]) #D = (3, 'abc', -7.22, ['a', 5])

A = ('a', 'aa', 'aaa')
B = A + (1, 2) + (True, False) #B = ('a', 'aa', 'aaa', 1, 2, True, False)

```

Рисунок 8.11 – Конкатенация

```

A = (1, 2, 3) * 3 #A = (1, 2, 3, 1, 2, 3, 1, 2, 3)

B = ('ab', ['1', '12']) * 2 #B = ('ab', ['1', '12'], 'ab', ['1', '12'])

```

Рисунок 8.12 – Повторение

```

A = ('abc', 'abcd', 'bcd', 'cde')
for item in A:
    print(item)

A = (-1, 3, -8, 12, -20)
i = 0
k = 0
while i < len(A):
    if(A[i]<0):
        k = k + 1
    i = i + 1
print('k = ', k)

A = ('abc', 'ad', 'bcd')
B = [item * 2 for item in A]
print('A = ', A)
print('B = ', B)
'''
abc
abcd
bcd
cde
k = 3
A = ('abc', 'ad', 'bcd')
B = ['abcbcb', 'adadad', 'bcdcbcd']
'''

```

Рисунок 8.13 – Обход кортежа в цикле

```

A = ('abc', 'abcd', 'bcd', 'cde')
item = str(input('s ='))
if(item in A):
    print(item, ' in ', A, ' = True')
else:
    print(item, ' in ', A, ' = False')
'''
s = abc
abc in ('abc', 'abcd', 'bcd', 'cde') = True
'''

```

Рисунок 8.14 – Проверка вхождения элемента в кортеж

```

A = ('sun', 'mon', 'tue', 'wed', 'thu', 'fri', 'sat')
day = str(input('Enter day: '))
if day in A:
    num = A.index(day)
    print('Number of day = ', num + 1)
else:
    num = -1
    print('Wrong day')
'''
d = 1
'''

```

Рисунок 8.15 – Поиск позиции элемента в кортеже

```

A = ('ab', 'ac', 'ab', 'ab', 'ca', 'ad', 'jklmn')
d1 = A.count('ab')
d2 = A.count('jprst')
d3 = A.count('ca')

print('d1 = ', d1)
print('d2 = ', d2)
print('d3 = ', d3)
'''
d1 = 3
d2 = 0
d3 = 1
'''

```

Рисунок 8.16 – Количество вхождений элемента в кортеж

```

import sys

if __name__ == '__main__':
    A = tuple(map(int, input().split()))
    if len(A) != 10:
        print('Неверный размер кортежа', file=sys.stderr)
        exit(1)
    s = 0
    for item in A:
        if abs(item) < 5:
            s += item
    print(s)

```

Рисунок 8.17 – Пример программы с кортежем 1

```

import sys

if __name__ == '__main__':
    A = list(map(int, input().split()))
    if len(A) != 10:
        print('Неверный размер кортежа', file=sys.stderr)
        exit(1)
    s = sum(a for a in A if abs(a) < 5)
    print(s)

```

Рисунок 8.18 – Пример программы с кортежем 2

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    tpl = tuple(input('tpl = ').split())

    i = 1
    flag = False

    while i < len(tpl) - 1:
        if int(tpl[i]) > int(tpl[i - 1]) \
            and int(tpl[i]) > int(tpl[i + 1]):
            flag = True
            print(f'ind1 = {i - 1}, ind2 = {i}, ind3 = {i + 1}')
            break
        i += 1

    if not flag:
        print('Таких троек нет')

```

Рисунок 8.19 – Индивидуальное задание, вариант 8

Контрольные вопросы:

1. Что такое списки в языке Python?

Список – это структура данных, которая содержит упорядоченный набор элементов, т. е. хранит последовательность элементов.

2. Каково назначение кортежей в языке Python?

Кортежи неизменяемы и от них можно найти хеш, если они, конечно, сами не содержат изменяемых типов.

Из-за этого кортежи более легковесны, а возможность найти хеш позволяет использовать кортежи в виде ключей для словарей.

Например, можно хранить содержимое некоторых координат массива

```
d = {(0, 1): 'X', (2, 3): 'O'}
```

3. Как осуществляется создание кортежей?

```
tpl = (1, 2, 3, 4, 5)
```

```
tpl = tuple(input())
```

```
tpl = tuple([1, 2, 3, 4])
```

4. Как осуществляется доступ к элементам кортежа?

```
el = tpl[0]
```

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. По этой причине лучше деструктуризировать кортежи на несколько переменных для дальнейшей работы.

6. Какую роль играют кортежи в множественном присваивании?

При помощи кортежа можно присвоить одновременно нескольким переменным разные значения в одной строке

7. Как выбрать элементы кортежа с помощью среза?

```
srez = tpl[1:3]
```

Это всё осуществляется так же как и в списках через синтакс двоеточия

8. Как выполняется конкатенация и повторение кортежей?

Конкатенация выполняется при помощи знака +, повторение при помощи знака *

9. Как выполняется обход элементов кортежа?

Обход может выполняться при помощи цикла while или цикла for

10. Как проверить принадлежность элемента кортежу?

Принадлежность элемента к кортежу можно проверить при помощи in

```
If perem in tpl:
```

11. Какие методы работы с кортежами Вам известны?

.index(<perem>) – возвращает индекс первого вхождения элемента <perem>

.count(<perem>) – возвращает количество вхождений элемента <perem>

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

Допустимо использовать только len() для работы с остальными функциями необходимо переводить кортеж в список (По найденной информации в интернете

13. Как создать кортеж с помощью спискового включения.

```
tpl = tuple([i * 2 for i in range(50)])
```