

МИНИСТЕРСТВО НАУКИ И 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

Харченко Екатерина Владимировна

Проверил:

К.т.н., доцент кафедры инфокоммуникаций

Воронкин Роман Александрович

Работа защищена с оценкой: _____

Ставрополь, 2021 г.

РАБОТА С КОРТЕЖАМИ В ЯЗЫКЕ PYTHON

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

1. Создание общедоступного репозиторий на GitHub, в котором используется лицензия MIT и язык программирования Python.

2. Выполняем клонирование созданного репозитория.

```
C:\Users\Катя> git clone https://github.com/ekaterina533/lab2.3.git
Cloning into 'lab2.3'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

C:\Users\Катя>cd lab2.3

C:\Users\Катя\lab2.3>git checkout develop
error: pathspec 'develop' did not match any file(s) known to git

C:\Users\Катя\lab2.3>git checkout -b develop
Switched to a new branch 'develop'

C:\Users\Катя\lab2.3>_
```

3. Проработка примера из лабораторной работы.

```
-1 2 4 8 3 4 9 4 5 2
18
```

Рис. 1.1 – Результат работы программы.

4. Решим индивидуальные задания из 5 лабораторной работы с помощью кортежей без использования списков.

– 1 задание из лабораторной работы №5

```
Введите список10 55 87 45 66 22 88 11 2 5
Количество элементов: 5
Разность элементов: -66
```

Рис. 1.2 – Результат работы программы.

– 2 задание из лабораторной работы №5.

```
Введите элементы-8 2 0 4 60 8 0
Количество элементов равных 0: 2
Сумма элементов, расположенных после минимального элемента: 74
Элементы расположенные по возрастанию : (0, 0, 2, 4, -8, 8, 60)
```

Рис. 1.3 – Результат работы программы.

10. UML-диаграммы деятельности решения индивидуальных заданий:

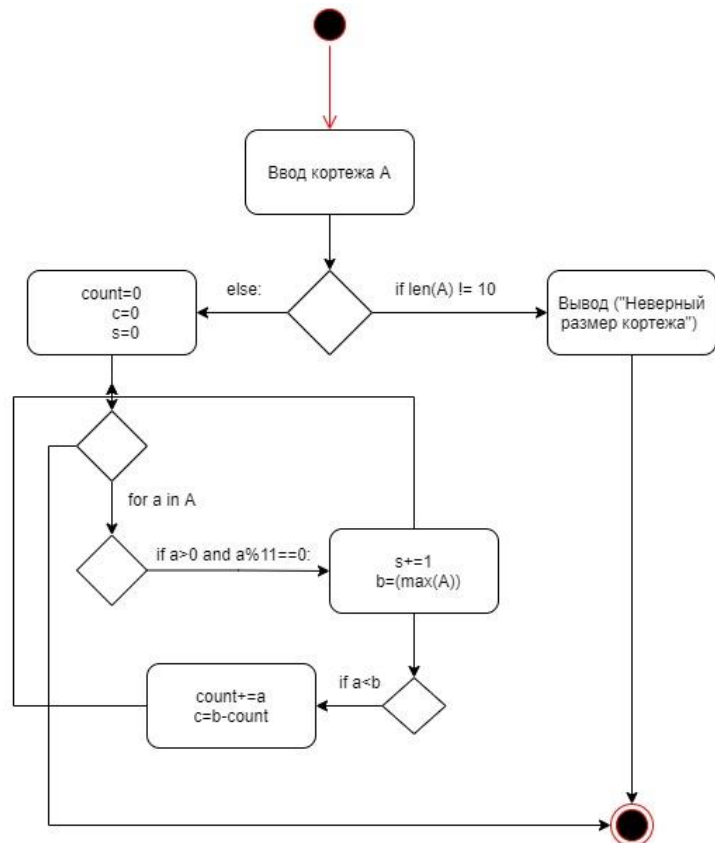


Рис. 1.4 - UML-диаграммы деятельности для 1 задания.

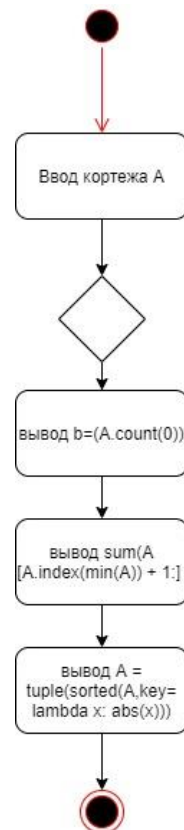


Рис. 1.5 - UML-диаграммы деятельности для 2 задания.

11. Зафиксируем сделанные изменения в репозитории.

```
C:\Users\Катя\lab2.3>git add .

C:\Users\Катя\lab2.3>git commit -m "создание 1 и 2 задания"
[develop de8a6fd] создание 1 и 2 задания
2 files changed, 38 insertions(+)
create mode 100644 "\320\267\320\260\320\264\320\260\320\275\320\270\320\2651.py"
create mode 100644 "\320\267\320\260\320\264\320\260\320\275\320\270\320\2652.py"

C:\Users\Катя\lab2.3>git add .

C:\Users\Катя\lab2.3>git commit -m "создание примера"
[develop db470b4] создание примера
1 file changed, 17 insertions(+)
create mode 100644 "\320\277\321\200\320\270\320\274\320\265\321\200.py"
```

12. Выполняем слияние ветки для разработки с веткой master.

```
C:\Users\Катя\lab2.3>git merge develop
Updating 42d4c74..db470b4
Fast-forward
...60\320\264\320\260\320\275\320\270\320\2651.py" | 23 ++++++
...60\320\264\320\260\320\275\320\270\320\2652.py" | 15 ++++++
...277\321\200\320\270\320\274\320\265\321\200.py" | 17 ++++++
3 files changed, 55 insertions(+)
create mode 100644 "\320\267\320\260\320\264\320\260\320\275\320\270\320\2651.py"
create mode 100644 "\320\267\320\260\320\264\320\260\320\275\320\270\320\2652.py"
create mode 100644 "\320\277\321\200\320\270\320\274\320\265\321\200.py"
```

13. Отправляем сделанные изменения на сервер GitHub.

```
C:\Users\Катя\lab2.3>git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 1.76 KiB | 900.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/ekaterina533/lab2.3.git
42d4c74..db470b4 main -> main
```

Контрольные вопросы:

1. Что такое списки в языке Python?

Список – это изменяемый тип данных

2. Каково назначение кортежей в языке Python?

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

стоит использовать кортежи вместо списков. Одна из них – это обезопасить данные от случайного изменения. 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).

3. Как осуществляется создание кортежей?

Для создания пустого кортежа можно воспользоваться одной из следующих команд:

```
b = tuple(), a = (1, 2, 3, 4, 5)
```

4. Как осуществляется доступ к элементам кортежа?

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

```
a = (1, 2, 3, 4, 5)
print(a[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. 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 в множественном присваивании?

Используя множественное присваивание, можно совершать обмен значениями между двумя переменными:

```
a = 100
b = 'foo'
```

$(a, b) = (b, a)$

`a # 'foo'`

`b # 100`

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

$T3 = T1 + T2$

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 операция `in`

11. Какие методы работы с кортежами Вам известны?

Чтобы получить индекс (позицию) элемента в кортеже, нужно использовать метод `index()`. Общая форма вызова метода следующая:

`pos = T.index(item)`

Чтобы определить количество вхождений заданного элемента в кортеж используется метод `count`, общая форма которого следующая:

`k = T.count(item)`

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.

На базе кортежа можно создать список, верно и обратное утверждение. Для превращения списка в кортеж достаточно передать его в качестве аргумента функции `tuple()`.

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