

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

**Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций  
«Исследование возможностей Git для работы с  
локальными репозиториями»**

**Отчет по лабораторной работе № 2.5  
по дисциплине «Основы программной инженерии»**

Выполнил студент группы  
ПИЖ-б-о-21-1

Трушева В. О. .« » 2022г.

Подпись студента \_\_\_\_\_

Работа защищена «  
» \_\_\_\_\_ 20\_\_ г.

Проверила Воронкин Р.А.

\_\_\_\_\_

(подпись)

Ставрополь 2022

## Методика и порядок выполнения работы

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.
3. Выполните клонирование созданного репозитория.
4. Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm.
5. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.
6. Создайте проект PyCharm в папке репозитория.
7. Проработайте примеры лабораторной работы. Создайте для каждого примера отдельный модуль языка Python. Зафиксируйте изменения в репозитории.

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

```
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
20     print(s)
21
if __name__ == '__main__':
```

1\_Task x

D:\fgit\8\_LR\PyCharm\Scripts\python.exe D:\fgit\8\_LR\Task\1\_Task.

4 5 8 9 5 7 8 4 7 8

8

Process finished with exit code 0

Рисунок 1 – Пример 1

```
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
10    # Проверить количество элементов кортежа.
11    if len(A) != 10:
12        print("неверный размер кортежа", 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
```

if \_\_name\_\_ == '\_\_main\_\_':

1\_Task\_LC x

D:\fgit\8\_LR\PyCharm\Scripts\python.exe D:\fgit\8\_LR\Task\1\_Task

1 2 3 4 5 6 7 8 9 10

10

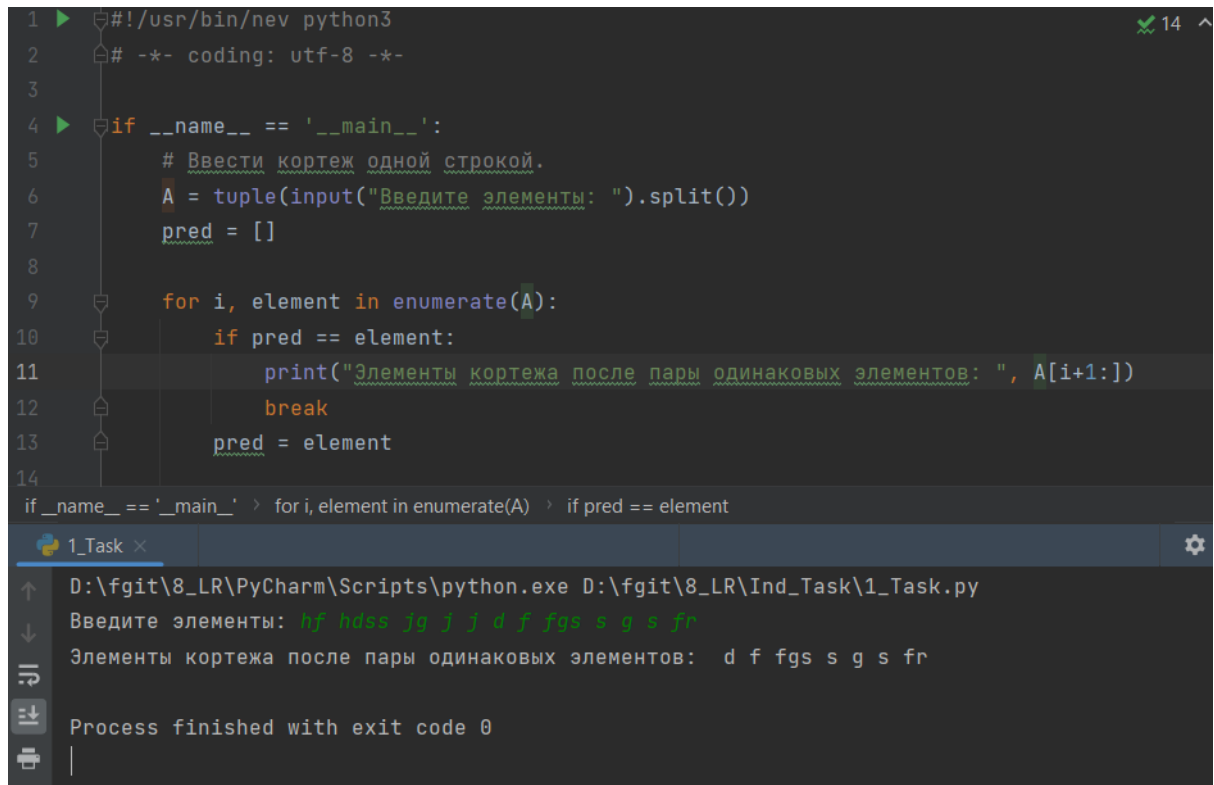
Process finished with exit code 0

Рисунок 2 – Пример 1 (Списковое включение)

8. Приведите в отчете скриншоты результатов выполнения каждой из программ примеров при различных исходных данных вводимых с клавиатуры.

9. Приведите в отчете скриншоты работы программ решения индивидуального задания.

Вариант – 5. Условие. Если в кортеже есть хотя бы одна пара одинаковых соседних элементов, то напечатать все элементы, следующие за элементами первой из таких пар.



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == '__main__':
5     # Ввести кортеж одной строкой.
6     A = tuple(input("Введите элементы: ").split())
7     pred = []
8
9     for i, element in enumerate(A):
10         if pred == element:
11             print("Элементы кортежа после пары одинаковых элементов: ", A[i+1:])
12             break
13         pred = element
```

1 Task x

D:\fgit\8\_LR\PyCharm\Scripts\python.exe D:\fgit\8\_LR\Ind\_Task\1\_Task.py

Введите элементы: hf hds jg j j d f fgs s g s fr

Элементы кортежа после пары одинаковых элементов: d f fgs s g s fr

Process finished with exit code 0

Рисунок 3 – Индивидуальное задание 1

10. Зафиксируйте сделанные изменения в репозитории.
11. Добавьте отчет по лабораторной работе в формате PDF в папку doc репозитория. Зафиксируйте изменения.
12. Выполните слияние ветки для разработки с веткой main / master.
13. Отправьте сделанные изменения на сервер GitHub.
14. Отправьте адрес репозитория GitHub на электронный адрес преподавателя.

#### Вопросы для защиты работы

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. Кортежи в памяти занимают меньший объем по сравнению со списками. Кортежи работают быстрее, чем списки.

## 3. Как осуществляется создание кортежей?

```
a = ()
```

```
b = tuple()
```

## 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.

## 6. Какую роль играют кортежи в множественном присваивании?

Используя множественное присваивание, можно проверить интересный трюк: 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  $T2 = T1[i:j]$  здесь

- ☐  $T2$  – 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?

Для кортежей можно выполнять операцию конкатенации, которая обозначается символом  $+$ .  $T3 = T1 + T2$

9. Как выполняется обход элементов кортежа?

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

10. Как проверить принадлежность элемента кортежу?

Проверка вхождения элемента в кортеж - оператор `in`.

11. Какие методы работы с кортежами Вам известны?

`index()`, `count()`.

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

Доступно.

13. Как создать кортеж с помощью спискового включения.

Так же как и список.