**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра Математической кибернетики и информационных технологий

**Отчет по лабораторной работе №3**

по дисциплине «Теория языков программирования»

Выполнила:

студентка группы БСТ1801

Золотарёва П.Ю

Москва 2019

1. **Цель работы** освоение базовых навыков работы с функциями и файлами в Python.
2. **Задание**: Реализовать задания в виде пользовательских функций. Реализовать единое меню выбора соответствующих функций заданий №1-4 в виде**:** 
   1. Первое задание:

Посчитайте количество файлов в заданной пользователем директории (папке) и вывести на экран.

* 1. Второе задание:

Выведите информацию о товарах в файле products.txt, отсортировав их по цене.

* 1. Третье задание:

Добавить ко второму заданию пользовательский интерфейс по уменьшению количества всех товаров указанных пользователем номеров на введенное пользователем число.

* 1. Четвертое задание:

Добавьте к пользовательскому интерфейсу из задачи №3 возможность сохранения новых данных обратно в файл. Предусмотрите возможность сохранения данных в отсортированном виде в отдельный файл по выбору пользователя.

1. **Ход работы** 
   1. Код программы

import os

print("0 - выйти из программы \n1 - Функция номер 1 \n2 - Функция номер 2\n3 - Функция номер 3\n4 - Функция номер 4")

name\_file = "products.txt";

file = open(name\_file, "r")

sort = []

for line in file:

line = line.strip()

line = line.split(";")

sort.append(line)

max = int(sort[len(sort)-1][0])

access = True

def Work():

global access

what\_to\_continue = input('Хотите ли вы продолжить? Y/N \n')

if what\_to\_continue.lower() == 'n':

access = False

def Find():

path = input('Укажите директорию: ')

count = len([f for f in os.listdir(path) if os.path.isfile(os.path.join(path, f))])

print("Кол-во файлов в директории: " + str(count))

def Sort():

global sort

new\_sort = sort

for j in range(0, len(new\_sort)):

for i in range(0, len(new\_sort)-1):

if int(new\_sort[i][2]) > int(new\_sort[i+1][2]):

field = new\_sort[i]

new\_sort[i] = new\_sort[i+1]

new\_sort[i+1] = field

return new\_sort

def ChangeNumber():

global max

new\_sort = Sort()

id = []

print('Введите id товаров, у которых хотите изменить количество.\n'

'stop - остановка ввода id товаров. \n'

'Максимальный id = ', max)

workID = True

while workID:

numID = input('ID: ')

if numID.isdigit():

numID = int(numID)

if numID <= max:

id.append(numID)

else:

print('Максимальный ID = ', max, ', Вы ввели: ', numID)

elif numID.lower() == 'stop':

workID = False

else:

print('Неверный ID')

count = int(input('Введите кол-во единиц товара, на которое хотите уменьшить..,'

'если значение пользователя будет превышать количество товаров, то количество станет равным 0: '))

for i in range(0, len(new\_sort)):

for j in range(0, len(id)):

if int(new\_sort[i][0]) == int(id[j]):

if int(new\_sort[i][3]) - count >= 0:

new\_sort[i][3] = int(new\_sort[i][3]) - count

else:

new\_sort[i][3] = 0

return new\_sort

def Save():

new\_sort = ChangeNumber()

work = True

while work:

answer = input('Вы желаете сохранить результат как новый файл? Y/N \n')

if answer.lower() == 'y':

file\_name = input('Тогда, введите название файла: ')

new\_file = open(file\_name, 'r+')

for i in range(0, len(new\_sort)):

for j in range(0, len(new\_sort[i])):

if 0 <= j <= 2:

new\_file.write(str(new\_sort[i][j]) + ";")

else:

new\_file.write(str(new\_sort[i][j]))

new\_file.write('\n')

if answer.lower() == 'n':

new\_file = open('products.txt', 'r+')

for i in range(0, len(new\_sort)):

for j in range(0, len(new\_sort[i])):

if 0 <= j <= 2:

new\_file.write(str(new\_sort[i][j]) + ";")

else:

new\_file.write(str(new\_sort[i][j]))

new\_file.write('\n')

work = False

new\_file.close()

else:

work = False

while access:

cmd = input('Введите команду: ')

if cmd == "0":

check\_exit = input('Закрыть программу? Y/N \n')

if check\_exit.lower() == 'y' or '1':

access = False

elif check\_exit.lower() != 'n' or '0':

print('Неверная команда')

if cmd == "1":

Find()

if cmd == "2":

print(Sort())

if cmd == "3":

print(ChangeNumber())

if cmd == "4":

Save()

if not cmd.isdigit():

print('Неизвестная команда')

if not 'check\_exit' in globals() or not 'check\_exit' in locals():

Work()

* 1. Результат работы программы

Скриншот выполнения первой подпрограммы представлен на рисунке 1

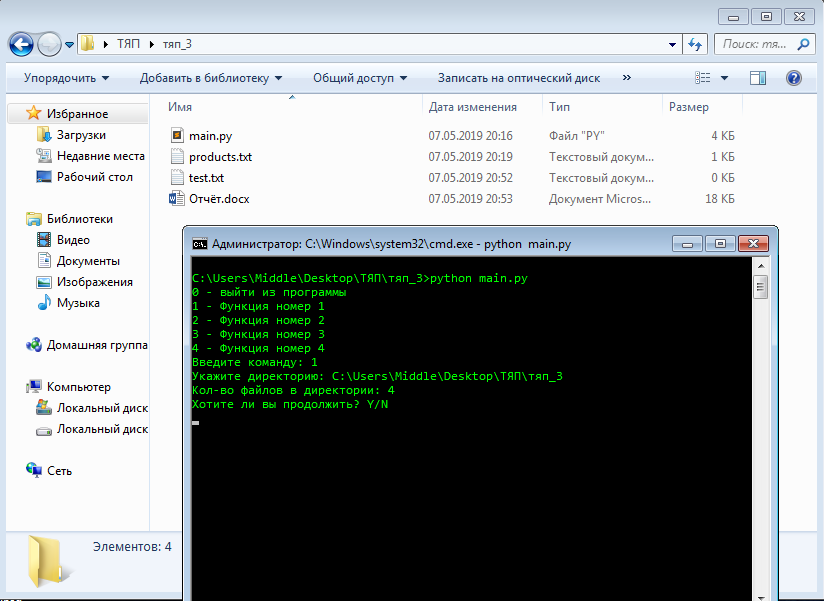


Рисунок 1 - выполнение первой подпрограммы

Скриншот выполнения второй подпрограммы представлен на рисунке 2

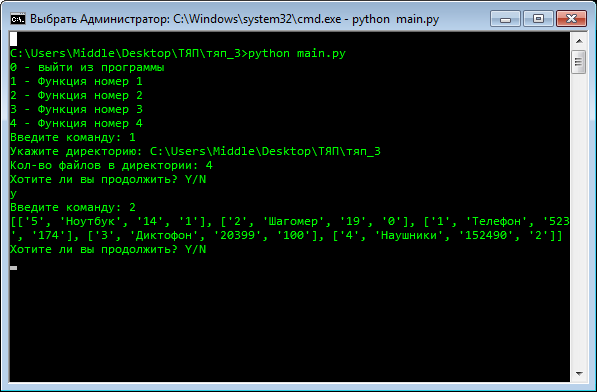


Рисунок 2 - скриншот выполнения второй подпрограммы

Скриншот выполнения третьей подпрограммы представлен на рисунке 3.

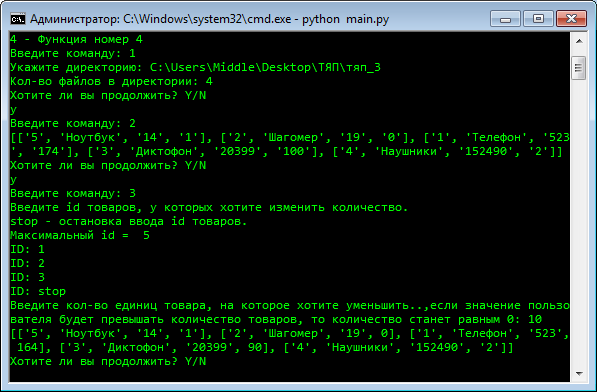


Рисунок 3 - результат выполнения третьей подпрограммы

Скриншот выполнения четвёртой подпрограммы представлен на рисунке 4.

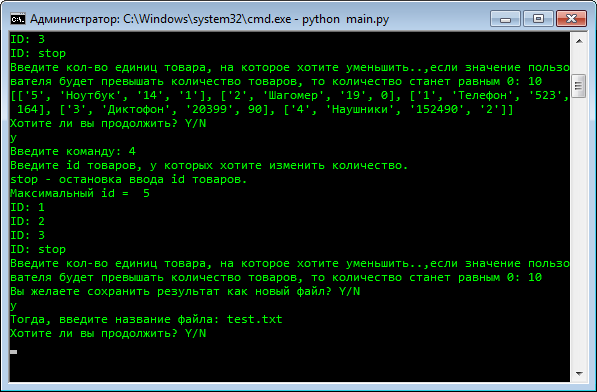


Рисунок 4 - выполнение 4 подпрограммы

Содержимое нового файла test.txt представлено на рисунке 5

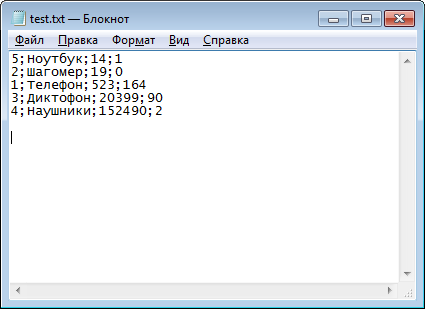


Рисунок 5 - содержимое нового файла test.txt

1. Выполнив данную лабораторную работу, я получила опыт работы с чтением и записью данных в файл на ЯП Python, а также созданию многофункционального консольного приложения, для решения конктретно поставленных задач с помощью возможностей ЯП Python.