**Федеральное государственное образовательное**

**бюджетное учреждение высшего образования**

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ**

**ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**ИНСТИТУТ ЗАОЧНОГО И ОТКРЫТОГО ОБРАЗОВАНИЯ**

**ДЕПАРТАМЕНТ АНАЛИЗА ДАННЫХ,**

**ПРИНЯТИЯ РЕШЕНИЙ И ФИНАНСОВЫХ ТЕХНОЛОГИЙ**

**ПРАКТИЧЕСКАЯ РАБОТА № 6**

**по дисциплине «Алгоритмы и структуры данных в языке Python»**

**Тема: коллекции и файлы**

Направление подготовки - 09.03.03 «Прикладная информатика»

Студент(ка) **Белоусов А.А.**

(Ф.И.О.)

Курс1 № группы **ЗБ-ПИ1-1**

Преподаватель **Петросов Д.А.**

(Ф.И.О.)

Москва 2020

Задания

Коллекции:

**Задание №1**

Создайте коллекцию на свободную тему и выполните следующие ниже задания.

**Задание №2**

Подсчет количества членов коллекции с помощью функции len().

**Задание №3**

Проверку принадлежности элемента данной коллекции c помощью оператора in.

**Задание №4**

Выполните поиск подстроки.

**Задание №5**

Обход коллекции с применением оператора цикла.

**Задание №6**

Найдите максимальный, минимальный элементы коллекции и сумму элементов.

**Задание №7**

Найдите количество определённого пользователем элемента коллекции.

**Задание №8**

Выполните конвертацию типа созданной вами коллекции.

**Задание №9**

Выполните сортировку элементов коллекции.

Файлы:

**Задание №1**

Реализовать решение квадратного уравнения через дискриминант.

**Задание №2**

Реализовать создание, запись, чтение и удаление файла с данными о пользователе. Пользователь выбирает действие самостоятельно, а так же указывает путь к размещению файла.

**Задание №3.1**

Применить функции при выполнении практических работ (не менее 5 штук)

Код программы

Файл Коллекции.py

**def** main():  
 print(**"Задание №1"**)  
 PC\_energy = [(**"видекарта"**, 75), (**"процессор"**, 58), (**"материнская плата"**, 30), (**"DDR4 - 8Гб"**, 20),  
 (**"DDR4 - 8Гб"**, 20), (**"SSD - 1Тб"**, 10), (**"монитор"**, 15)]  
 print(PC\_energy)  
 print()  
  
 print(**"Задания №№2-4"**)  
 print(**f"Колличество элементов компьютера: {**len(PC\_energy)**}"**)  
 print(**f"Монитор входит в борку? {'монитор' in** [item[0] **for** item **in** PC\_energy]**}"**)  
 print(**f"SSD входит в борку? {**any([**'SSD' in** item[0] **for** item **in** PC\_energy])**}"**)  
 print()  
  
 print(**"Задание №5"**)  
 print(**"Элементы компьютера: "**)  
 **for** item **in** PC\_energy:  
 print(item[0])  
 print()  
  
 print(**"Задание №6"**)  
 print(**f"Элемент с наибольшим электропотреблением: {**max(PC\_energy, key=**lambda** item: item[1])[0]**}"**)  
 print(**f"Элемент с наименьшим электропотреблением: {**min(PC\_energy, key=**lambda** item: item[1])[0]**}"**)  
 print(**f"Общее энергопотребление компонентов компьютера: {**sum([item[1] **for** item **in** PC\_energy])**}"**)  
 print()  
  
 print(**"Задание №7"**)  
 input\_item = input(**"Введите название элемента: "**)  
 print(**f"Колличество данных элементов в компьютере: {**[input\_item **in** item[0] **for** item **in** PC\_energy].count(**True**)**}"**)  
 print()  
  
 print(**"Задание №8"**)  
 dict\_PC = {}  
 **for** item **in** PC\_energy:  
 **if** item[0] **in** dict\_PC:  
 dict\_PC[item[0]] += item[1]  
 **else**:  
 dict\_PC[item[0]] = item[1]  
 print(dict\_PC.items())  
 print()  
  
 print(**"Задание №9"**)  
 PC\_energy.sort(reverse=**True**, key=**lambda** item: item[1])  
 print(**"Элементы компьютера: "**)  
 **for** item **in** PC\_energy:  
 print(**f"{**item[0]**}: {**item[1]**}"**)  
 print()  
  
  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 main()

Файл user\_data.py

**import** shelve  
**import** os  
  
  
**def** create\_file(FILENAME):  
 fileexists = **False  
 try**:  
 **with** shelve.open(FILENAME, **"n"**) **as** data\_file:  
 fileexists = **True** print(**"Файл создан заново"**)  
 **except** IOError:  
 print(**"Ошибка при открытии файла"**)  
 choose\_next(FILENAME, fileexists)  
  
  
**def** delete\_file(FILENAME):  
 **if** os.path.isfile(FILENAME + **".dat"**):  
 os.remove(FILENAME + **".dat"**)  
 **if** os.path.isfile(FILENAME + **".bak"**):  
 os.remove(FILENAME + **".bak"**)  
 **if** os.path.isfile(FILENAME + **".dir"**):  
 os.remove(FILENAME + **".dir"**)  
 print(**"Файл удалён"**)  
 choose\_next(FILENAME, **False**)  
  
  
**def** append\_data(FILENAME):  
 **try**:  
 **with** shelve.open(FILENAME) **as** data\_file:  
 print(**"Удаление данных, для завершения введите stop в качестве названия поля"**)  
 **while** (**True**):  
 field\_name = input(**"Введите название поля: "**)  
 **if** field\_name == **"stop"**:  
 **break** field\_value = input(**"Введите значение поля: "**)  
 **try**:  
 data\_file[field\_name] = field\_value  
 **except** IOError:  
 print(**"Ошибка при записи данных в файл"**)  
 **except** IOError:  
 print(**"Ошибка при открытии файла"**)  
 choose\_next(FILENAME, **True**)  
  
  
**def** delete\_data(FILENAME):  
 **try**:  
 **with** shelve.open(FILENAME) **as** data\_file:  
 print(**"Ввод или изменение данных, для завершения введите stop в качестве названия поля"**)  
 **while** (**True**):  
 **try**:  
 **if not** data\_file:  
 print(**"Файл пуст"**)  
 **break** field\_name = input(**"Введите название поля: "**)  
 **if** field\_name == **"stop"**:  
 **break  
  
 if** field\_name **in** data\_file:  
 **del** data\_file[field\_name]  
 print(**"Поле "** + field\_name + **" удалено"**)  
 **else**:  
 print(**"Поле "** + field\_name + **" отсутствует в файле"**)  
 **except** IOError:  
 print(**"Ошибка при удалении данных в файл"**)  
 **except** IOError:  
 print(**"Ошибка при открытии файла"**)  
 choose\_next(FILENAME, **True**)  
  
  
**def** read\_file(FILENAME):  
 **try**:  
 **with** shelve.open(FILENAME) **as** data\_file:  
 **try**:  
 **if not** data\_file:  
 print(**"Файл пуст"**)  
 **for** field **in** data\_file.items():  
 print(field[0] + **" : "** + field[1])  
 **except** IOError:  
 print(**"Ошибка при чтении данных из файла"**)  
 **except** IOError:  
 print(**"Ошибка при открытии файла"**)  
 choose\_next(FILENAME, **True**)  
  
  
**def** input\_comand(can\_use):  
 **while** (**True**):  
 action = input()  
 **if** action **in** can\_use:  
 **return** action  
 **else**:  
 print(**"Введено недопустимое значение. Повторите ввод"**)  
  
  
**def** choose\_next(FILENAME, fileexist):  
 actions = {  
 **"1"**: create\_file,  
 **"2"**: delete\_file,  
 **"3"**: append\_data,  
 **"4"**: delete\_data,  
 **"5"**: read\_file,  
 **"6"**: get\_filename,  
 **"7"**: exit\_programm  
 }  
  
 print(**"Выберите следующие действие"**)  
 **if** fileexist:  
 print(  
 **"1 - очистить файл, 2 - удалить файл, 3 - добавить или изменить данные, 4 - удалить данные, 5 - прочитать данные, 6 - указать новый путь к файлу, 7 - выйти из программы"**)  
 actions[input\_comand(list(actions.keys()))](FILENAME)  
 **else**:  
 print(**"1 - создать файл, 6 - указать новый путь к файлу, 7 - выйти из программы"**)  
 actions[input\_comand([**"1"**, **"6"**, **"7"**])](FILENAME)  
  
  
**def** get\_filename(\*args):  
 FILENAME = input(**"Введите путь к файлу (без расширения):\n"**)  
 fileexist = **False  
 if** os.path.isfile(FILENAME + **".dat"**):  
 print(**"Файл найден"**)  
 fileexist = **True  
 else**:  
 print(**"Файл не найден"**)  
 choose\_next(FILENAME, fileexist)  
  
  
**def** exit\_programm(\*args):  
 exit(0)  
  
  
**def** main():  
 get\_filename()  
  
  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 main()

Результат выполнения программы

Файл Коллекции.py

**Задание №1**

[('видекарта', 75), ('процессор', 58), ('материнская плата', 30), ('DDR4 - 8Гб', 20), ('DDR4 - 8Гб', 20), ('SSD - 1Тб', 10), ('монитор', 15)]

**Задания №№2-4**

Колличество элементов компьютера: 7

Монитор входит в борку? True

SSD входит в борку? True

**Задание №5**

Элементы компьютера:

видекарта

процессор

материнская плата

DDR4 - 8Гб

DDR4 - 8Гб

SSD - 1Тб

монитор

**Задание №6**

Элемент с наибольшим электропотреблением: видекарта

Элемент с наименьшим электропотреблением: SSD - 1Тб

Общее энергопотребление компонентов компьютера: 228

**Задание №7**

Введите название элемента: DDR

Колличество данных элементов в компьютере: 2

**Задание №8**

dict\_items([('видекарта', 75), ('процессор', 58), ('материнская плата', 30), ('DDR4 - 8Гб', 40), ('SSD - 1Тб', 10), ('монитор', 15)])

**Задание №9**

Элементы компьютера:

видекарта: 75

процессор: 58

материнская плата: 30

DDR4 - 8Гб: 20

DDR4 - 8Гб: 20

монитор: 15

SSD - 1Тб: 10

Файл user\_data.py

**Задание №1**

Было выполнено в практической работе №3

Код функции:

*# задание 4***def** solveSqrEqual(a=0, b=0, c=0):  
 d = b \*\* 2 - 4 \* a \* c  
 **if** d > 0:  
 **return** (-b + math.sqrt(d)) / 2 \* a, (-b - math.sqrt(d)) / 2 \* a  
 **if** d == 0:  
 **return** -b / 2 \* a

**Задание №2**

**Введите путь к файлу (без расширения):**

данные нового пользователя

Файл не найден

Выберите следующие действие

1 - создать файл, 6 - указать новый путь к файлу, 7 - выйти из программы

1

Файл создан заново

Выберите следующие действие

1 - очистить файл, 2 - удалить файл, 3 - добавить или изменить данные, 4 - удалить данные, 5 - прочитать данные, 6 - указать новый путь к файлу, 7 - выйти из программы

5

Файл пуст

Выберите следующие действие

1 - очистить файл, 2 - удалить файл, 3 - добавить или изменить данные, 4 - удалить данные, 5 - прочитать данные, 6 - указать новый путь к файлу, 7 - выйти из программы

4

Ввод или изменение данных, для завершения введите stop в качестве названия поля

Файл пуст

Выберите следующие действие

1 - очистить файл, 2 - удалить файл, 3 - добавить или изменить данные, 4 - удалить данные, 5 - прочитать данные, 6 - указать новый путь к файлу, 7 - выйти из программы

3

Удаление данных, для завершения введите stop в качестве названия поля

Введите название поля: Имя

Введите значение поля: Пётр

Введите название поля: Фамилия

Введите значение поля: Питонов

Введите название поля: Возраст

Введите значение поля: 19

Введите название поля: e-mail

Введите значение поля: myemail@gmail.ru

Введите название поля: stop

Выберите следующие действие

1 - очистить файл, 2 - удалить файл, 3 - добавить или изменить данные, 4 - удалить данные, 5 - прочитать данные, 6 - указать новый путь к файлу, 7 - выйти из программы

5

Имя : Пётр

Фамилия : Питонов

Возраст : 19

e-mail : myemail@gmail.ru

Выберите следующие действие

1 - очистить файл, 2 - удалить файл, 3 - добавить или изменить данные, 4 - удалить данные, 5 - прочитать данные, 6 - указать новый путь к файлу, 7 - выйти из программы

4

Ввод или изменение данных, для завершения введите stop в качестве названия поля

Введите название поля: почта

Поле почта отсутствует в файле

Введите название поля: e-mail

Поле e-mail удалено

Введите название поля: фамилия

Поле фамилия отсутствует в файле

Введите название поля: Фамилия

Поле Фамилия удалено

Введите название поля: stop

Выберите следующие действие

1 - очистить файл, 2 - удалить файл, 3 - добавить или изменить данные, 4 - удалить данные, 5 - прочитать данные, 6 - указать новый путь к файлу, 7 - выйти из программы

5

Имя : Пётр

Возраст : 19

Выберите следующие действие

1 - очистить файл, 2 - удалить файл, 3 - добавить или изменить данные, 4 - удалить данные, 5 - прочитать данные, 6 - указать новый путь к файлу, 7 - выйти из программы

1

Файл создан заново

Выберите следующие действие

1 - очистить файл, 2 - удалить файл, 3 - добавить или изменить данные, 4 - удалить данные, 5 - прочитать данные, 6 - указать новый путь к файлу, 7 - выйти из программы

5

Файл пуст

Выберите следующие действие

1 - очистить файл, 2 - удалить файл, 3 - добавить или изменить данные, 4 - удалить данные, 5 - прочитать данные, 6 - указать новый путь к файлу, 7 - выйти из программы

2

Файл удалён

Выберите следующие действие

1 - создать файл, 6 - указать новый путь к файлу, 7 - выйти из программы

6

Введите путь к файлу (без расширения):

../данные пользователя

Файл найден

Выберите следующие действие

1 - очистить файл, 2 - удалить файл, 3 - добавить или изменить данные, 4 - удалить данные, 5 - прочитать данные, 6 - указать новый путь к файлу, 7 - выйти из программы

5

Имя : Павел

возраст : 20

e-mail : email@mail.ru

город : Москва

Выберите следующие действие

1 - очистить файл, 2 - удалить файл, 3 - добавить или изменить данные, 4 - удалить данные, 5 - прочитать данные, 6 - указать новый путь к файлу, 7 - выйти из программы

7

**Задание №3.1**

Было выполнено в текущей и предыдущих практических работах.

Вывод

В ходе выполнения практической работы были получены навыки работы с коллекциями в языке Python, а также опыт работы с файлами.