



Министерство науки и высшего образования Российской Федерации  
Калужский филиал  
федерального государственного бюджетного  
образовательного учреждения высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

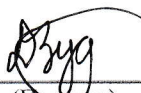
КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

## ДОМАШНЯЯ РАБОТА №2

### «Работа с файлами»

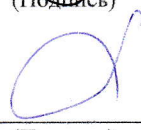
ДИСЦИПЛИНА: «Перспективные языки программирования»

Выполнил: студент гр. ИУК4-32Б

  
(Подпись)

(Зудин Д.В.)  
(Ф.И.О.)

Проверил:

  
(Подпись)

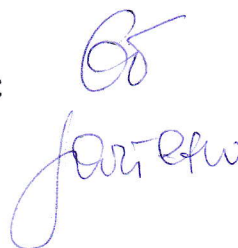
(Пчелинцева Н.И.)  
(Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:



Калуга, 2022 г.

**Цель:** приобретение практических навыков работы с файлами и файловой системой на языке программирования Python.

**Задачи:**

1. Ознакомиться со способами работы с файлами и файловой системой в Python;
2. Изучить способы работы с файлами формата csv, json, xml;
3. Закрепить полученные в ходе выполнения домашней работы навыки.

## **Вариант №20**

### **Формулировка задания №1**

Выберите любую папку на своём компьютере, имеющую вложенные директории.

Выведите на печать в терминал и сохраните в файл txt названия всех файлов, которые содержат введенное с консоли слово. Заархивируйте данную папку средствами python.

### **Листинг программы для задания №1**

```
import os
import shutil

print("Поиск в папке Test файлов, содержащих в названии введенное слово")
string = input("Введите слово: ")
path = 'C:/Users/user/OneDrive/Рабочий стол/3-ий семестр/Перспективные языки программирования/Test'
res = []
for current_dir, dirs, files in os.walk(path):
    for i in files:
        if i.find(string) != -1:
            res.append(i)

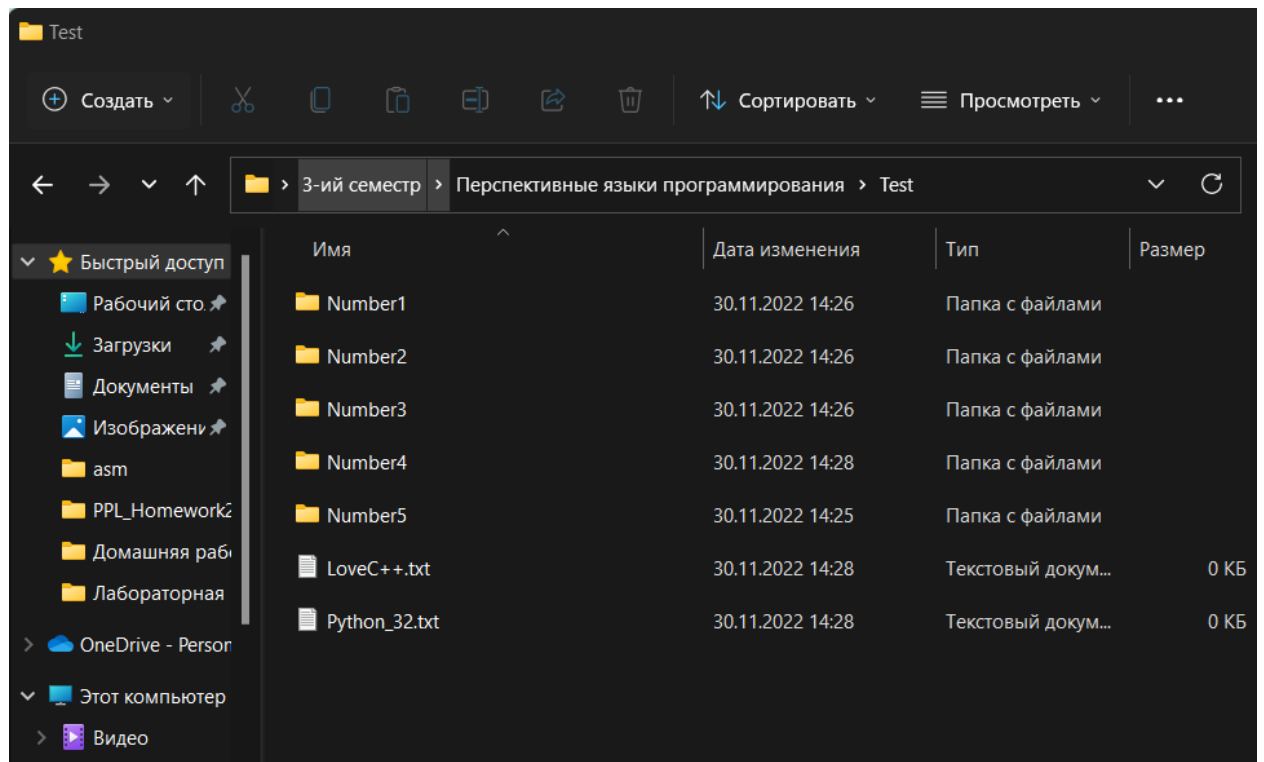
with open("Task_1.txt", "w") as f:
    f.write('\n'.join(res))

print("Искомые файлы:")
print('\n'.join(res))

shutil.make_archive("Test", 'zip', path)
```

### **Результат выполнения программы для задания №1**

1. Исходная папка до выполнения программы



Некоторые папки также содержат файлы, но есть и пустые.

## 2. Вывод в консоль

Поиск в папке Test файлов, содержащих в названии введенное слово

Введите слово: Python

Искомые файлы:

Python\_32.txt

Python\_1.txt

Python\_2.txt

Python.txt

Python\_1.txt

## 3. Текстовый файл «Task\_1.txt»

Python\_32.txt

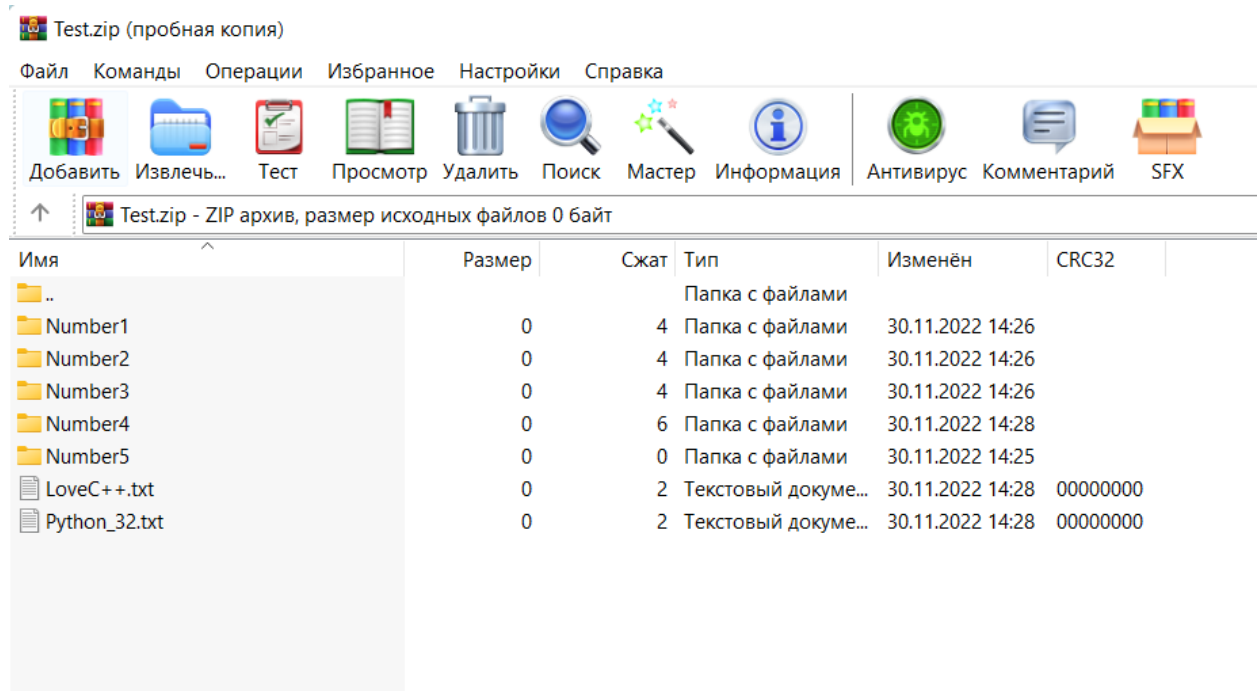
Python\_1.txt

Python\_2.txt

Python.txt

Python\_1.txt

#### 4. Исходная папка после выполнения программы



#### Формулировка задания №2

Создать файл в формате csv, содержащий сведения о покупателях в интернет-магазине.

Структура: id, ФИО, дата рождения, общая сумма покупок, количество заказов.

Предусмотреть возможность корректировки файла по вводимому коду корректировки, например:

- 1 – удалить запись (по id);
- 2 – добавить новую запись;
- 3 – изменить запись (по id);
- 4 – получить информацию (по id);
- 5 – отсортировать покупателей по:
  - ФИО
  - Дате рождения
  - Общей сумме покупок
  - Количеству заказов
- 6 – сохранить в отдельный файл и вывести на консоль покупателя с минимальной суммой покупок и покупателя с минимальным количеством заказов.

#### Листинг файла shoppers.py

```
class Shopper:
    def __init__(self):
```

```

        """Инициализация покупателя"""
        self.__shopper_id = 0                # ID
        self.__name = ''                     # ФИО
        self.__date_of_birthday = ''         # Дата рождения
        self.__total = 0                     # Общая сумма покупок
        self.__orders = 0                    # Количество заказов

    def __str__(self):
        """Форматирование вывода информации о покупателе"""
        return f"ID: {self.__shopper_id}\n" \
               f"ФИО: {self.__name}\n" \
               f"Дата рождения: {self.__date_of_birthday}\n" \
               f"Общая сумма покупок: {self.__total} р.\n" \
               f"Количество заказов: {self.__orders} шт."

    @property
    def shopper_id(self):
        return self.__shopper_id

    @shopper_id.setter
    def shopper_id(self, shopper_id: int):
        if shopper_id > 0:
            self.__shopper_id = shopper_id
        else:
            raise ValueError("Incorrect input!")

    @property
    def name(self):
        return self.__name

    @name.setter
    def name(self, name: str):
        self.__name = name

    @property
    def date_of_birthday(self):
        return self.__date_of_birthday

    @date_of_birthday.setter
    def date_of_birthday(self, date_of_birthday: str):
        valid = True
        if len(date_of_birthday) != 10 and date_of_birthday[2] != '.' and

```

```

date_of_birthday[5] != '.':
    valid = False
    if not 0 < int(date_of_birthday[:2]) < 32:
        valid = False
    if not 0 < int(date_of_birthday[3:5]) < 13:
        valid = False
    if not 1900 < int(date_of_birthday[6:10]) < 2012:
        valid = False

    if valid:
        self.__date_of_birthday = date_of_birthday
    else:
        raise ValueError("Incorrect input!")

@property
def total(self):
    return self.__total

@total.setter
def total(self, total: float):
    if total > 0:
        self.__total = total
    else:
        raise ValueError("Incorrect input!")

@property
def orders(self):
    return self.__orders

@orders.setter
def orders(self, orders: int):
    if orders > 0:
        self.__orders = orders
    else:
        raise ValueError("Incorrect input!")

```

## Листинг программы для задания №2

```

import os
import csv
from enum import Enum
from shoppers import Shopper

```

```

class Commands(Enum):
    DELETE_SHOPPER = 1
    ADD_SHOPPER = 2
    CHANGE_SHOPPER = 3
    GET_SHOPPER_INFO = 4
    SORT_SHOPPERS_NAME = 5
    SORT_SHOPPERS_DATE_OF_BIRTHDAY = 6
    SORT_SHOPPERS_TOTAL = 7
    SORT_SHOPPERS_ORDERS = 8
    SAVE_IN_CSV_FILE = 9
    LOAD_FROM_CSV_FILE = 10
    PRINT_SHOPPERS = 11
    INDIVIDUAL_TASK = 12
    EXIT = 0

def input_int(message: str, start: int, end: int) -> int:
    valid = False
    n = 0
    while not valid:
        try:
            n = int(input(message))
        except ValueError:
            print('Вы ввели не число. Попробуйте снова.')
        else:
            if start <= n <= end:
                valid = True
            else:
                print('Введённое число вне диапазона!')
    return n

def input_shopper(data: list[Shopper]) -> Shopper:
    """Добавление записи о покупателе"""
    new_shopper = Shopper()
    new_id, valid_id = -1, False
    while not valid_id:
        new_id = int(input("Введите ID покупателя: "))
        valid_id = new_id not in list(i.shopper_id for i in data)
        if not valid_id:
            print("ID должен быть уникальным!")
    valid = False

```

```

while not valid:
    try:
        new_shopper.shopper_id = new_id
        new_shopper.name = input("Введите ФИО: ")
        new_shopper.date_of_birthday = input("Введите дату рождения: ")
        new_shopper.total = int(input("Введите общую сумму покупок: "))
        new_shopper.orders = int(input("Введите количество заказов: "))
    except ValueError:
        print("Некорректный ввод! Попробуйте снова.")
    except IndexError:
        print("Некорректный ввод! Попробуйте снова.")
    else:
        valid = True
return new_shopper

def m_print_shoppers(data: list[Shopper]):
    """Вывод списка покупателей"""
    if not data:
        print("Список пустой!")
    else:
        print("Покупатели:")
        for i in data:
            print(i)
            print()

def m_add_shopper(data: list[Shopper]):
    """Добавление покупателя"""
    data.append(input_shopper(data))

def m_delete_shopper(data: list[Shopper]):
    """Удаление покупателя"""
    if not data:
        print("Список пустой!")
    else:
        shopper_id = int(input("Введите ID удаляемого покупателя: "))
        index = -1
        for i in range(len(data)):
            if data[i].shopper_id == shopper_id:
                index = i

```



```

        break
    if index != -1:
        data.pop(index)
        print(f"Покупатель с ID {shopper_id} удалён")
    else:
        print("Покупателя с таким ID не существует!")

def m_change_shopper(data: list[Shopper]):
    """Изменение информации о покупателе"""
    if not data:
        print("Список пустой!")
    else:
        shopper_id = int(input("Введите ID изменяемого покупателя "))
        index = -1
        for i in range(len(data)):
            if data[i].shopper_id == shopper_id:
                index = i
                break
        if index != -1:
            data[index] = input_shopper(data)
        else:
            print("Покупателя с таким ID не существует!")

def m_get_shopper_info(data: list[Shopper]):
    """Получить информацию о покупателе"""
    if not data:
        print("Список пустой!")
    else:
        shopper_id = int(input("Введите ID просматриваемого покупателя: "))
        find_shopper = False
        for i in data:
            if i.shopper_id == shopper_id:
                find_shopper = True
                print(i)
        if not find_shopper:
            print("Покупателя с таким ID не существует!")

def m_sort_shoppers_name(data: list[Shopper]):
    """Сортировка по имени"""

```

```

data.sort(key=lambda x: x.name)
print("Покупатели отсортированы по ФИО")

def m_sort_shoppers_date_of_birthday(data: list[Shopper]):
    """Сортировка по дате рождения"""
    data.sort(key=lambda x: x.date_of_birthday)
    print("Покупатели отсортированы по дате рождения")

def m_sort_shoppers_total(data: list[Shopper]):
    """Сортировка по общей сумме покупок"""
    data.sort(key=lambda x: x.total)
    print("Покупатели отсортированы по общей сумме покупок")

def m_sort_shoppers_orders(data: list[Shopper]):
    """Сортировка по количеству заказов"""
    data.sort(key=lambda x: x.orders)
    print("Покупатели отсортированы по количеству заказов")

def m_save_in_csv_file(data: list[Shopper], data_file: str):
    """Сохранение данных в csv-файл"""
    with open(data_file, 'w', encoding='utf8', newline='') as file:
        writer = csv.writer(file)
        for i in data:
            if i:
                writer.writerow([i.shopper_id, i.name, i.date_of_birthday,
i.total, i.orders])
    print("Данные успешно сохранены в файл")

def m_load_from_csv_file(data: list[Shopper], data_file: str):
    """Загрузка данных из csv-файла"""
    with open(data_file, 'r', encoding='utf8') as file:
        reader = csv.reader(file)
        for curr_shopper in reader:
            if curr_shopper:
                shopper = Shopper()
                print(curr_shopper)
                shopper.shopper_id = int(curr_shopper[0])

```

```

        shopper.name = curr_shopper[1]
        shopper.date_of_birthday = curr_shopper[2]
        shopper.total = int(curr_shopper[3])
        shopper.orders = int(curr_shopper[4])
        data.append(shopper)

print("Данные успешно загружены из файла")

def m_individual_task(data: list[Shopper], data_file: str):
    """Индивидуальное задание"""
    m_sort_shoppers_total(data)
    min_total = data[0]
    m_sort_shoppers_orders(data)
    min_orders = data[0]

    with open(data_file, 'w', encoding='utf8', newline='') as file:
        writer = csv.writer(file)
        writer.writerow([min_total.shopper_id, min_total.name,
min_total.date_of_birthday,
                        min_total.total, min_total.orders])
        writer.writerow([min_orders.shopper_id, min_orders.name,
min_orders.date_of_birthday,
                        min_orders.total, min_orders.orders])

    print("Информация о покупателе с минимальной общей суммой заказа
загружена в файл")
    print("Информация о покупателе с минимальным количеством заказов
загружена в файл")

def print_menu():
    """Меню"""
    print(f" 1. Удалить запись по id\n"
          f" 2. Добавить новую запись\n"
          f" 3. Изменить запись по id\n"
          f" 4. Получить информацию по id\n"
          f" 5. Сортировка по ФИО\n"
          f" 6. Сортировка по дате рождения\n"
          f" 7. Сортировка по общей сумме покупок\n"
          f" 8. Сортировка по количеству заказов\n"
          f" 9. Сохранить данные в файл\n"
          f"10. Загрузить данные из файла\n"
          f"11. Вывести данные\n")

```

```
f"12. Индивидуальное задание\n"  
f" 0. Выход")
```

```
def main():  
    """Основная логика программы"""  
    _exit = False  
    data_file = 'data.csv'  
    individual_task_file = 'task.csv'  
    data = []  
    while not _exit:  
        os.system('cls')  
        print_menu()  
        command = input_int("Введите команду: ", 0, 12)  
        if command == Commands.DELETE_SHOPPER.value:  
            m_delete_shopper(data)  
        elif command == Commands.ADD_SHOPPER.value:  
            m_add_shopper(data)  
        elif command == Commands.CHANGE_SHOPPER.value:  
            m_change_shopper(data)  
        elif command == Commands.GET_SHOPPER_INFO.value:  
            m_get_shopper_info(data)  
        elif command == Commands.SORT_SHOPPERS_NAME.value:  
            m_sort_shoppers_name(data)  
        elif command == Commands.SORT_SHOPPERS_DATE_OF_BIRTHDAY.value:  
            m_sort_shoppers_date_of_birthday(data)  
        elif command == Commands.SORT_SHOPPERS_TOTAL.value:  
            m_sort_shoppers_total(data)  
        elif command == Commands.SORT_SHOPPERS_ORDERS.value:  
            m_sort_shoppers_orders(data)  
        elif command == Commands.SAVE_IN_CSV_FILE.value:  
            m_save_in_csv_file(data, data_file)  
        elif command == Commands.LOAD_FROM_CSV_FILE.value:  
            m_load_from_csv_file(data, data_file)  
        elif command == Commands.PRINT_SHOPPERS.value:  
            m_print_shoppers(data)  
        elif command == Commands.INDIVIDUAL_TASK.value:  
            m_individual_task(data, individual_task_file)  
        elif command == Commands.EXIT.value:  
            _exit = True  
    if not _exit:  
        os.system('pause')
```

```
if __name__ == '__main__':  
    main()
```

## Результат выполнения программы для задания №2

### Меню

1. Удалить запись по id
2. Добавить новую запись
3. Изменить запись по id
4. Получить информацию по id
5. Сортировка по ФИО
6. Сортировка по дате рождения
7. Сортировка по общей сумме покупок
8. Сортировка по количеству заказов
9. Сохранить данные в файл
10. Загрузить данные из файла
11. Вывести данные
12. Индивидуальное задание
0. Выход

### Добавление покупателя

Введите команду: 2

Введите ID покупателя: 1

Введите ФИО: Зудин Даниил Васильевич

Введите дату рождения: 25.01.2003

Введите общую сумму покупок: 5000

Введите количество заказов: 5

Для продолжения нажмите любую клавишу . . .

Введите команду: 2

Введите ID покупателя: 2

Введите ФИО: Исаев Станислав Алексеевич

Введите дату рождения: 25.06.2003

Введите общую сумму покупок: 1200

Введите количество заказов: 7

Для продолжения нажмите любую клавишу . . .

Введите команду: 2

Введите ID покупателя: 3

Введите ФИО: Хрущев Никита Сергеевич

Введите дату рождения: 12.01.2003

Введите общую сумму покупок: 7000

Введите количество заказов: 1  
Для продолжения нажмите любую клавишу . . .

Введите команду: 2  
Введите ID покупателя: 4  
Введите ФИО: Иову Дмитрий Олегович  
Введите дату рождения: 23.08.2005  
Введите общую сумму покупок: 12000  
Введите количество заказов: 3  
Для продолжения нажмите любую клавишу . . .

### **Вывод информации о покупателях**

Введите команду: 11  
Покупатели:  
ID: 1  
ФИО: Зудин Даниил Васильевич  
Дата рождения: 25.01.2003  
Общая сумма покупок: 5000 р.  
Количество заказов: 5 шт.

ID: 2  
ФИО: Исаев Станислав Алексеевич  
Дата рождения: 25.06.2003  
Общая сумма покупок: 1200 р.  
Количество заказов: 7 шт.

ID: 3  
ФИО: Хрущев Никита Сергеевич  
Дата рождения: 12.01.2003  
Общая сумма покупок: 7000 р.  
Количество заказов: 1 шт.

ID: 4  
ФИО: Иову Дмитрий Олегович  
Дата рождения: 23.08.2005  
Общая сумма покупок: 12000 р.  
Количество заказов: 3 шт.

### **Удаление покупателя**

Введите команду: 1  
Введите ID удаляемого покупателя: 4  
Покупатель с ID 4 удалён

Для продолжения нажмите любую клавишу . . .

Введите команду: 11

Покупатели:

ID: 1

ФИО: Зудин Даниил Васильевич

Дата рождения: 25.01.2003

Общая сумма покупок: 5000 р.

Количество заказов: 5 шт.

ID: 2

ФИО: Исаев Станислав Алексеевич

Дата рождения: 25.06.2003

Общая сумма покупок: 1200 р.

Количество заказов: 7 шт.

ID: 3

ФИО: Хрущев Никита Сергеевич

Дата рождения: 12.01.2003

Общая сумма покупок: 7000 р.

Количество заказов: 1 шт.

Для продолжения нажмите любую клавишу . . .

### **Получить информацию о покупателе по id**

Введите команду: 4

Введите ID просматриваемого покупателя: 1

ID: 1

ФИО: Зудин Даниил Васильевич

Дата рождения: 25.01.2003

Общая сумма покупок: 5000 р.

Количество заказов: 5 шт.

Для продолжения нажмите любую клавишу . . .

### **Сортировка покупателей по дате рождения**

Введите команду: 6

Покупатели отсортированы по дате рождения

Для продолжения нажмите любую клавишу . . .

ID: 3

ФИО: Хрущев Никита Сергеевич

Дата рождения: 12.01.2003

Общая сумма покупок: 7000 р.

Количество заказов: 1 шт.

ID: 1

ФИО: Зудин Даниил Васильевич

Дата рождения: 25.01.2003

Общая сумма покупок: 5000 р.

Количество заказов: 5 шт.

ID: 2

ФИО: Исаев Станислав Алексеевич

Дата рождения: 25.06.2003

Общая сумма покупок: 1200 р.

Количество заказов: 7 шт.

Для продолжения нажмите любую клавишу . . .

### **Сортировка покупателей по общей сумме покупок**

Введите команду: 7

Покупатели отсортированы по общей сумме покупок

Для продолжения нажмите любую клавишу . . .

Введите команду: 11

Покупатели:

ID: 2

ФИО: Исаев Станислав Алексеевич

Дата рождения: 25.06.2003

Общая сумма покупок: 1200 р.

Количество заказов: 7 шт.

ID: 1

ФИО: Зудин Даниил Васильевич

Дата рождения: 25.01.2003

Общая сумма покупок: 5000 р.

Количество заказов: 5 шт.

ID: 3

ФИО: Хрущев Никита Сергеевич

Дата рождения: 12.01.2003

Общая сумма покупок: 7000 р.

Количество заказов: 1 шт.



Для продолжения нажмите любую клавишу . . .

### **Сортировка покупателей по количеству заказов**

Введите команду: 8

Покупатели отсортированы по количеству заказов

Для продолжения нажмите любую клавишу . . .

Введите команду: 11

Покупатели:

ID: 3

ФИО: Хрущев Никита Сергеевич

Дата рождения: 12.01.2003

Общая сумма покупок: 7000 р.

Количество заказов: 1 шт.

ID: 1

ФИО: Зудин Даниил Васильевич

Дата рождения: 25.01.2003

Общая сумма покупок: 5000 р.

Количество заказов: 5 шт.

ID: 2

ФИО: Исаев Станислав Алексеевич

Дата рождения: 25.06.2003

Общая сумма покупок: 1200 р.

Количество заказов: 7 шт.

Для продолжения нажмите любую клавишу . . .

### **Сортировка покупателей по ФИО**

Введите команду: 5

Покупатели отсортированы по ФИО

Для продолжения нажмите любую клавишу . . .

Введите команду: 11

Покупатели:

ID: 1

ФИО: Зудин Даниил Васильевич

Дата рождения: 25.01.2003

Общая сумма покупок: 5000 р.

Количество заказов: 5 шт.

ID: 2

ФИО: Исаев Станислав Алексеевич

Дата рождения: 25.06.2003

Общая сумма покупок: 1200 р.

Количество заказов: 7 шт.

ID: 3

ФИО: Хрущев Никита Сергеевич

Дата рождения: 12.01.2003

Общая сумма покупок: 7000 р.

Количество заказов: 1 шт.

Для продолжения нажмите любую клавишу . . .

### **Сохранение данных в csv-файл**

Введите команду: 9

Данные успешно сохранены в файл

Для продолжения нажмите любую клавишу . . .

### **Загрузка данных из csv-файла**

Введите команду: 10

['1', 'Зудин Даниил Васильевич', '25.01.2003', '5000', '5']

['2', 'Исаев Станислав Алексеевич', '25.06.2003', '1200', '7']

['3', 'Хрущев Никита Сергеевич', '12.01.2003', '7000', '1']

Данные успешно загружены из файла

Для продолжения нажмите любую клавишу . . .

### **Индивидуальное задание**

Введите команду: 12

Покупатели отсортированы по общей сумме покупок

Покупатели отсортированы по количеству заказов

Информация о покупателе с минимальной общей суммой заказа загружена в файл

Информация о покупателе с минимальным количеством заказов загружена в файл

Для продолжения нажмите любую клавишу . . .

### **Основной csv-файл после выполнения программы**

1,Зудин Даниил Васильевич,25.01.2003,5000,5

2,Исаев Станислав Алексеевич,25.06.2003,1200,7

3,Хрущев Никита Сергеевич,12.01.2003,7000,1

### **Csv-файл индивидуального задания**

2,Исаев Станислав Алексеевич,25.06.2003,1200,7

3,Хрущев Никита Сергеевич,12.01.2003,7000,1

### **Выводы:**

В ходе работы были приобретены практические навыки работы с файлами и файловой системой на языке программирования Python.