**Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики**

**Программирование**

**Домашнее задание**

**Реализация программной модели инфокоммуникационной системы**

Выполнил:

Затикян Сергей Арменович

Факультет ИКТ

Группа K3121

Преподаватель:

Казанова Полина Петровна

**Санкт-Петербург**

**2022**

**Цели работы:**

Создать программное обеспечение системы обработки данных: “Программа для контроля собственных денежных средств” и реализовать в нём следующие функции:

* добавление продукта в коллекцию;
* просмотр всех записей;
* фильтрация покупок по дате или категории;
* сортировка по возрастанию или убыванию цены;
* удаление записей;
* выход из программы.

**Задачи:**

* Провести анализ предметной области и требований.
* Придумать алгоритмы для решения конкретных задач.
* Написать программу для данных алгоритмов на языке Python.
* Получить требуемый результат, убедившись в работоспособности программы.

**Ход работы:**

Перед началом создания алгоритма необходимо провести анализ предметной области и требований. Наиболее похожим программным обеспечением являются приложения электронного денежного дневника. В нем пользователи могут записывать свои покупки и могу контролировать свои расходы. По сравнению с другими приложениями данного программное обеспечение будет иметь больше возможностей, например сортировка по любому из параметров.

Для выполнения работы был написан файл, в который находился первоначальный список покупок (рис. 1).

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 1*

Затем программа считывает данный файл, записывает в переменную количество записей, а в list\_goods список всех покупок (рис. 2).

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 2*

Далее на выбор пользователю даются команды: ‘выйти’, ‘добавить’, ‘удалить’, ‘очистить список’, ‘отсортировать’ и ‘просмотреть все записи’. Команды ‘добавить’, ‘удалить’, ‘отсортировать’ и ‘просмотреть все записи’ вызывают соответствующие им функции – add, delete, sorting или printing. Команда ‘очистить список’ вызывает функцию clear, которая опустошает список покупок. При вводе команды ‘выйти’, цикл завершается. Если введённая команда не совпала ни с одной из предложенных, то на экран выводится уведомление о неверном вводе команды (рис. 3).

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 3*

Первая из них – функция add (рис. 4), позволяющая добавлять элемент в список. В исходный список добавляется ячейка с информацией о товаре: датой покупки, категорией товара, его наименованием, ценой, количеством и стоимостью покупки. Функция написана при помощи конструкции try – except, чтобы в случае ошибки ввода данных, программа просто выводила уведомление об ошибке и возвращалась в основное меню. В случае, если пользователь введёт отрицательную цену и не натуральное число товаров, программа также выведет уведомление об ошибки и вернётся в главное меню.

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 4*

Также в конце функции реализована возможность отмены создания записи о покупке.

Далее создана функция empty (рис. 5), проверяющая наш список на наличие элементов.

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 5*

Следующей на очереди является функция delete (рис. 6 и рис. 7). В случае, если список пуст, на экран выводится уведомление об этом.

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 6*

Далее предлагается выбрать параметр, по которому будет вестись поиск. Если введённого значение среди данных не оказалось, то на экран выводится уведомление об ошибке.

Если же параметр поиск был введён верно, то далее происходит ввод значения выбранного параметра, а в переменную num кладётся индекс выбранного параметра в ячейке списка.

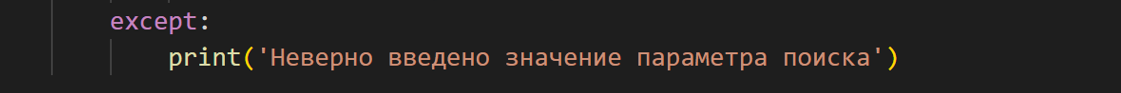
Далее, запускается цикл, пробегающий по элементам списка (рис. 7). При нахождении совпадения происходит вывод на экран информации о покупке и запрос пользователю на подтверждение удаление этой записи. Если пользователь введёт «да», то найденная ячейка будет удалена, цикл остановлен, а в переменную info помещено значение None в качестве маркера. После цикла, если в переменной info лежит None, то выводится уведомление об успешном удалении записи. В противном случае будет выведено: «Продукта с введенной информацией нет в списке».

Изображение выглядит как текст, внутренний, монитор, экран

Автоматически созданное описание

*Рисунок 7*

Если на этапе ввода значения параметра поиска возникнет ошибка ввода, то на экране появится уведомление о соответствующей ошибке (рис. 8).



*Рисунок 8*

Следующая функция sorting отвечает за сортировку списка по определённому параметру (рис. 9). Если список пуст, то на экран выведется соответствующее сообщение. Иначе, пользователю будет задан вопрос об отмене сортировки. При ответе «нет», программа запросит параметр, по которому нужно сортировать. Если такового не окажется, то на экран выведется уведомление о соответствующей ошибке. Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 9*

При правильном вводе, будет произведена сортировка (рис. 10). Далее пользователю будет задан вопрос: «Отсортировать по возрастанию?». При ответе «нет», список будет перевернут.

*Изображение выглядит как текст

Автоматически созданное описание*

*Рисунок 10*

Последней функцией в классе является функция printing (рис. 11), отвечающая за вывод всего списка покупок на экран.

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 11*

А теперь рассмотрим работу данной программы.

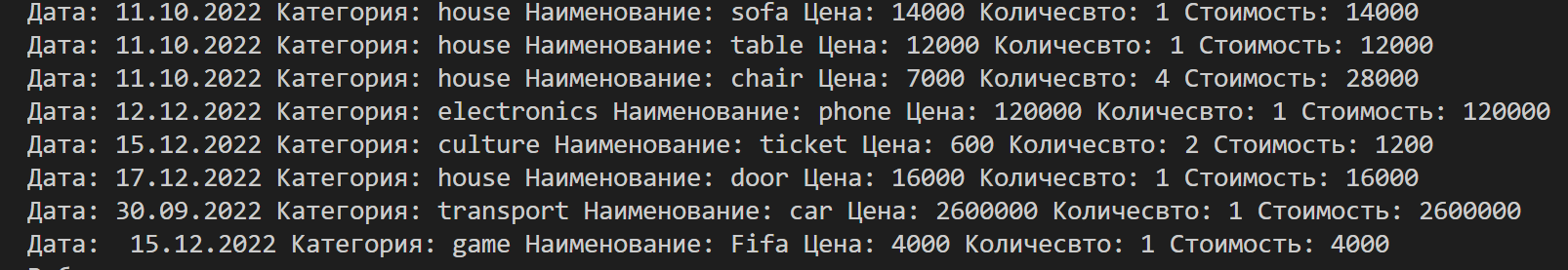
Введём данные покупок в список (рис. 12):

Изображение выглядит как текст

Автоматически созданное описание

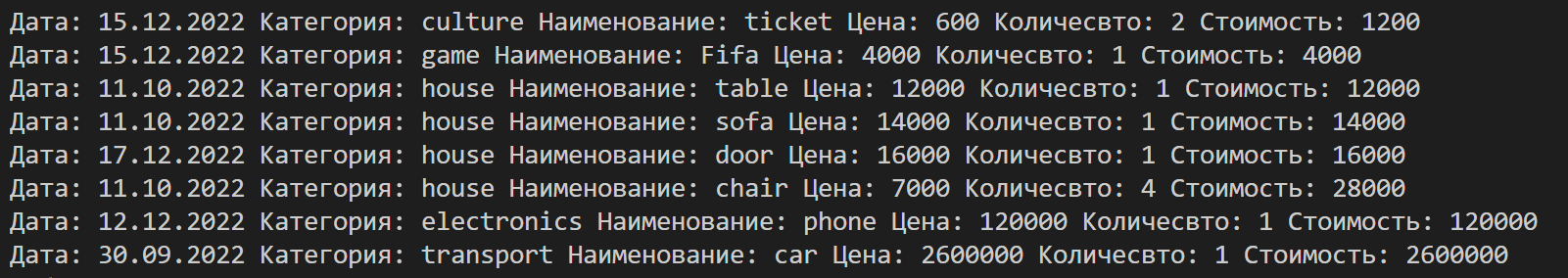
*Рисунок 12*

Все данные в файле (рис. 13)

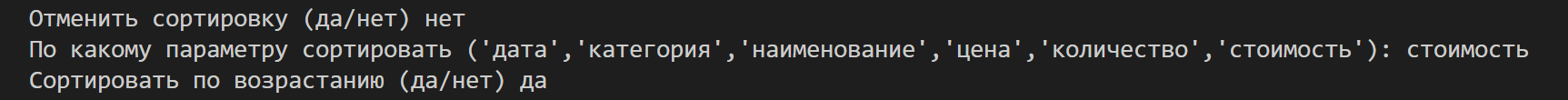


*Рисунок 13*

Теперь отсортируем список по возрастанию стоимости покупок (рис. 14) и посмотрим содержимое списка (рис. 15):



*Рисунок 14*



*Рисунок 15*

Теперь удалим запись о покупке ноутбука (рис. 16) и снова выведем на экран содержимое списка (рис. 17):

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 16*

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 17*

Теперь введём команду «выйти» и посмотрим, как данные записались в файл (рис. 19):

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 17*

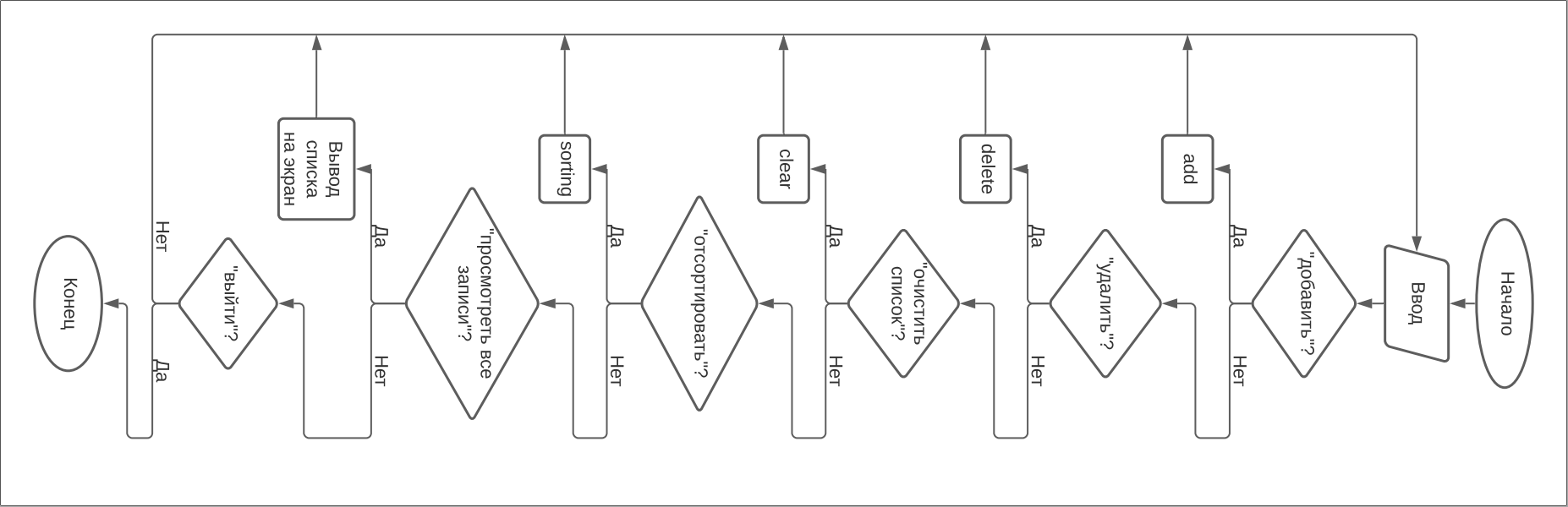
Снова запустим программу и теперь применим команду «очистить список» и снова попробуем вывести на экран записи (рис. 18):

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 18*

Ниже приведена блок-схема, иллюстрирующая работу данной программы (рис. 19).



*Рисунок 19*

**Вывод:**

В ходе данной работы были написаны многочисленные функции для реализации поставленных целей. Также была рассмотрена работа с вводом данных с клавиатуры и файла и выводом данных на экран и в файл. Все поставленные цели программа выполняет.