Министерство цифрового развития, связи и массовых коммуникаций

Ордена трудового Красного Знамени

федеральное государственное бюджетное

образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра «Математическая кибернетика

и информационные технологии»

Отчет по курсовой работе:  
 по дисциплине

«Структуры и алгоритмы обработки данных»

Выполнил: студент группы

БВТ2104

Сокорев Алексей Сергеевич

Проверил(а):

Москва 2023

Содержание:

Оглавление

[Постановка задачи: 3](#_Toc146753786)

[Цель работы: 3](#_Toc146753787)

[Задачи: 3](#_Toc146753788)

[Реализация: 4](#_Toc146753789)

[1.Модуль base.py 4](#_Toc146753790)

[2. Модуль help.py 5](#_Toc146753791)

[3. Модуль shop.py. 6](#_Toc146753792)

[Вывод: 7](#_Toc146753793)

[Список используемой литературы: 7](#_Toc146753794)

[Фотоматериалы/фотоотчёт по проделанной работе: 8](#_Toc146753795)

[1. Модуль base.py 8](#_Toc146753796)

[2. Модуль help.py 11](#_Toc146753797)

[3.Модуль shop.py 14](#_Toc146753798)

# Постановка задачи:

Написать программу, которая будет считывать данные из CSV файла, содержащего информацию о продажах товаров в магазине. Данные в файле содержатся в следующем формате:   
| Номер заказа | Дата заказа | Название товара | Категория товара | Количество продаж | Цена за единицу | Общая стоимость |

# Цель работы:

Научиться считывать данные из CSV - файла и выполнить ряд поставленных заданий, используя алгоритмы обработки данных. Также следует применить те методы сортировок и поиска, которые были описаны в предыдущих лабораторных работах.

# Задачи:

1. Рассчитать общую выручку магазина.

2. Найти товар, который был продан наибольшее количество раз.

3. Найти товар, который принес наибольшую выручку.

4. Составить отчет, содержащий информацию об общей выручке магазина, количестве проданных единиц каждого товар и доле каждого товара в общей выручке. Для решения задач необходимо использовать структуры данных, такие как массивы и хеш-таблицы, а также различные алгоритмы обработки данных, например, сортировку и поиск. Также необходимо учитывать возможные ошибки ввода-вывода и обрабатывать их в соответствии с требованиями.

# Реализация:

## 1.**Модуль base.py** - В нем реализованы базовые функции для работы help модуля

**Функция selection\_sort** - принимает в качестве аргументов два массива (значения в этих двух массивах связаны индексами – они являются характеристиками одного товара). Она сортирует числа по возрастанию и соответственно второй массив (так как число связано с элементом второго массива). Возвращает два отсортированных массива. В качестве алгоритма сортировки выбрана сортировка вставкой – так как она проста в реализации, и мы имеем небольшое кол-во входных данных.

**Функция** **open\_file** – принимает в качестве аргумента файл (в нашем случае table.csv – exel таблицу с данными) Она открывает файл и значения из файла сохраняет в переменную array, в виде вложенного массива. Возвращает переменную в которой хранятся данные из файла.

**Функция hash\_table** – принимает в качестве аргумента переменную с данными файла. Она преобразует вложенный массив в хэш-таблицу, у которой ключ – индекс товара, а значение – хэш-таблица (в ней каждой характеристике товара присвоен свой ключ (пример: name : Samsung Galaxy Tab S8)). Возвращает на выходе хэш-таблицу с индексами товаров и их характеристиками.

**Функция binary\_search** – принимает в качестве аргумента отсортированный массив и искомую переменную. Запускает цикл, в каждой итерации сравнивается искомый элемент с центральным. По результатам сравнения, если искомый элемент меньше центрального – удаляются элементы, идущие после среднего (средний в счет); если искомый элемент больше среднего – удаляются элементы, идущие до среднего (средний в счет); если средний равен искомому – возвращается искомый элемент; если в ходе цикла не удалось найти искомый элемент – то функция возвращает None.

**Функция bubble\_sort** – алгоритм сортировки пузырём(обменом). Принимает в качестве аргумента неотсортированный массив, и флаг (по умолчанию он равен 0). Если указан флаг 0, то алгоритм отсортирует массив по возрастанию, если 1 – то по убыванию (если флаг указан неверно – выдаст ошибку). Возвращает отсортированный массив.

**Функция find\_borders** – вспомогательная функция для нахождения границ для алгоритма поиска Кнут-Моррис-Прат. Принимает в качестве аргумента строку. Возвращает массив чисел.

**Функция knut\_moris\_prat** – алгоритм поиска подстроки в строке. Принимает в качестве аргумента строку и искомую подстроку. Возвращает массив с индексами появлений подстроки в строке.

## 2. **Модуль help.py** – В нем реализован класс ShopHelper который содержит функционал по работе с данными из файла (атрибут в виде хэш-таблицы и различные методы)

**Функция инициализации** – инициализация атрибута класса. В качестве атрибута выступает хэш-таблица с данными о товаре

**Функция info** – вспомогательная функция, которая выводит в консоль хэш-таблицу с данными о товаре. Использует для просмотра и отслеживания данных из таблицы.

**Функция \_\_all\_money** – вспомогательная функция, определенная внутри класса (здесь используется костыль в виде двух нижних подчеркиваний, чтобы она не использовалась вне класса). Она проходится по всем индексам (ключам) товаров, и по ключу [“price”] суммирует выручку конкретного товара. Возвращает переменную содержащую число - общую выручу магазина.

**Функция all-money** – подсчитывает общую выручку магазина. Для расчётов использует вспомогательную функцию \_\_all\_money. Возвращает строку с информацией о общей выручке.

**Функция most\_sales** – проходится по индексам товаров, добавляя в отдельные массивы данные о кол-во продаж и о названии товара. После данного цикла, вызывается функция selection\_sort, которая на выходе возвращает два отсортированных массива. Если по кол-во продаж больше один товар – то функция возвращает строку с информацией о двух элементов массивов, которые берутся по индексу -1. Если есть больше одного товара больших по кол-ву продаж – то функция возвращает строку с информацией о нескольких элементов двух массивов, которые берутся через срез (от конца до первого нахождения с левой стороны (str.count())). Возвращает строку с информацией о товаре с самым большим кол-вом продаж.

**Функция most\_all\_price** – аналогично функции most\_sales за исключением того, что она выводит строку содержащею информацию о товаре с самой большой выручкой.

**Функция percentages** – вспомогательная функция. В качестве аргумента принимает число (информацию об выручке). В вычислениях используется вспомогательная функция \_\_all\_money. Возвращает число – информирующие о том, сколько процентов число (выручка) составляет от общей выручки.

**Функция \_\_report\_of\_gain** – вспомогательная функция. Проходится по индексам товаров – добавляя в новый массив report массивы содержащее информацию об названии товара, кол-во продаж товара, общей выручке товара и о процентном содержании выручки от общей выручки магазина. После цикла сортирует массив по убыванию процентного содержания выручки от общей. Возвращает отсортированный массив.

**Функция report\_of\_gain** – выводит информацию об общей выручке магазина. А также информацию о каждом товаре – названии, кол-во продаж, выручке, процент выручки от общей выручки магазина. Использует вспомогательную функцию \_\_report\_of\_gain. Используется функция из библиотеки tabulate – которая формирует таблицу

**(Доп.) Функция diagram** – выводит диаграмму, показывающею сколько процентов каждый товар занимает. Используется для этого библиотека matplotlib.pyplot. Формируется массив, содержащий проценты выручки от общей, и на его основе строится диаграмма.

(Доп.) **Функция search\_by\_index** – принимает в качестве аргумента искомый индекс. Создаёт массив всех индексов и после его сортирует. После производится поиск индекса в массиве, если он найден – то программа предложить вывести на экран информацию о товаре по этому индексу (Да/нет). Если не наход, то программа выведет на экран строку, сообщающую что товар не найден.

(Доп.) **Функция search\_productmaker** – принимает в качестве аргумента строку (искомого производителя). Инициализируется счетчик. Запускает цикл, в котором мы перебираем по индексам хэш-таблицу. Если в характеристике name искомый производитель нашелся, мы инкрементируем счетчик. В конце, программа если выведет сроку, сообщающую о кол-во товаров проданных от такого-то производителя. Если счетчик пусто – то программа выведет сообщение, что ни одного товара от такого производителя не было продано.

## 3. **Модуль shop.py** – из-этого модуля запускается программа, в ней вызываются большинство функций, реализованных в модуле help.py.

**Функция main** – запуск программы

# Вывод:

В ходе работы я реализовал функционал обработки данных из файла, для этого мне пришлось использовать различные структуру данных и алгоритмы. Учитывал возможные ошибки ввода - вывода и проводил их обработку в соответствии с требованиями.

# Список используемой литературы:

1) Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих (2017) Адитья Бхаргава.

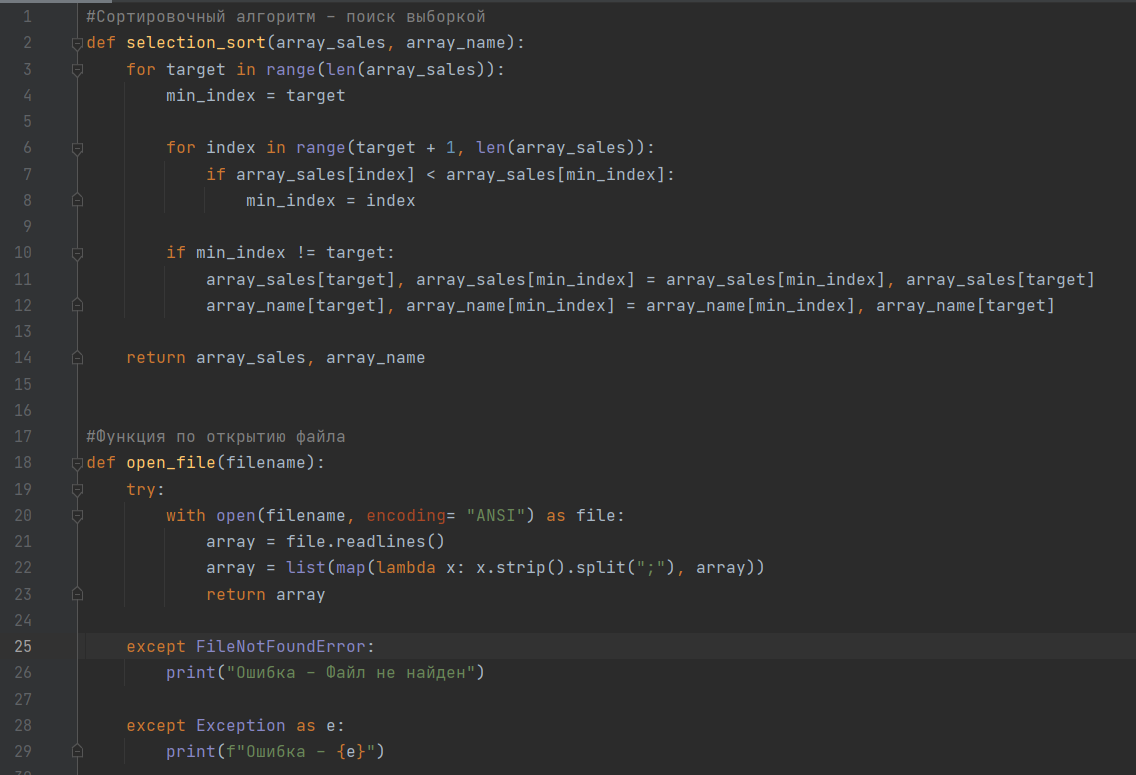
2) Алгоритмы для начинающих. Теория и практика для разработчика (2018). Панос Луридас.

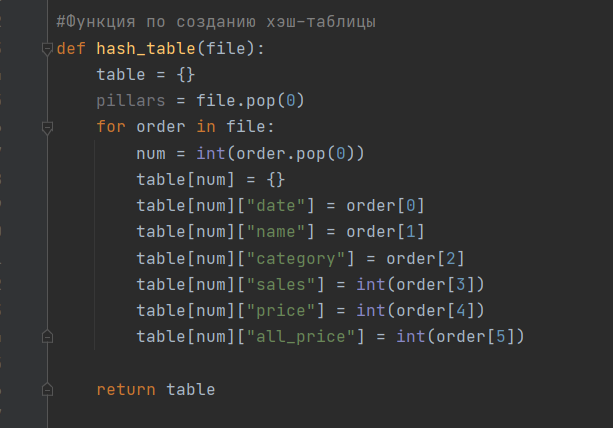
3) Гид по computer science для каждого программиста (2020). Вильям Спрингер.

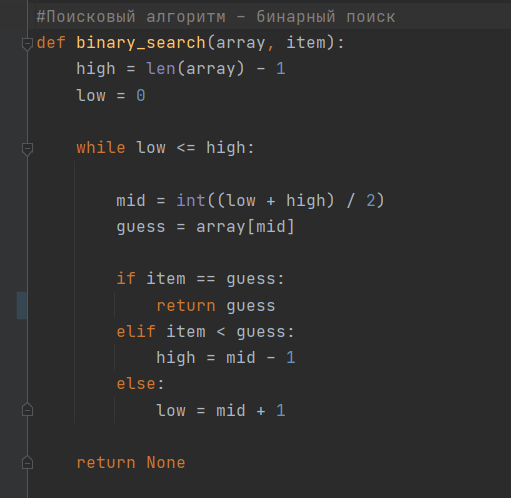
4) Алгоритмы, справочник с примерами на С, С++, Java, Python. (2017) Джордж Хайнеман, Гэри Поллис, Стэнли Селков.

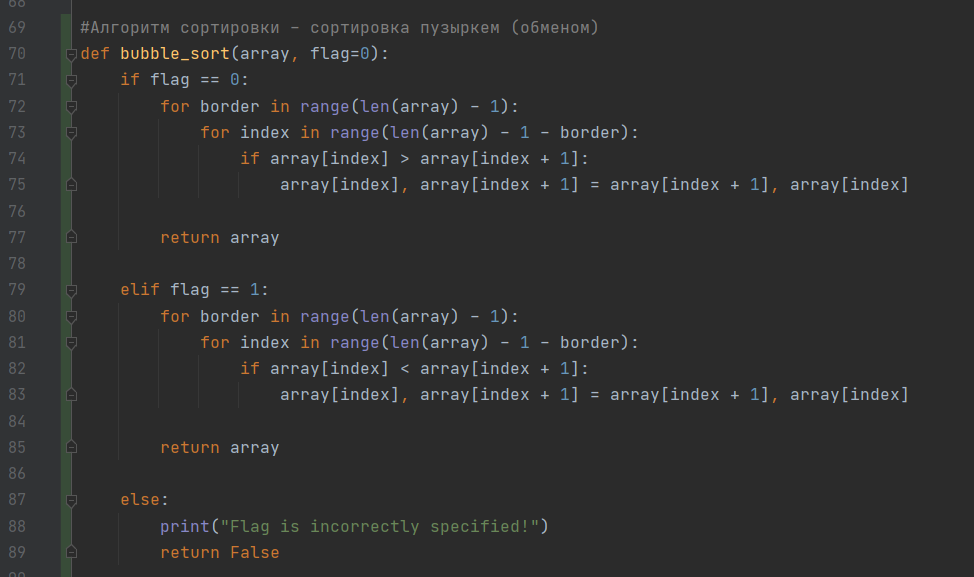
# Фотоматериалы/фотоотчёт по проделанной работе:

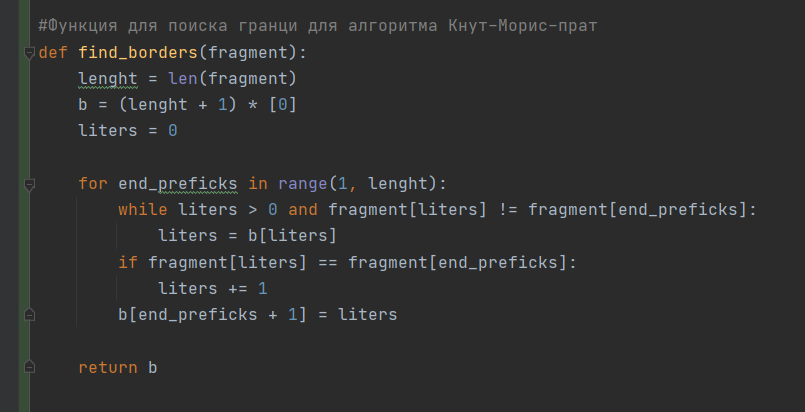
## 1. Модуль base.py

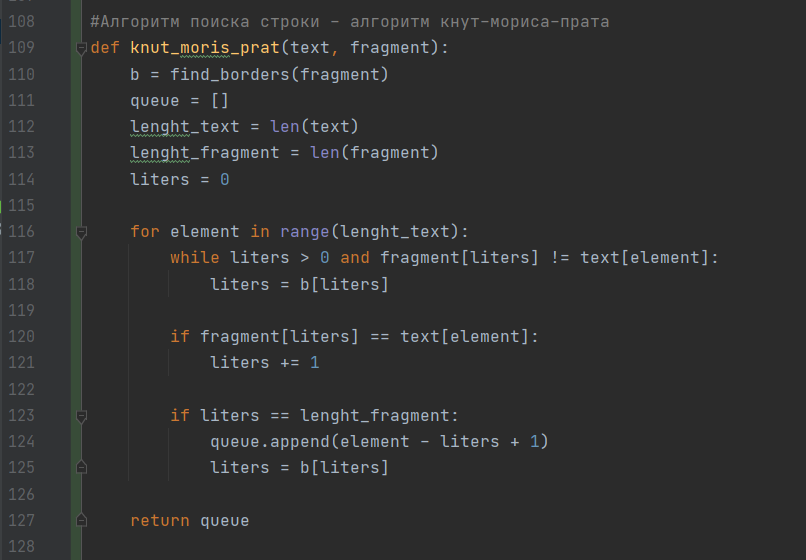




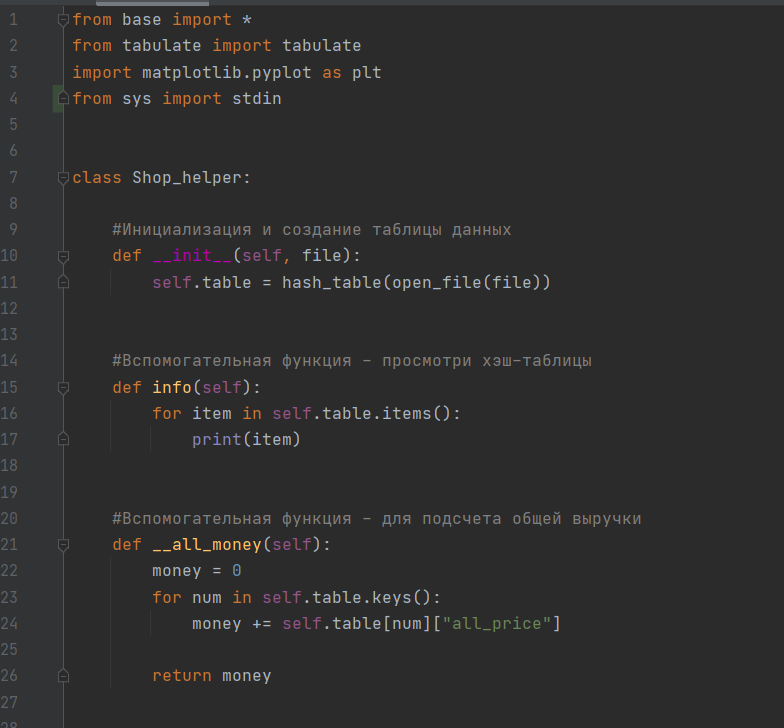


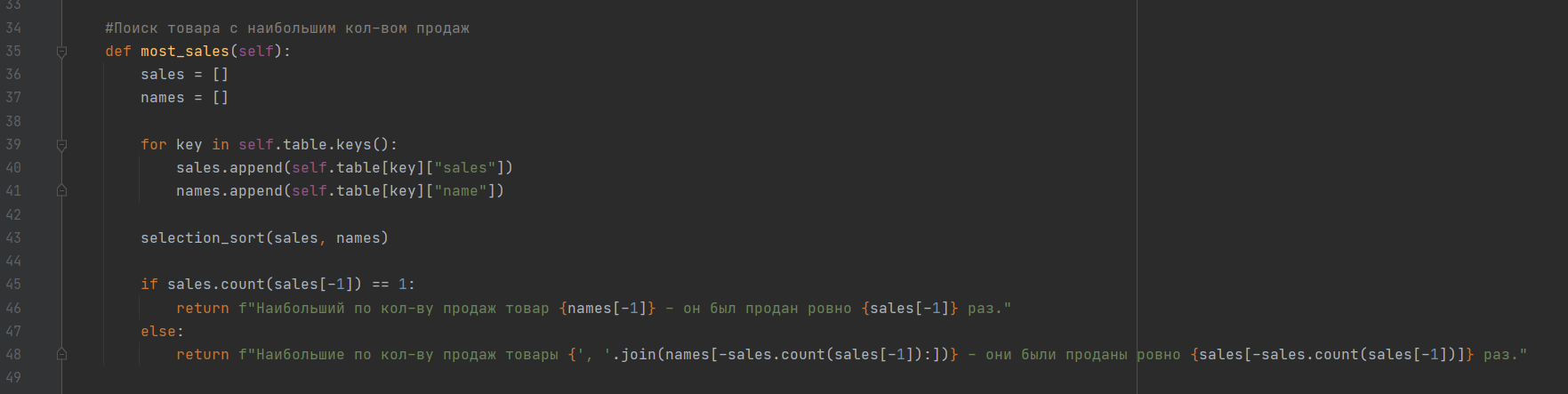






## 2. Модуль help.py

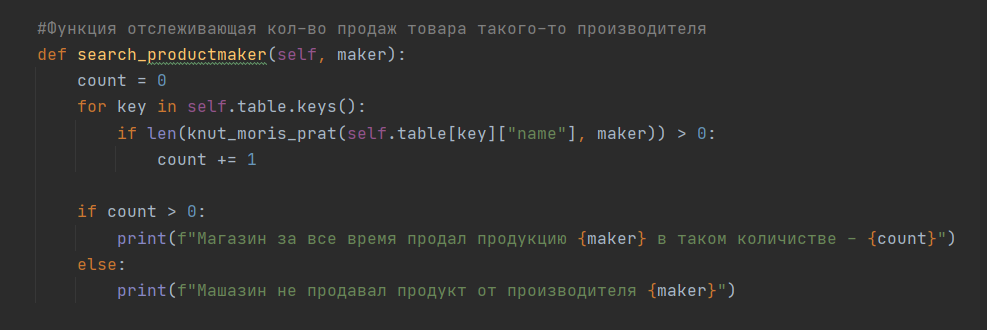












## 3.Модуль shop.py

