Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Ордена Трудового Красного Знамени федеральное государственное бюджетное

образовательное учреждение высшего образования

«Московский технический университет связи и информатики» (МТУСИ)

Кафедра Математической кибернетики и информационных технологий

Курсовая работа по дисциплине

«Структуры и алгоритмы обработки данных»

Выполнил

студент группы БФИ2201:

Балыко В.В.

Проверил: Симонов С.Е.

Москва 2024

Содержание

[Введение 2](#_Toc167349770)

[Ход работы 3](#_Toc167349771)

[Программный код 5](#_Toc167349772)

[Результат выполнения работы 8](#_Toc167349773)

[Вывод 10](#_Toc167349774)

# Введение

Целью данного курсового проекта является разработка системы управления продажами, с помощью которой можно обрабатывать и анализировать данные. Система помогает автоматизировать процесс учета продаж и исключить ошибки при обработке данных.

В современных условиях рынок сталкивается с необходимостью оперативного и точного учета продаж. Это важно для принятия управленческих решений, планирования запасов и оценки эффективности различных товарных категорий. Ручной учет данных может привести к ошибкам и замедлить процессы анализа, что в конечном итоге отрицательно сказывается на прибыльности и конкурентоспособности компании.

# Ход работы

1. Хранение данных в формате CSV:  
 Для хранения данных о продажах был выбран формат CSV, так как он обеспечивает эффективную запись и считывание данных. В программе используется библиотека csv для чтения и записи данных. При запуске программы данные из файла sales.csv считываются и загружаются в дек для обработки.

2. Автоматический расчет показателей:

Для автоматического расчета показателей в программе реализованы следующие функции:

- Общая выручка: Функция подсчитывает общую выручку, суммируя стоимость всех проданных товаров.

- Подсчет количества продаж: Функция подсчитывает количество проданных единиц каждого товара.

- Подсчет общей стоимости продаж: Функция подсчитывает общую стоимость продаж для каждого товара.

- Поиск самого продаваемого товара: Функция находит товар, который был продан наибольшее количество раз.

- Поиск товара с наибольшей выручкой: Функция находит товар, который принес наибольшую выручку.

Эти функции обеспечивают автоматический расчет и обновление показателей при добавлении новых данных о продажах.

3. Анализ данных и генерация отчетов:

На основе введенных данных система генерирует отчеты, включающие информацию:

- Самый продаваемый товар: Для определения этого товара используется функция, подсчитывающая общее количество проданных единиц для каждого товара и выбирающая товар с наибольшим значением.

- Товар с наибольшей выручкой: Для определения этого товара используется функция, подсчитывающая общую стоимость продаж для каждого товара и выбирающая товар с максимальной выручкой.

- Доля каждого товара в общей выручке: Это позволяет определить, какой вклад каждый товар внес в общую прибыль компании.

4. Сохранение и экспорт отчетов:

Для сохранения сгенерированных отчетов в формате CSV реализована функция, которая записывает результаты анализа в новый файл sales\_report.csv.

5. Дополнительные возможности:

- Сортировка данных: Для сортировки данных в отчете используется быстрая сортировка. Она позволяет эффективно упорядочить данные по доле товаров.

- Визуализация данных: Для визуализации данных используется библиотека matplotlib, что позволяет наглядно представить количество продаж товаров.

- Интерфейс ввода данных о товарах:

Интерфейс для ввода данных о продажах был создан с использованием библиотеки tkinter. Он позволяет пользователю вводить информацию о продажах, такую как название товара, категория, количество проданных единиц, цена за единицу и дата продажи. Для обеспечения корректности вводимых данных были реализованы проверки: функция, проверяющая, что название товара содержит только буквы, и функция, проверяющая, что количество и цена содержат только цифры.

# Программный код

**import** csv

**import** matplotlib.pyplot **as** plt

**import** tkinter **as** tk

**from** tkinter **import** messagebox, ttk

**from** datetime **import** datetime

**from** collections **import** deque

*# Чтение данных из CSV-файла*

orders **=** deque()

**with** open('sales.csv', newline**=**'', encoding**=**'utf-8') **as** csvfile:

reader **=** csv**.**reader(csvfile)

header **=** next(reader) *# Пропустить заголовок*

**for** row **in** reader:

orders**.**append(row)

*# Переменные для расчетов*

total\_revenue **=** 0

product\_sales **=** {}

product\_revenue **=** {}

*# Обработка данных*

**while** orders:

order **=** orders**.**popleft() *# Извлекаем заказы из начала дека*

order\_id **=** int(order[0])

order\_date **=** order[1]

product\_name **=** order[2]

product\_category **=** order[3]

quantity\_sold **=** int(order[4])

unit\_price **=** float(order[5])

total\_cost **=** float(order[6])

*# Общая выручка*

total\_revenue **+=** total\_cost

*# Подсчет количества продаж*

**if** product\_name **not** **in** product\_sales:

product\_sales[product\_name] **=** 0

product\_sales[product\_name] **+=** quantity\_sold

*# Подсчет общей стоимости продаж*

**if** product\_name **not** **in** product\_revenue:

product\_revenue[product\_name] **=** 0

product\_revenue[product\_name] **+=** total\_cost

print(f'Общая выручка: {total\_revenue:.2f}')

*# Найти товар, который был продан наибольшее количество раз*

most\_sold\_product **=** **None**

max\_sales **=** 0

**for** product, sales **in** product\_sales**.**items():

**if** sales **>** max\_sales:

max\_sales **=** sales

most\_sold\_product **=** product

print(f'Товар, который был продан наибольшее количество раз: {most\_sold\_product} ({max\_sales} единиц)')

*# Найти товар, который принес наибольшую выручку*

highest\_revenue\_product **=** **None**

max\_revenue **=** 0

**for** product, revenue **in** product\_revenue**.**items():

**if** revenue **>** max\_revenue:

max\_revenue **=** revenue

highest\_revenue\_product **=** product

print(f'Товар, который принес наибольшую выручку: {highest\_revenue\_product} ({max\_revenue:.2f})')

*# Составить отчет*

report **=** []

**for** product **in** product\_sales:

total\_sales **=** product\_sales[product]

total\_income **=** product\_revenue[product]

revenue\_share **=** (total\_income **/** total\_revenue) **\*** 100

report**.**append([product, total\_sales, total\_income, revenue\_share])

*# Сортировка отчета по доле выручки (QuickSort)*

**def** quicksort(arr, low, high):

**if** low **<** high:

pi **=** partition(arr, low, high)

quicksort(arr, low, pi **-** 1)

quicksort(arr, pi **+** 1, high)

**def** partition(arr, low, high):

pivot **=** arr[high][3] *# Доля выручки как опорный элемент*

i **=** low **-** 1

**for** j **in** range(low, high):

**if** arr[j][3] **>** pivot: *# Сортировка по убыванию*

i **+=** 1

arr[i], arr[j] **=** arr[j], arr[i]

arr[i **+** 1], arr[high] **=** arr[high], arr[i **+** 1]

**return** i **+** 1

*# Применение быстрой сортировки к отчету*

quicksort(report, 0, len(report) **-** 1)

*# Печать отчета*

print('\nОтчет:')

**for** item **in** report:

print(f'{item[0]}: {item[1]} ед., {item[2]:.2f}, {item[3]:.2f}%')

*# Сохранить отчет в новый CSV-файл*

**with** open('sales\_report.csv', mode**=**'w', newline**=**'', encoding**=**'utf-8') **as** csvfile:

writer **=** csv**.**writer(csvfile)

writer**.**writerow(['Название товара', 'Количество продаж', 'Общая стоимость', 'Доля выручки'])

**for** item **in** report:

writer**.**writerow(item)

print('\nОтчет сохранен в файл sales\_report.csv')

*# Построить гистограмму для всех товаров*

product\_names **=** list(product\_sales**.**keys())

sales\_counts **=** list(product\_sales**.**values())

plt**.**figure(figsize**=**(12, 8))

bars **=** plt**.**bar(product\_names, sales\_counts, color**=**'thistle')

plt**.**title('Количество продаж по товарам')

*# Добавить метки на столбцах*

**for** bar **in** bars:

yval **=** bar**.**get\_height()

plt**.**text(bar**.**get\_x() **+** bar**.**get\_width() **/** 2.0, yval, int(yval), ha**=**'center', va**=**'bottom')

plt**.**tight\_layout()

plt**.**show()

# Результат выполнения работы

На рисунке 1 показано, как реализовано добавление данных о товаре: пользователь может ввести название товара, выбрать категорию из выпадающего списка или создать свою, ввести количество товара и цену за единицу.

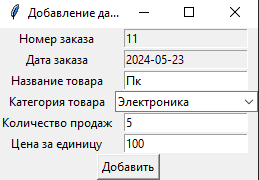


Рисунок 1 - Окно с добавлением данных о товаре

На рисунке 2 видно, что программа действительно добавила новый товар с номером: 11, датой: 2024-05-23, названием: “Пк”, категорией: “Электроника”, количеством продаж: 5 и ценой за единицу: 100.  
 Также программа автоматически подсчитала общую стоимость.

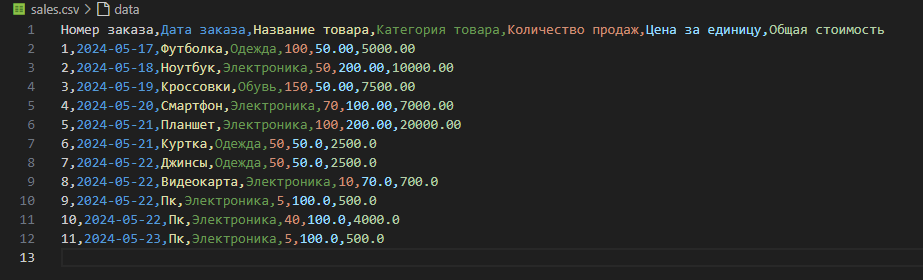


Рисунок 2 - Результат добавления товара

На рисунке 3 программа обрабатывает неправильно введенный формат данных и сообщает пользователю,

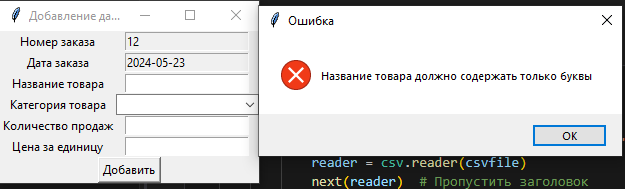


Рисунок 3 - Обработка ошибок

На рисунке 4 представлен отчет, в котором содержится вся необходимая информация.

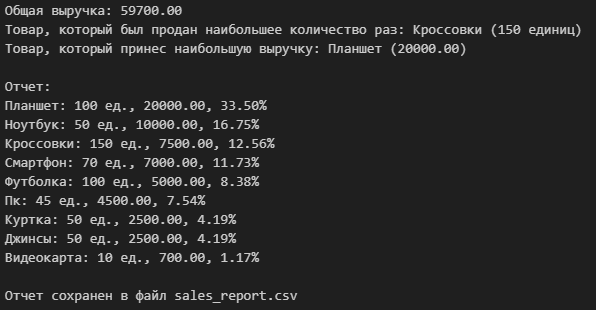


Рисунок 4 - Сгенерированный программой отчет

На рисунке 5 изображен график, который показывает количество продаж каждого товара.

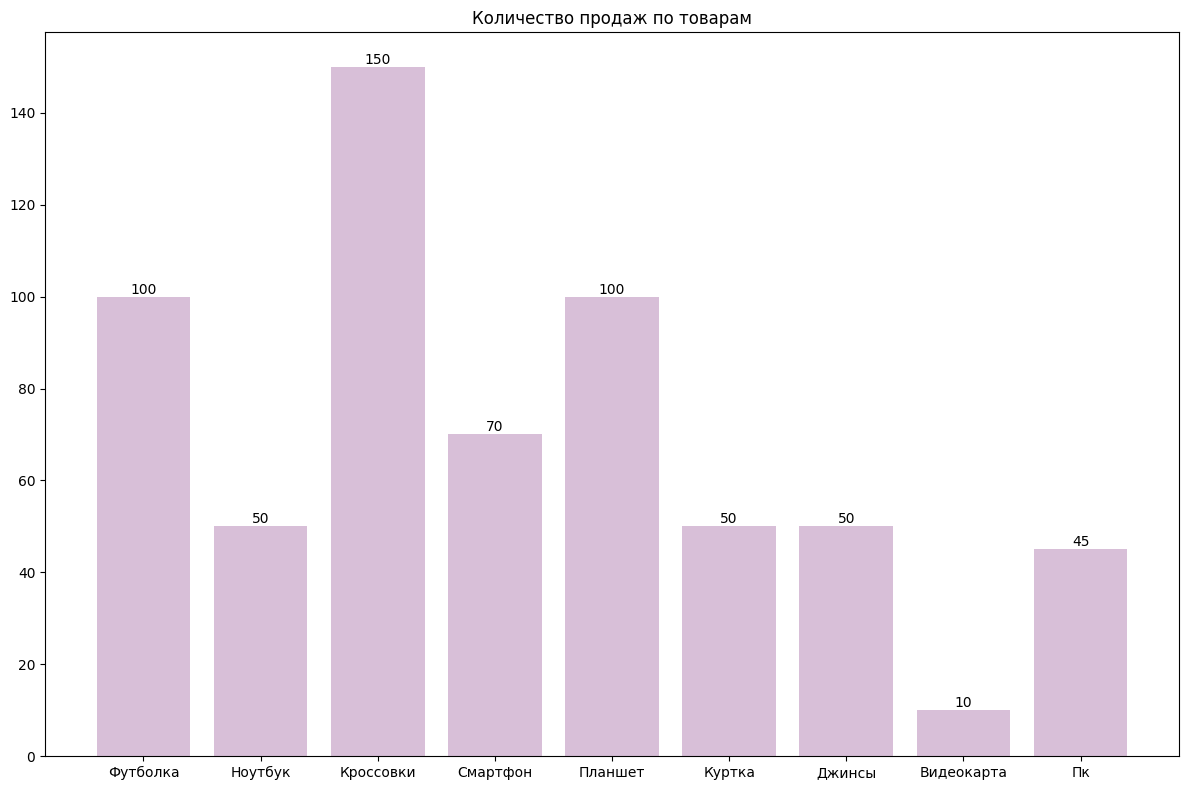


Рисунок 5 - График количества продаж товаров

# Вывод

Разработанная программа является эффективным инструментом для анализа данных о продажах. Она позволяет получить информацию об общей выручке, популярных товарах и их вкладе в общий доход.

Использование хеш-таблиц, массивов, алгоритмов сортировки и дека позволило оптимизировать работу программы, повысить ее производительность.