

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»
Кафедра «Математической кибернетики и информационных технологий»

Курсовая работа
По дисциплине «Структуры и алгоритмы обработки данных»

Выполнил: Студенты группы БФИ2201
Ле Хюи Туан
Проверил:
Павликов А.Е

Москва
2024

Оглавление

Введение.....	3
Техническое задание.....	4
Заключение	9

Введение

Была разработана программа для анализа данных из CSV файла. Программа подсчитывает общую выручку магазина, определяет наиболее продаваемый товар и товар, приносящий наибольшую выручку, а также формирует отчет с информацией о товарах. Для решения этих задач использовались массивы и собственная реализация хэш-таблиц с уникальной хэш-функцией. Также применялись алгоритмы сортировки, такие как сортировка Шелла, и бинарный поиск.

1. **Анализ данных** - процесс исследования и интерпретации данных с целью получения информации и выводов.
2. **CSV (Comma-Separated Values)** - формат текстового файла для представления табличных данных, где каждая строка представляет собой одну запись, разделенную запятыми.
3. **Выручка** - сумма денег, полученных от продажи товаров или услуг за определенный период времени.
4. **Ключевые показатели** - основные числовые характеристики, которые используются для оценки эффективности бизнеса или процессов.
5. **Подробный отчет** - документ или презентация, содержащая подробную информацию о результатах анализа данных, представленных в удобной и понятной форме.
6. **Прибыль** - разница между выручкой и издержками на производство и продажу товаров или услуг, которая остается у предприятия в виде чистой прибыли.

Описание программы с использованием

Программа написана на Python на Jupyter и состоит из нескольких функций, каждая из которых выполняет определенную задачу для правильного функционирования проекта.

Техническое задание

Техническое задание для данной курсовой работы включает следующие требования: разработать программу, которая считывает данные из CSV файла с информацией о продажах товаров в магазине. Формат данных в файле:

| Номер заказа | Дата заказа | Название товара | Категория товара |
Количество продаж | Цена за единицу | Общая стоимость |.

Необходимо выполнить следующие задачи:

1. Вычислить общую выручку магазина.
2. Определить товар с наибольшим количеством продаж.
3. Найти товар, который принес максимальную выручку.
4. Создать отчет с общей информацией о выручке магазина, количестве проданных единиц каждого товара и доле каждого товара в общей выручке.

Для решения этих задач нужно использовать массивы и хеш-таблицы, а также применять различные алгоритмы обработки данных, такие как сортировка и поиск. Важно также предусмотреть обработку возможных ошибок ввода-вывода в соответствии с требованиями.

. **csv** для чтения данных из CSV файла.

. **Matplotlib.pyplot** — графическая библиотека для создания визуализаций

```
# Функция для чтения данных из CSV-файла
def read_csv_file(file_path):
    data = []
    try:
        with open(file_path, 'r', newline="", encoding='utf-8') as file:
            reader = csv.reader(file)
            next(reader) # Пропустить заголовочную строку
            for row in reader:
                data.append(row)
    except Exception as e:
        print(f"Ошибка чтения файла {file_path}: {e}")
    return data
```

Этот код определяет функцию с именем `read_csv_file`, которая используется для чтения данных из файла CSV. Вот суть кода:

Открыть файл CSV: функция открывает файл CSV, указанный в параметре `file_path`, для чтения данных из этого файла.

Чтение данных: функция использует цикл для чтения каждой строки файла CSV с помощью модуля `csv.reader()`. Функция `next(reader)` используется для пропуска первой строки в CSV-файле, предполагая, что эта строка содержит заголовки столбцов.

Обработка ошибок. Любые ошибки, возникающие при чтении файла CSV, обрабатываются путем распечатки сообщения об ошибке вместе с именем файла и конкретным сообщением об ошибке.

Возврат данных: после чтения всего файла CSV данные возвращаются в виде списка строк, каждая строка представляет собой список значений, соответствующих столбцам в файле CSV.

```
# Функция для вычисления процента проданных товаров
def calculate_percentage_sold(data):
    total_quantity_sold = sum([int(row[4]) for row in data])
    percentage_sold = {}
    for row in data:
        product = row[2]
        quantity_sold = int(row[4])
        percentage = (quantity_sold / total_quantity_sold) * 100
        percentage_sold[product] = percentage
    return percentage_sold
```

Этот код определяет функцию с именем `Calculate_percentage_sold`, которая используется для расчета процента от количества проданных продуктов в предоставленных данных. Вот суть кода:

Подсчитать общее количество проданных продуктов. Функция вычисляет общее количество проданных продуктов, проходя каждую строку данных и суммируя количество проданных продуктов из соответствующего столбца (здесь 5-й столбец данных).

Вычислите процент проданных продуктов: как только вы получите общее количество проданных продуктов, функция перебирает каждую строку данных и вычисляет процент количества проданных продуктов по сравнению с общим количеством проданных продуктов. Результаты сохраняются в словаре с названием продукта в качестве ключа и процентом проданных товаров в качестве значения.

Возврат результатов. Наконец, словарь содержит процент каждого продукта, возвращаемого функцией.

Функция поиска самого продаваемого товара в файле

```
def save_results(file_path, total_revenue, most_sold_product,
highest_revenue_product, percentage_sold):
    try:
        with open(file_path, 'w', newline="", encoding='utf-8') as file:
            file.write(f"Общий доход: ${total_revenue}\n")
            file.write(f"Самый продаваемый товар: {most_sold_product}\n")
            file.write(f"Товар с наибольшим доходом: {highest_revenue_product}\n")
            file.write("\n")
            file.write("Процент продажи каждого товара:\n")
            for product, percentage in percentage_sold.items():
                file.write(f"{product}: {percentage:.2f}%\n")
            print("Результаты сохранены в файл успешно.")
    except Exception as e:
        print(f"Ошибка при сохранении результатов в файле {file_path}: {e}")
```

```

# Путь к CSV-файлу
file_path = 'sales_data.csv'
# Путь к файлу результатов
output_file_path = 'results.txt'
try:
    # Чтение данных из CSV-файла
    data = read_csv_file(file_path)

    # Вычисление общего дохода
    total_revenue = sum(float(row[4]) * float(row[5]) for row in data)

    # Поиск самого продаваемого товара
    most_sold_product = max(data, key=lambda x: int(x[4]))[2]

    # Поиск товара с наибольшим доходом
    highest_revenue_product = max(data, key=lambda x: float(x[4]) *
float(x[5]))[2]

    # Вычисление процента продажи каждого товара
    percentage_sold = calculate_percentage_sold(data)

    # Сохранение результатов в текстовом файле
    save_results(output_file_path, total_revenue, most_sold_product,
highest_revenue_product, percentage_sold)
    # Построение графика
    products = list(percentage_sold.keys())
    percentages = list(percentage_sold.values())
    plt.figure(figsize=(8, 6))
    plt.bar(products, percentages, color='skyblue')
    plt.xlabel('Товар')
    plt.ylabel('Процент (%)')
    plt.title('Диаграмма процента продажи каждого товара')
    plt.xticks(rotation=45, ha='right')
    plt.show()
except FileNotFoundError:
    print("Файл не найден.")
except Exception as e:
    print(f"Произошла ошибка: {e}")

```

Этот код определяет функцию с именем `save_results`, которая используется для сохранения результатов вычислений в текстовый файл. Вот суть кода:

Запись результатов в текстовый файл: функция открывает или создает текстовый файл по пути, указанному в параметре `file_path`. Затем он записывает в этот файл результаты вычислений, включая общий доход (`total_revenue`), самый продаваемый продукт (`most_sold_product`), продукт с самым высоким доходом (`highest_revenue_product`) и процент проданных продуктов (`percentage_sold`).

Обработка ошибок. Если при записи результатов в текстовый файл возникает какая-либо ошибка, она обрабатывается путем печати определенного сообщения об ошибке.

В код также входит исполняемый файл, предназначенный для чтения данных из CSV-файла, расчета необходимых метрик и последующего вызова функции `save_results` для сохранения результатов в текстовый файл. Наконец, он рисует диаграмму, чтобы визуализировать процент проданной продукции.

Заключение

Была создана программа, обрабатывающая CSV-файл и выводящая статистику по продажам магазина. Программа также визуализирует данные в виде столбчатых диаграмм, что позволяет наглядно представить информацию о продажах по товарам и категориям.

≡ results.txt

```
1  Общий доход: $1146300.0
2  Самый продаваемый товар: NIKE
3  Товар с наибольшим доходом: JORDAN
4
5  Процент продажи каждого товара:
6  NIKE: 52.46%
7  ADIDAS: 2.34%
8  GUCCI: 0.94%
9  JORDAN: 1.87%
10 CHANEL: 7.87%
11
```

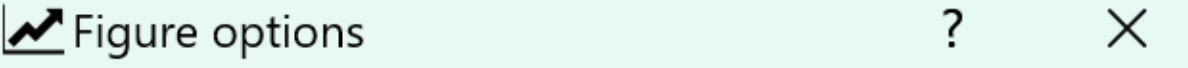


Figure options?×

Axes

Title

ra của mỗi sản phẩm

X-Axis

Min

0.0

Max

5880.0

Label

Số lượng bán ra

Scale

linear

Y-Axis

Min

-0.6400000000000001

Max

4.6400000000000001

Label

Sản phẩm

Scale

linear

(Re-)Generate automatic legend

☐

OK

Cancel

Apply

Общая выручка магазина: 912300.0000

Товар, который был продан наибольшее количество раз: NIKE (6190 piece.)

Товар, который принес наибольшую выручку: JORDAN (317000.00 \$.)

ОТЧЕТ ПО КАЖДОМУ ТОВАРУ:

Название товара	Количество продаж	Доля в общей выручке
NIKE	6190	24.6081%
ADIDAS	500	12.8795%
GUCCI	145	3.0144%
JORDAN	2500	34.7473%
CHANEL	1340	24.7506%

