

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 «Математическая кибернетика и информационные технологии»

Курсовая работа

по дисциплине

«Структуры и алгоритмы обработки данных»

Тема: «Программа: Продажа товаров в магазине»

Выполнил:

студент группы БФИ2201

Елонго дидо Лунге

Проверил:

Павликов А.Е.

Москва 2024

Оглавление

1.Введение.....	3
2.Этапы выполнения	4
3.Представление найденных данных в файле CSV	8
4.Результат выполнения	9
5.Алгоритмы и структуры данных	10
Заключение	11

Программа: Продажа товаров в магазине.

1. Введение

Цель данной программы — анализировать данные о продажах товаров в магазине, представленные в формате CSV. Программа считывает данные из файла, вычисляет ключевые показатели (общую выручку, самый продаваемый товар и товар с наибольшей выручкой), а также создает подробный отчет. Это помогает понять, какие товары наиболее популярны и приносят наибольшую прибыль, что важно для принятия бизнес-решений.

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.

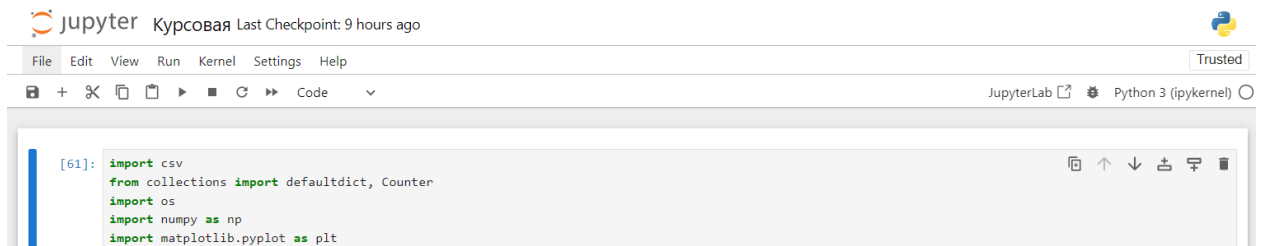
6. **Прибыль** - разница между выручкой и издержками на производство и продажу товаров или услуг, которая остается у предприятия в виде чистой прибыли.

Описание программы с использованием JUPYTER

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

2. Этапы выполнения

Импорт необходимых модулей



- **csv** для чтения данных из CSV файла.
- **defaultdict** и **Counter** из **collections** для удобной работы со словарями и подсчетами.
- **NumPy** — это библиотека, обеспечивающая поддержку многомерных массивов, а также множество полезных математических функций.
- **Matplotlib.pyplot** — графическая библиотека для создания визуализаций.

Функции данных в файле CSV:

```
def read_sales_data(file_path):
    sales_data = []
    try:
        with open(file_path, mode='r', encoding='utf-8') as file:
            reader = csv.DictReader(file)
            sales_data = [row for row in reader]
    except Exception as e:
        print(f"Error reading file {file_path}: {e}")
    return sales_data
```

Этот код определяет функцию с именем `read_sales_data`, которая принимает путь к файлу в качестве параметра. Внутри функции используется блок `try-Exception` для обработки потенциальных ошибок при чтении файла.

В блоке `try` код открывает указанный файл в режиме чтения ('r') и использует `csv.DictReader` для чтения его содержимого в виде словарей. Прочитанные строки сохраняются в списке под названием «`sales_data`».

Если при попытке чтения файла возникает исключение (например, если файл не существует или существуют проблемы с разрешениями), исключение перехватывается и отображается на экране с сообщением о том, какой файл вызвал проблему.

Наконец, независимо от ситуации (успех или ошибка), функция возвращает список «`sales_data`», который содержит данные, извлеченные из файла CSV.

Как работает расчет общего оборота

```
def calculate_total_revenue(sales_data):
    total_revenue = 0
    for item in sales_data:
        total_revenue += float(item['Общая стоимость'])
    return total_revenue
```

- Принимает список словарей sales_data.
- Вычисляет общую выручку, суммируя значение ключа Общая стоимость для каждого элемента в sales_data.
- Возвращает общую выручку total_revenue.

Функция поиска самого продаваемого товара в файле

```
def find_most_sold_product(sales_data):
    product_sales = {}
    for item in sales_data:
        product_name = item['Название товара']
        if product_name not in product_sales:
            product_sales[product_name] = {'name': product_name, 'quantity': 0}
        product_sales[product_name]['quantity'] += int(item['Количество продаж'])
    sorted_products = sorted(product_sales.values(), key=lambda x: x['quantity'], reverse=True)
    return sorted_products[0]['name'], sorted_products[0]['quantity']
```

- Принимает список словарей sales_data.
- Использует счетчик для подсчета количества продаж каждого продукта.
- Возвращает самый продаваемый продукт most_sold_product в формате кортежа (название продукта, количество).
- find_most_sold_product», который принимает в качестве входных данных словарь данных о продажах.
- Затем функция сортирует товары по их проданному количеству в порядке убывания.

Функция поиска продукта с наибольшим оборотом в файле CSV.

```
def find_highest_revenue_product(sales_data):
    product_revenue = defaultdict(float)
    for item in sales_data:
        product_revenue[item['Название товара']] += float(item['Общая стоимость'])
    highest_revenue_product = max(product_revenue, key=product_revenue.get)
    return [highest_revenue_product, product_revenue[highest_revenue_product]]
```

- Принимает список словарей sales_data.
- Использует defaultdict для расчета общего дохода для каждого продукта.
- Возвращает продукт с наибольшим доходом plus_revenue_product в формате кортежа (название продукта, доход).
- функция определяет, какой продукт принес наибольший доход, используя метод max() для ключей словаря product_revenue на основе их соответствующих значений.

Функция генерации отчетов

```
def generate_report(sales_data):
    total_revenue = calculate_total_revenue(sales_data)
    most_sold_product, most_sold_quantity = find_most_sold_product(sales_data)
    highest_revenue_product, highest_revenue = find_highest_revenue_product(sales_data)

    product_sales = Counter()
    product_revenue = defaultdict(float)

    for item in sales_data:
        product_sales[item['Название товара']] += int(item['Количество продаж'])
        product_revenue[item['Название товара']] += float(item['Общая стоимость'])

    report = {
        'total_revenue': total_revenue,
        'most_sold_product': most_sold_product,
        'most_sold_quantity': most_sold_quantity,
        'highest_revenue_product': highest_revenue_product,
        'highest_revenue': highest_revenue,
        'product_sales': dict(product_sales),
        'product_revenue': dict(product_revenue)
    }

    return report
```

- Создайте отчет, содержащий общий доход, самый продаваемый продукт в магазине, самый прибыльный продукт и подробную информацию по каждому продукту.
- Используйте словарь для вызова функций.
- Возвращает отчет как отчет по словарю.

Функция печати программы

```
def print_report(report):
    print(f"Общая выручка магазина: {report['total_revenue']:.4f}")
    print(
        f"Товар, который был продан наибольшее количество раз: {report['most_sold_product']} ({report['most_sold_quantity']} piece.)"
    )
    print(
        f"Товар, который принес наибольшую выручку: {report['highest_revenue_product']} ({report['highest_revenue']:.2f} $.)"
    )

    print("\nОТЧЕТ ПО КАЖДОМУ ТОВАРУ:")

    print(f"{'Название товара':<40} {'Количество продаж':<30} {'Доля в общей выручке':<30}")
    for product, sales in report['product_sales'].items():
        revenue_share = (report['product_revenue'][product] / report['total_revenue']) * 100
        print(f"product:<40} {sales:<30} {revenue_share:.4f}%")
```

- Отчет о запуске и публикации.
- Обратите внимание, что данные вам инструкции точны.
- Подробная рассылка, нажмите кнопку «Продать» и поделитесь общим доходом.
- печать каждого товара в магазине
- чтение темы и количества желаемого продукта

Основная часть программы

```
print(T {product:<40> {sales:<30> {revenue_share:<47>}} )
if __name__ == "__main__":
    file_path = 'sales_data.csv'
    sales_data = read_sales_data(file_path)

    if sales_data:
        report = generate_report(sales_data)
        print_report(report)
```

- Устанавливает путь к файлу sales_data.csv.
- Считывает данные из файла с помощью read_sales_data.
- Если данные считаны успешно, генерирует отчет и распечатывает его.
- он генерирует отчет, используя эти данные, с помощью функции generate_report(), а затем отображает этот отчет, вызывая функцию print_report().

Код Представление результата в процентах (с помощью графика)

```
products = list(report['product_sales'].keys())
quantities = list(report['product_sales'].values())

plt.figure(figsize=(8, 4))
plt.pie(sizes, labels=labels, autopct='%1.1f%%')
plt.axis('equal')
plt.xlabel('Products')
plt.ylabel('Доля в общей выручке')
plt.title('Magasin ELONGOshop', size=14)
plt.show()
```

Код представляет собой комбинацию Python и библиотеки Matplotlib для создания круговой диаграммы на основе данных, содержащихся в словаре report['product_sales']. Продукты извлекаются в виде списка с помощью метода keys(), а проданные количества извлекаются в виде списка с помощью метода values().

Кодовое представление результата (с использованием диаграммы)

```
products = list(report['product_sales'].keys())
quantities = list(report['product_sales'].values())

plt.figure(figsize=(4, 4))
plt.bar(products, quantities)
plt.xlabel('Products')
plt.ylabel('Количество продаж')
plt.title('Most Sold Products')
plt.show()
```

Этот код Python использует библиотеку Matplotlib для создания гистограммы, отображающей продажи продукции. Вот что делает каждая строка:

1. `products = list(report['product_sales'].keys())`: эта строка извлекает ключи из словаря 'product_sales', содержащегося в отчете, и сохраняет их в списке под названием 'products'. Эти ключи представляют названия продуктов.
2. `quantities = list(report['product_sales'].values())`: эта строка извлекает значения из словаря 'product_sales' (которые представляют собой проданные количества) и сохраняет их в списке под названием 'quantities'.
3. `plt.figure(figsize=(4, 4))`: эта строка создает новую фигуру размером 4 на 4 дюйма для отображения графика.

4. `plt.bar(продукты, количества)`: эта строка создает гистограмму, где ось X представляет продукты (содержащиеся в списке "продукты"), а ось Y представляет проданные количества (содержащиеся в списке "продукты"). список количеств).



5. `plt.xlabel('Products')`: устанавливает метку оси X как «Продукты».

6. `plt.ylabel('Количество продаж')`: устанавливает метку оси Y на русском языке, вероятно, обозначающую «Количество продаж».

7. `plt.title('Самые продаваемые товары')`: это дает диаграмме заголовок, например «Самые продаваемые товары».

8. `plt.show()`: Наконец, эта команда отображает полную диаграмму со всеми ее элементами, как определено ранее.

3.Представление найденных данных в файле CSV

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12	Номер заказа,Дата заказа,Название товара,Категория товара,Количество продаж,Цена за единицу,Общая стоимость,,,									
13										
14	2024-04-02,BASKET,NIKE,1,50,700,3500,,,									
15	2024-02-01,SHORT,ADIDAS,2,250,20,5000,,,									
16	2024-05-03,WATCH,GUCCI,3,45,5000,22500,,,									
17	2024-05-04,BASKET,JORDAN,4,300,980,294000,,,									
18	2024-07-01,SHORT,NIKE,5,540,150,81000,,,									
19	2024-02-04,SHORT,JORDAN,6,2000,10,20000,,,									
20	2024-05-06,COLOR,GUCCI,7,100,50,5000,,,									
21	2024-05-07,BASKET,ADIDAS,8,250,450,112500,,,									
22	2024-05-08,SHORT,NIKE,9,5600,25,140000,,,									
23	2024-05-09,BAC,JORDAN,10,200,15,3000,,,									
24	2024-05-10,BAC,CHANEL,11,500,250,125000,,,									
25	2024-05-11,BASKET,CHANEL,12,840,120,100800,,,									

4.Результат выполнения

Общая выручка магазина: 912300.0000

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

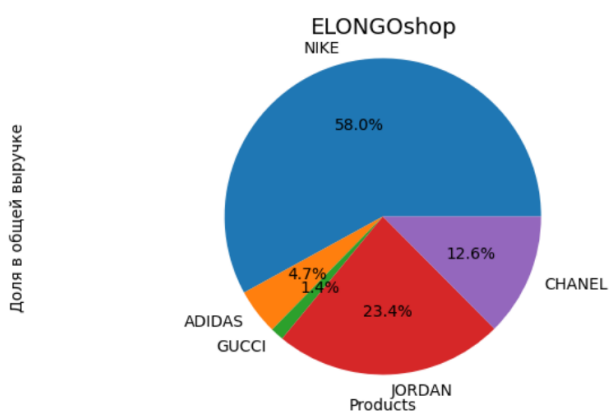
Товар, который принес наибольшую выручку: JORDAN (317000.00 \$.)

ОТЧЕТ ПО КАЖДОМУ ТОВАРУ:

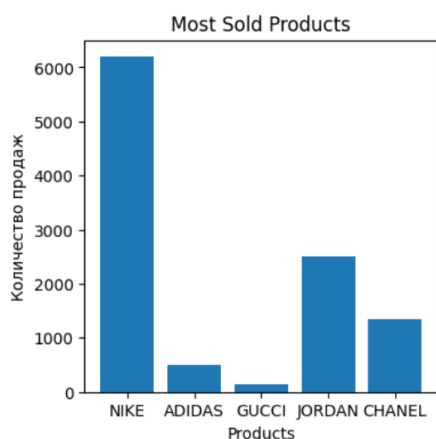
Название товара	Количество продаж	Доля в общей выручке
NIKE	6190	24.6081%
ADIDAS	500	12.8795%
GUCCI	145	3.0144%
JORDAN	2500	34.7473%
CHANEL	1340	24.7506%

ELONGOshop

Результат выполнения графика



Результат выполнения диаграммы



5. Алгоритмы и структуры данных

Используемые структуры данных:

- **Списки:** для хранения строк из CSV файла в виде словарей. Мы используем список для хранения строк файла CSV в качестве словарей, поскольку каждая строка файла CSV обычно представляет собой запись с разными полями. При использовании списка словарей каждый элемент в списке соответствует записи и представлен словарем, где ключами являются имена столбцов, а значениями — данные, связанные с этими столбцами. Это обеспечивает организованную и легкодоступную структуру для манипулирования и обработки данных, извлеченных из файла CSV.
- **Словари:** для хранения данных о продажах, выручке и для генерации отчета. Словари часто используются для хранения данных о продажах, доходах и для создания отчетов благодаря своей структуре «ключ-значение». В контексте продаж и доходов это позволяет связать конкретную информацию с каждым ключевым элементом. Например, мы можем использовать название продукта в качестве ключа и связать с этим ключом такие данные, как проданное количество, цена за единицу и т. д. Эта структура упрощает доступ к соответствующим данным при создании отчетов, поскольку можно быстро получить доступ к информации, связанной с данным ключом. Кроме того, словари обеспечивают большую гибкость в манипулировании и организации данных, что делает их популярным выбором для управления информацией, связанной с продажами и доходами.
- **Counter:** для подсчета количества продаж каждого товара. Счетчик используется для подсчета количества продаж каждого продукта, поскольку он помогает точно и автоматически отслеживать количество проданных единиц. Это позволяет легче управлять запасами, анализировать тенденции продаж, а также отслеживать индивидуальную эффективность каждого продукта.
- **defaultdict:** для подсчета общей выручки для каждого товара. Мы используем defaultdict для расчета общего дохода по каждому продукту, поскольку он позволяет создать словарь со значениями по умолчанию для ключей, которые еще не существуют. Это позволяет избежать необходимости проверять, существует ли уже ключ в словаре, прежде чем добавлять к нему значение, что упрощает и оптимизирует процесс расчета общего дохода для каждого продукта.

Используемые алгоритмы:

- **Суммирование:** для вычисления общей выручки магазина.
Алгоритм суммирования используется для расчета общего оборота магазина, поскольку он позволяет эффективно суммировать объемы продаж, совершенных за определенный период. Применяя этот алгоритм, можно быстро и точно получить общую сумму дохода от реализованных продаж, что важно для оценки общих финансовых показателей магазина.
- **Поиск максимума:** для нахождения товара с наибольшей выручкой.
Алгоритм максимального поиска используется для поиска продукта с наибольшим доходом, поскольку позволяет просматривать список продуктов и сравнивать доходы, связанные с каждым из них. Найдя продукт с максимальным доходом, можно быстро определить, какой из рассмотренных продуктов приносит наибольший доход. Это облегчает принятие решений, выделяя продукт, который приносит наибольший вклад в общий доход.
- **Подсчет:** для определения самого продаваемого товара.
Алгоритм подсчета часто используется для определения самого продаваемого товара, поскольку он может эффективно подсчитывать количество вхождений каждого товара в набор данных. Просто подсчитав количество продаж каждого товара и определив тот, у которого наибольшее число, можно легко определить, какой товар продается лучше всего. Это делает его быстрым и простым методом определения популярных или самых продаваемых предметов в данном наборе.

Заключение

Код считывает данные о продажах из файла CSV, вычисляет общий доход, находит самый продаваемый продукт и тот, который принес наибольший доход. Затем он генерирует подробный отчет по каждому продукту, отображает этот отчет на экране и создает диаграммы (круговую и гистограмму) для визуализации данных о продажах.