Разработка многооконного приложения для управления интернет-магазином на C# Windows Forms

Функциональные требования с подсказками

1. **Главное окно (MainForm):**
    - **Меню с вкладками:**
        - **Подсказка:** Используйте элемент `MenuStrip` для создания меню. Добавьте пункты "Управление заказами", "Управление клиентами" и "Управление товарами". При нажатии на них открывайте соответствующие окна (`OrdersForm`, `CustomersForm`, `ProductsForm`) с помощью `ShowDialog()` или `Show()`.
    - **Кнопки для экспорта и импорта всех данных (JSON):**
        - **Подсказка:** Реализуйте кнопки "Экспорт" и "Импорт" в панели инструментов (`ToolStrip`). Для работы с JSON используйте библиотеку `System.Text.Json`. Создайте методы `ExportAllData()` и `ImportAllData()` в отдельном классе `DataManager`. Храните данные в коллекциях (`List<>`).

2. **Окно управления заказами (OrdersForm):**
    - **Таблица (`DataGridView`):**
        - **Подсказка:** Добавьте `DataGridView` для отображения заказов. Свяжите таблицу с `BindingList<Order>`, чтобы упростить обновление данных.
    - **Поля для добавления/редактирования заказа:**
        - **Подсказка:** Используйте `ComboBox` для выбора клиента и товара, заполняя их списки из заранее инициализированных коллекций (`List<Customer>` и `List<Product>`). Для ввода количества используйте `NumericUpDown`.
    - **Кнопки:**
        - **"Добавить заказ":** В методе обработчика проверяйте, заполнены ли все поля. После добавления обновляйте таблицу вызовом `dataGridView.Refresh()`.
        - **"Редактировать заказ":** При выборе заказа из таблицы автоматически заполняйте поля для редактирования. После изменения данных обновляйте таблицу.
        - **"Удалить заказ":** Удаляйте выбранный заказ и обновляйте данные в таблице.
        - **"Поиск заказа":** Реализуйте текстовое поле для ввода поискового запроса. Для поиска по ID или имени клиента выполните цикл по списку заказов.

- **Дополнительная подсказка:** Убедитесь, что редактирование и удаление активны только при выбранном заказе. Это можно настроить с помощью свойства `Enabled` у кнопок.

3. **Окно управления клиентами (CustomersForm):**
   - **Таблица для отображения клиентов:**
     - **Подсказка:** Аналогично окну заказов, используйте `DataGridView` и `BindingList<Customer>` для упрощения работы с данными.
   - **Поля для добавления/редактирования клиента:**
     - **Подсказка:** Для ввода телефона используйте элемент `MaskedTextBox` с маской, например `+7 (000) 000-00-00`. Для проверки корректности email можно использовать метод проверки строки на наличие символа `@`.
   - **Кнопки:**
     - **"Добавить клиента":** Добавляйте нового клиента в коллекцию и обновляйте таблицу.
     - **"Редактировать клиента":** После выбора строки из таблицы заполняйте поля редактирования. Убедитесь, что данные корректно сохраняются.
     - **"Удалить клиента":** Удаляйте клиента из коллекции и предупреждайте пользователя о возможном удалении связанных заказов.
     - **"Поиск клиента":** Реализуйте поиск по имени или телефону, выполняя цикл по списку клиентов.

4. **Окно управления товарами (ProductsForm):**
   - **Таблица для отображения товаров:**
     - **Подсказка:** Отобразите данные о товарах (например, название, цена, количество на складе) в `DataGridView`, связав его с `BindingList<Product>`.
   - **Поля для добавления/редактирования товара:**
     - **Подсказка:** Используйте `TextBox` для ввода названия товара, `NumericUpDown` для цены и количества на складе.
   - **Кнопки:**
     - **"Добавить товар":** Добавляйте товар в коллекцию с проверкой уникальности названия.
     - **"Редактировать товар":** Позволяйте изменять только цену и количество, но не название.
     - **"Удалить товар":** Удаляйте товар из коллекции и предупреждайте пользователя о возможном удалении связанных заказов.
     - **"Поиск товара":** Выполняйте поиск по названию или ID товара, используя цикл по списку товаров.

5. **Окно просмотра деталей заказа (OrderDetailsForm):**
    - **Отображение информации:**
        - **Подсказка:** Используйте элементы `Label` для отображения информации о клиенте, товаре и деталях заказа. Получите данные из объекта заказа, переданного в это окно через конструктор.
    - **Кнопка "Закрыть":**
        - **Подсказка:** Используйте метод `Close()` для закрытия окна.

Дополнительные подсказки

1. **Хранение данных:**
    - Используйте коллекции (`List<>`) для хранения данных о клиентах, товарах и заказах.
    - Пример классов:

```
public class Customer
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Phone { get; set; }
    public string Email { get; set; }
}

public class Product
{
    public int Id { get; set; }
    public string Name { get; set; }
    public decimal Price { get; set; }
    public int Stock { get; set; }
}

public class Order
{
    public int Id { get; set; }
    public Customer Customer { get; set; }
    public Product Product { get; set; }
    public int Quantity { get; set; }
    public decimal TotalPrice => Product.Price *
Quantity;
}
```

2. **Экспорт и импорт:**
    - Для экспорта и импорта данных используйте `System.Text.Json`.
    - Пример структуры JSON:

```
{
    "Customers": [
        { "Id": 1, "Name": "Иван Иванов", "Phone":
"+79991234567", "Email": "ivan@example.com" }
```

```
    ],
    "Products": [
        { "Id": 1, "Name": "Ноутбук", "Price": 50000,
"Stock": 5 }
    ],
    "Orders": [
        { "Id": 1, "CustomerId": 1, "ProductId": 1,
"Quantity": 2 }
    ]
}
```

3. **Логика поиска:**
   - Используйте простые циклы для поиска нужного элемента в коллекции, например:

```
var result = new List<Order>();
foreach (var order in orders)
{
    if (order.Customer.Name.Contains(searchQuery))
    {
        result.Add(order);
    }
}
```

Подсказки для Unit-тестов

1. **Среда для тестирования:**
   - Используйте NUnit или MSTest. Убедитесь, что вся бизнес-логика отделена от интерфейса (создайте классы `OrderManager`, `CustomerManager`, `ProductManager`).
2. **Пример методов для тестирования:**
   - `AddOrder`: Добавление заказа.
   - `RemoveOrder`: Удаление заказа.
   - `FindCustomerByName`: Поиск клиента по имени.
   - `EditProduct`: Редактирование товара.
3. **Пример теста:**

```
[Test]
public void AddOrder_ShouldAddOrderToList()
{
    var manager = new OrderManager();
    var customer = new Customer { Id = 1, Name = "Иван
Иванов" };
    var product = new Product { Id = 1, Name = "Ноутбук",
Price = 50000 };
    manager.AddOrder(customer, product, 1);
    Assert.AreEqual(1, manager.Orders.Count);
}
```