**Шпаргалка по Приложениям 1 и 2**

**Кратко, структурированно, с акцентами на ключевые моменты**

**📌 Приложение 1: Описание предметной области**

**🔹 Основные сущности системы:**

1. **Партнеры**
   * *Поля:* Тип, название компании, ИНН, ФИО директора, контакты, рейтинг.
   * *Особенности:*
     + Скидки зависят от объема продаж.
     + Взаиморасчеты только в рублях.
2. **Заявки**
   * *Жизненный цикл:*
     1. Создание → 2. Предоплата (3 дня) → 3. Производство → 4. Доставка → 5. Полная оплата.
   * *Важно:* Автоматическая отмена, если нет предоплаты.
3. **Материалы (склад)**
   * *Поля:* Тип, название, поставщик, количество, минимальный остаток, стоимость.
   * *Действия:* Резервирование, списание, контроль остатков.
4. **Продукция**
   * *Поля:* Артикул, тип, вес, себестоимость, материалы для производства.
   * *Расчеты:* Минимальная стоимость для партнера.
5. **Сотрудники**
   * *Доступ:* Турникеты по картам.
   * *Роли:* Менеджеры (работа с партнерами), мастера (контроль качества).

**🔹 Связи между сущностями:**

* Партнер ↔ Заявка (1:M).
* Материалы ↔ Продукция (M:M через таблицу использования).

**📌 Приложение 2: Руководство по стилю**

**🔹 Визуальные требования:**

1. **Цвета:**
   * Основной фон: #FFFFFF (белый).
   * Доп. фон: #BFD6F6 (голубой).
   * Акцент: #405C73 (темно-синий).
2. **Шрифт:**
   * *Constantia* (везде).
3. **Логотип и иконка:**
   * Не искажать пропорции/цвет.
   * Логотип — на главной форме.
4. **Элементы интерфейса:**
   * Все формы должны иметь заголовок.
   * Кнопки действий выделять акцентным цветом.

**🔹 Пример кнопки (HTML):**

html

Copy

Download

Run

<button style="background-color: #405C73; color: white; font-family: Constantia;">

Сохранить

</button>

**📌 Шпаргалка по интеграции данных**

**Как связать сущности из Приложения 1 в БД (SQL):**

sql

Copy

Download

-- Таблица "Партнеры"

CREATE TABLE Partners (

PartnerID INT PRIMARY KEY,

CompanyName VARCHAR(100) NOT NULL,

Rating INT DEFAULT 0 CHECK (Rating >= 0)

);

-- Таблица "Заявки"

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

PartnerID INT,

FOREIGN KEY (PartnerID) REFERENCES Partners(PartnerID)

);

-- Таблица "Материалы\_для\_продукции" (M:M)

CREATE TABLE ProductMaterials (

ProductID INT,

MaterialID INT,

Quantity DECIMAL(10, 2),

PRIMARY KEY (ProductID, MaterialID)

);

**📌 Экстренные подсказки**

1. **Если забыли связи между сущностями:**
   * Партнеры → Заявки → Продукция → Материалы.
2. **Для интерфейса (Flask):**

python

Copy

Download

@app.route('/orders')

def orders():

orders = db.execute("SELECT \* FROM Orders").fetchall()

return render\_template('orders.html', orders=orders, bg\_color="#BFD6F6")

1. **Проверка данных:**
   * У партнера должен быть рейтинг ≥ 0.
   * Цена продукции/материалов ≥ 0.

**🎯 Итог:**

* **Приложение 1** — основа для проектирования БД.
* **Приложение 2** — гайд по UI/UX.
* **Обязательно:** Соблюдать цветовую схему и шрифт!

**У вас всё получится!** 💙  
Если нужно углубиться в конкретный раздел — спрашивайте!

теперь нужно сделать код для того чтобы импортировать данные в программу. и сделай лучше под wpfapp всё шпаргалки.

**Шпаргалка для WPF-приложения (C#)**

**Импорт данных, стилизация по гайду и работа с БД**

**📌 1. Импорт данных (CSV/Excel в SQLite)**

**🔹 Класс для работы с данными:**

csharp

Copy

Download

using System.Data.SQLite;

using System.IO;

using System.Windows;

public class DataImporter

{

private string \_dbPath = "database.db";

private string \_connectionString;

public DataImporter()

{

\_connectionString = $"Data Source={\_dbPath};Version=3;";

InitializeDatabase();

}

// Создание БД и таблиц

private void InitializeDatabase()

{

if (!File.Exists(\_dbPath))

{

SQLiteConnection.CreateFile(\_dbPath);

using (var conn = new SQLiteConnection(\_connectionString))

{

conn.Open();

string sql = @"

CREATE TABLE Materials (

MaterialID INTEGER PRIMARY KEY,

Name TEXT NOT NULL,

Quantity REAL CHECK (Quantity >= 0)

);";

new SQLiteCommand(sql, conn).ExecuteNonQuery();

}

}

}

// Импорт из CSV

public void ImportFromCsv(string filePath)

{

try

{

var lines = File.ReadAllLines(filePath);

using (var conn = new SQLiteConnection(\_connectionString))

{

conn.Open();

foreach (var line in lines.Skip(1)) // Пропуск заголовка

{

var values = line.Split(',');

string sql = $"INSERT INTO Materials (Name, Quantity) VALUES ('{values[0]}', {values[1]})";

new SQLiteCommand(sql, conn).ExecuteNonQuery();

}

}

MessageBox.Show("Данные успешно импортированы!", "Успех");

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка: {ex.Message}", "Ошибка");

}

}

}

**🔹 Использование в WPF:**

xml

Copy

Download

Run

<!-- MainWindow.xaml -->

<Window x:Class="WpfApp.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Импорт данных" Height="300" Width="400">

<Grid Background="#BFD6F6">

<Button Content="Импорт CSV"

Click="ImportButton\_Click"

Background="#405C73"

Foreground="White"

FontFamily="Constantia"

Width="120" Height="40"/>

</Grid>

</Window>

csharp

Copy

Download

// MainWindow.xaml.cs

private void ImportButton\_Click(object sender, RoutedEventArgs e)

{

var openFileDialog = new Microsoft.Win32.OpenFileDialog

{

Filter = "CSV files (\*.csv)|\*.csv"

};

if (openFileDialog.ShowDialog() == true)

{

new DataImporter().ImportFromCsv(openFileDialog.FileName);

}

}

**📌 2. Стилизация по Приложению 2**

**🔹 Стандартные стили (App.xaml):**

xml

Copy

Download

Run

<Application.Resources>

<Style TargetType="Button">

<Setter Property="Background" Value="#405C73"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="FontFamily" Value="Constantia"/>

<Setter Property="Margin" Value="10"/>

</Style>

<Style TargetType="TextBox">

<Setter Property="Background" Value="#FFFFFF"/>

<Setter Property="FontFamily" Value="Constantia"/>

</Style>

</Application.Resources>

**🔹 Главное окно с логотипом:**

xml

Copy

Download

Run

<Window>

<Grid Background="#BFD6F6">

<!-- Логотип -->

<Image Source="logo.png" Width="100" Height="50" HorizontalAlignment="Left" Margin="10"/>

<!-- Контент -->

<StackPanel VerticalAlignment="Center">

<TextBlock Text="Список материалов" FontSize="16" HorizontalAlignment="Center"/>

<ListView x:Name="MaterialsList" Background="#FFFFFF">

<ListView.View>

<GridView>

<GridViewColumn Header="Название" DisplayMemberBinding="{Binding Name}"/>

<GridViewColumn Header="Количество" DisplayMemberBinding="{Binding Quantity}"/>

</GridView>

</ListView.View>

</ListView>

</StackPanel>

</Grid>

</Window>

**📌 3. Работа с БД (Entity Framework Core)**

**🔹 Установка пакетов:**

bash

Copy

Download

Install-Package Microsoft.EntityFrameworkCore.Sqlite

Install-Package Microsoft.EntityFrameworkCore.Tools

**🔹 Модель и контекст:**

csharp

Copy

Download

public class Material

{

public int MaterialID { get; set; }

public string Name { get; set; }

public double Quantity { get; set; }

}

public class AppDbContext : DbContext

{

public DbSet<Material> Materials { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder options)

=> options.UseSqlite("Data Source=database.db");

}

**🔹 Загрузка данных в ListView:**

csharp

Copy

Download

using (var db = new AppDbContext())

{

MaterialsList.ItemsSource = db.Materials.ToList();

}

**📌 4. Обработка ошибок**

**🔹 Валидация при импорте:**

csharp

Copy

Download

try

{

if (!File.Exists(filePath))

throw new FileNotFoundException("Файл не найден");

var lines = File.ReadAllLines(filePath);

if (lines.Length == 0)

throw new Exception("Файл пуст");

}

catch (Exception ex)

{

MessageBox.Show($"Проверьте файл: {ex.Message}", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

**📌 5. Шпаргалка по ключевым командам**

| **Действие** | **Код/Команда** |
| --- | --- |
| **Открытие файла** | OpenFileDialog + File.ReadAllLines |
| **Подключение БД** | SQLiteConnection или DbContext |
| **Обновление ListView** | ListView.ItemsSource = db.Materials.ToList() |
| **Стиль кнопки** | Background="#405C73" FontFamily="Constantia" |
| **Проверка файла** | File.Exists(path) |

**🎯 Итог:**

1. **Импорт данных:** Через CSV + SQLite.
2. **Стили:** Цвета #FFFFFF, #BFD6F6, #405C73, шрифт Constantia.
3. **БД:** Entity Framework Core для удобства.
4. **Интерфейс:** Логотип, ListView, валидация.

**Совет:** Для сложных импортов используйте CsvHelper (NuGet).  
**Важно:** Всегда проверяйте путь к файлу и формат данных!

Если нужно добавить конкретный функционал — пишите! 😊