

Лабораторная работа №5: Добавление новой функции в программу

Задание

1. Добавить новую функцию в приложение по теме варианта.
2. Реализовать модульные тесты для проверки корректности работы новой функции.
3. Реализовать тест-кейсы для проверки корректности работы новой функции.

Пример выполнения задания

Новое функциональное требование:

Редактирования товара:

- Пользователь должен иметь возможность выбрать товар из списка и отредактировать его данные (название, цена, количество).
- После редактирования данные товара должны обновляться в списке.

Реализация нового требования:

1. **Добавляется кнопка "Редактировать" на форму:**

В конструкторе формы MainWindow добавляется новая кнопка для редактирования товара

```
private Button button4; // Кнопка для редактирования товара
```

В методе InitializeComponent добавляется инициализация и настройка кнопки:

```
this.button4 = new System.Windows.Forms.Button();  
this.button4.Location = new System.Drawing.Point(374, 241);  
this.button4.Name = "button4";  
this.button4.Size = new System.Drawing.Size(75, 23);  
this.button4.TabIndex = 11;  
this.button4.Text = "Редактировать";  
this.button4.UseVisualStyleBackColor = true;  
this.button4.Click += new System.EventHandler(this.editButton_Click);  
this.Controls.Add(this.button4);
```

2. **Добавляется обработчик события для кнопки "Редактировать":**

В классе `MainWindow` добавляется метод `editButton_Click`, который будет обрабатывать нажатие на кнопку редактирования:

```
private void editButton_Click(object sender, EventArgs e)
{
    try
    {
        if (productsList.SelectedIndex >= 0)
        {
            // Получаем выбранный товар
            Product selectedProduct = products[productsList.SelectedIndex];

            // Обновляем данные товара из текстовых полей
            selectedProduct.Name = nameTextBox.Text;
            selectedProduct.Price = numericUpDown1.Value;
            selectedProduct.Quantity = (int)numericUpDown2.Value;

            // Обновляем список товаров
            UpdateProductList();
            MessageBox.Show("Товар отредактирован!", "Успех",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            MessageBox.Show("Выберите товар для редактирования!",
                "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Ошибка", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
    }
}
```

3. Добавляется возможность пред заполнения полей при выборе товара:

Чтобы упростить редактирование, решено добавить возможность пред заполнения полей данными выбранного товара при выборе его в списке. Для этого добавляется обработчик события `SelectedIndexChanged` для `productsList`:

```
private void productsList_SelectedIndexChanged(object sender, EventArgs e)
{
    if (productsList.SelectedIndex >= 0)
    {
        Product selectedProduct = products[productsList.SelectedIndex];
        nameTextBox.Text = selectedProduct.Name;
        numericUpDown1.Value = selectedProduct.Price;
        numericUpDown2.Value = selectedProduct.Quantity;
    }
}
```

Не забываем подписаться на это событие в методе `InitializeComponent`:

```
this.productsList.SelectedIndexChanged += new
System.EventHandler(this.productsList_SelectedIndexChanged);
```

Для проверки новой функции редактирования товара с использованием **MS Test** (фреймворка для модульного тестирования в .NET), создадим модульные тесты. Мы будем тестировать методы, связанные с редактированием товара, такие как обновление названия, цены и количества.

Пример модульных тестов для проверки корректности:

1. Тест для проверки обновления названия товара

```
using System.Reflection;
using System.Windows.Forms;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using ProductManager;
```

```
[TestClass]
public class ProductTests
{
    [TestMethod]
    public void EditProductName_ShouldUpdateName()
    {

```

```
// Arrange
var product = new Product("Тестовый товар", 100, 10);

// Act
product.Name = "Новое название";

// Assert
Assert.AreEqual("Новое название", product.Name);
}
}
```

2. Тест для проверки обновления цены товара

[TestMethod]

```
public void EditProductPrice_ShouldUpdatePrice()
{
    // Arrange
    var product = new Product("Тестовый товар", 100, 10);

    // Act
    product.Price = 200;

    // Assert
    Assert.AreEqual(200, product.Price);
}
```

3. Тест для проверки обновления количества товара

[TestMethod]

```
public void EditProductQuantity_ShouldUpdateQuantity()
{
    // Arrange
    var product = new Product("Тестовый товар", 100, 10);

    // Act
    product.Quantity = 20;

    // Assert
    Assert.AreEqual(20, product.Quantity);
}
```

```
}
```

4. Тест для проверки корректности метода ToString после редактирования

[TestMethod]

```
public void EditProduct_ToString_ShouldReturnCorrectString()
```

```
{
```

```
    // Arrange
```

```
    var product = new Product("Тестовый товар", 100, 10);
```

```
    // Act
```

```
    product.Name = "Новое название";
```

```
    product.Price = 200;
```

```
    product.Quantity = 5;
```

```
    // Assert
```

```
    Assert.AreEqual("Товар: Новое название, Цена: 200, Количество: 5",  
product.ToString());
```

```
}
```

5. Тест для проверки доступности товара после редактирования

[TestMethod]

```
public void EditProduct_IsAvailable_ShouldReturnCorrectStatus()
```

```
{
```

```
    // Arrange
```

```
    var product = new Product("Тестовый товар", 100, 10);
```

```
    // Act
```

```
    product.Quantity = 0;
```

```
    // Assert
```

```
    Assert.IsFalse(product.IsAvailable());
```

```
}
```

6. Тест для проверки корректности редактирования товара через UI (интеграционный тест)

Для этого теста потребуется эмуляция взаимодействия с пользовательским интерфейсом. Мы можем использовать MainWindow и проверить, что данные товара обновляются корректно.

Если поля и методы объекта MainWindow приватные, то для тестирования таких элементов можно использовать следующие подходы:

1. **Рефлексия (Reflection)**

Рефлексия позволяет получить доступ к приватным полям и методам. Это мощный инструмент, но его использование усложняет код и может привести к ошибкам, если структура класса изменится.

2. **Изменение модификаторов доступа**

Если возможно, измените модификаторы доступа с private на internal или protected, и используйте атрибут [InternalsVisibleTo], чтобы сделать их доступными для тестового проекта.

3. **Добавление публичных методов или свойств для тестирования**

Добавьте в класс MainWindow публичные методы или свойства, которые будут использоваться только для тестирования. Это может быть полезно, если вы не хотите изменять существующую логику.

4. **Интеграционные тесты через UI**

Используйте инструменты для автоматизации тестирования UI, такие как **WinAppDriver** или **UI Automation**, чтобы взаимодействовать с элементами управления через пользовательский интерфейс.

Пример использования рефлексии для тестирования приватных полей и методов

Допустим, у вас есть класс MainWindow с приватными полями и методами:

```
public partial class MainWindow : Form
{
    private List<Product> products = new List<Product>();
    private TextBox nameTextBox;
    private NumericUpDown numericUpDown1;
    private NumericUpDown numericUpDown2;
    private ListBox productsList;

    private void editButton_Click(object sender, EventArgs e)
    {
        // Логика редактирования товара
    }
}
```

Тест с использованием рефлексии

```
using Microsoft.VisualStudio.TestTools.UnitTesting;
using System;
using System.Collections.Generic;
using System.Reflection;
using System.Windows.Forms;

[TestClass]
public class MainWindowTests
{
    [TestMethod]
    public void EditProductInUI_ShouldUpdateProductList()
    {
        // Arrange
        var mainWindow = new MainWindow();

        // Используем рефлекссию для доступа к приватным полям
        var productsField = typeof(MainWindow).GetField("products",
BindingFlags.NonPublic | BindingFlags.Instance);
        var nameTextBoxField = typeof(MainWindow).GetField("nameTextBox",
BindingFlags.NonPublic | BindingFlags.Instance);
        var numericUpDown1Field = typeof(MainWindow).GetField("numericUpDown1",
BindingFlags.NonPublic | BindingFlags.Instance);
        var numericUpDown2Field = typeof(MainWindow).GetField("numericUpDown2",
BindingFlags.NonPublic | BindingFlags.Instance);
        var productsListField = typeof(MainWindow).GetField("productsList",
BindingFlags.NonPublic | BindingFlags.Instance);

        // Инициализация приватных полей
        var products = new List<Product>();
        productsField.SetValue(mainWindow, products);

        var nameTextBox = new TextBox();
        nameTextBoxField.SetValue(mainWindow, nameTextBox);

        var numericUpDown1 = new NumericUpDown();
        numericUpDown1Field.SetValue(mainWindow, numericUpDown1);

        var numericUpDown2 = new NumericUpDown();
        numericUpDown2Field.SetValue(mainWindow, numericUpDown2);

        var productsList = new ListBox();
```

```

productsListField.SetValue(mainWindow, productsList);

// Добавляем тестовый товар
var product = new Product("Тестовый товар", 100, 10);
products.Add(product);
productsList.Items.Add(product.ToString());

// Эмуляция выбора товара в UI
productsList.SelectedIndex = 0;

// Эмуляция ввода новых данных
nameTextBox.Text = "Новое название";
numericUpDown1.Value = 200;
numericUpDown2.Value = 5;

// Используем рефлексиию для вызова приватного метода
var editButtonClickMethod =
typeof(MainWindow).GetMethod("editButton_Click", BindingFlags.NonPublic |
BindingFlags.Instance);

// Act
editButtonClickMethod.Invoke(mainWindow, new object[] { null, null });

// Assert
Assert.AreEqual("Новое название", product.Name);
Assert.AreEqual(200, product.Price);
Assert.AreEqual(5, product.Quantity);
}
}

```

Пример использования InternalsVisibleTo

Если вы можете изменить модификаторы доступа, сделайте поля и методы `internal`, а затем добавьте атрибут `[InternalsVisibleTo]` в файл `AssemblyInfo.cs` вашего основного проекта:

```
using System.Runtime.CompilerServices;
```

```
[assembly: InternalsVisibleTo("YourTestProjectName")]
```

Теперь в тестовом проекте вы сможете получить доступ к `internal` полям и методам.

Итог

- **Рефлексия** — мощный, но сложный инструмент для тестирования приватных элементов.
- **InternalsVisibleTo** — более чистый способ, если вы можете изменить модификаторы доступа.
- **Публичные методы для тестирования** — простой и понятный способ, но требует изменения основного кода.

Выберите подход, который лучше всего подходит для вашего проекта.

Примеры тест-кейсов для проверки корректности работы:

Тест-кейс 1: Редактирование названия товара

- **Идентификатор тест-кейса:** TC_EDIT_001
 - **Название тест-кейса:** Редактирование названия товара
 - **Цель тестирования:** Проверить, что название товара корректно обновляется после редактирования.
 - **Предусловия:**
 - В списке товаров есть хотя бы один товар.
 - Пользователь выбрал товар из списка.
 - **Шаги выполнения:**
 - Выберите товар из списка.
 - Измените значение в поле "Наименование товара" на новое название.
 - Нажмите кнопку "Редактировать".
 - **Ожидаемый результат:**
 - Название товара в списке обновляется на новое значение.
 - Появляется сообщение "Товар отредактирован!".
 - **Постусловия:** Данные товара сохранены в списке.
 - **Приоритет:** Высокий
-

Тест-кейс 2: Редактирование цены товара

- **Идентификатор тест-кейса:** TC_EDIT_002
- **Название тест-кейса:** Редактирование цены товара
- **Цель тестирования:** Проверить, что цена товара корректно обновляется после редактирования.
- **Предусловия:**
 - В списке товаров есть хотя бы один товар.
 - Пользователь выбрал товар из списка.
- **Шаги выполнения:**

- Выберите товар из списка.
 - Измените значение в поле "Цена, руб." на новое значение.
 - Нажмите кнопку "Редактировать".
 - **Ожидаемый результат:**
 - Цена товара в списке обновляется на новое значение.
 - Появляется сообщение "Товар отредактирован!".
 - **Постусловия:** Данные товара сохранены в списке.
 - **Приоритет:** Высокий
-

Тест-кейс 3: Редактирование количества товара

- **Идентификатор тест-кейса:** TC_EDIT_003
 - **Название тест-кейса:** Редактирование количества товара
 - **Цель тестирования:** Проверить, что количество товара корректно обновляется после редактирования.
 - **Предусловия:**
 - В списке товаров есть хотя бы один товар.
 - Пользователь выбрал товар из списка.
 - **Шаги выполнения:**
 - Выберите товар из списка.
 - Измените значение в поле "Количество" на новое значение.
 - Нажмите кнопку "Редактировать".
 - **Ожидаемый результат:**
 - Количество товара в списке обновляется на новое значение.
 - Появляется сообщение "Товар отредактирован!".
 - **Постусловия:** Данные товара сохранены в списке.
 - **Приоритет:** Высокий
-

Тест-кейс 4: Редактирование товара без выбора

- **Идентификатор тест-кейса:** TC_EDIT_004
- **Название тест-кейса:** Редактирование товара без выбора
- **Цель тестирования:** Проверить, что система корректно обрабатывает попытку редактирования без выбора товара.
- **Предусловия:**
 - В списке товаров есть хотя бы один товар.
 - Пользователь не выбрал товар из списка.
- **Шаги выполнения:**
 - Не выбирайте товар из списка.
 - Нажмите кнопку "Редактировать".
- **Ожидаемый результат:**

- Появляется сообщение "Выберите товар для редактирования!".
 - **Постусловия:** Данные товара не изменены.
 - **Приоритет:** Средний
-

Тест-кейс 5: Редактирование товара с некорректной ценой

- **Идентификатор тест-кейса:** TC_EDIT_005
 - **Название тест-кейса:** Редактирование товара с отрицательной ценой
 - **Цель тестирования:** Проверить, что система корректно обрабатывает попытку установки отрицательной цены.
 - **Предусловия:**
 - В списке товаров есть хотя бы один товар.
 - Пользователь выбрал товар из списка.
 - **Шаги выполнения:**
 - Выберите товар из списка.
 - Измените значение в поле "Цена, руб." на отрицательное число (например, -10).
 - Нажмите кнопку "Редактировать".
 - **Ожидаемый результат:**
 - Появляется сообщение "Цена должна быть положительной!".
 - Данные товара не изменяются.
 - **Постусловия:** Данные товара не изменены.
 - **Приоритет:** Средний
-

Тест-кейс 6: Редактирование товара с нулевым количеством

- **Идентификатор тест-кейса:** TC_EDIT_006
- **Название тест-кейса:** Редактирование товара с нулевым количеством
- **Цель тестирования:** Проверить, что система корректно обрабатывает установку нулевого количества товара.
- **Предусловия:**
 - В списке товаров есть хотя бы один товар.
 - Пользователь выбрал товар из списка.
- **Шаги выполнения:**
 - Выберите товар из списка.
 - Измените значение в поле "Количество" на 0.
 - Нажмите кнопку "Редактировать".
- **Ожидаемый результат:**
 - Количество товара в списке обновляется на 0.
 - Появляется сообщение "Товар отредактирован!".
- **Постусловия:** Данные товара сохранены в списке.

- **Приоритет:** Средний
-

Тест-кейс 7: Редактирование товара с отрицательным количеством

- **Идентификатор тест-кейса:** TC_EDIT_007
 - **Название тест-кейса:** Редактирование товара с отрицательным количеством
 - **Цель тестирования:** Проверить, что система корректно обрабатывает попытку установки отрицательного количества товара.
 - **Предусловия:**
 - В списке товаров есть хотя бы один товар.
 - Пользователь выбрал товар из списка.
 - **Шаги выполнения:**
 - Выберите товар из списка.
 - Измените значение в поле "Количество" на отрицательное число (например, -5).
 - Нажмите кнопку "Редактировать".
 - **Ожидаемый результат:**
 - Появляется сообщение "Количество не может быть отрицательным!".
 - Данные товара не изменяются.
 - **Постусловия:** Данные товара не изменены.
 - **Приоритет:** Средний
-

Тест-кейс 8: Редактирование товара с пустым названием

- **Идентификатор тест-кейса:** TC_EDIT_008
- **Название тест-кейса:** Редактирование товара с пустым названием
- **Цель тестирования:** Проверить, что система корректно обрабатывает попытку установки пустого названия товара.
- **Предусловия:**
 - В списке товаров есть хотя бы один товар.
 - Пользователь выбрал товар из списка.
- **Шаги выполнения:**
 - Выберите товар из списка.
 - Очистите поле "Наименование товара" (оставьте его пустым).
 - Нажмите кнопку "Редактировать".
- **Ожидаемый результат:**
 - Появляется сообщение "Наименование товара не может быть пустым!" (если добавлена валидация).
 - Данные товара не изменяются.

- **Постусловия:** Данные товара не изменены.
- **Приоритет:** Средний

Вопросы для самоконтроля

1. Какие подходы можно использовать для тестирования приватных методов и полей?
2. Как обеспечить валидацию данных при редактировании товара?
3. Какие преимущества и недостатки у использования рефлексии в тестах?

Варианты

1. Управление банковским счётом

В программу, которая уже реализует управление банковским счётом, можно добавить новую функцию **"История операций"**. Эта функция позволит пользователю просматривать историю всех операций по счёту, таких как создание счёта, пополнение и снятие средств. Это сделает программу более удобной и прозрачной для пользователя.

Описание новой функции: История операций

Функция будет включать следующие возможности:

1. Запись всех операций по счёту (создание счёта, пополнение, снятие).
 2. Отображение истории операций в отдельном окне или на форме.
 3. Возможность очистки истории операций.
-

Функциональные требования

1. Запись операций

- Каждая операция (создание счёта, пополнение, снятие) должна записываться в историю.
- Запись должна содержать:
 - Тип операции (например, "Создание счёта", "Пополнение", "Снятие").
 - Сумму операции (если применимо).
 - Дату и время операции.

2. Отображение истории операций

- Система должна предоставлять возможность просмотра истории операций.
- История должна отображаться в виде списка или таблицы.

3. Очистка истории операций

- Система должна позволять очищать историю операций.

2. Конвертер валют

В программу "Конвертер валют" можно добавить новую функцию **"Автоматическое обновление курсов валют"**. Эта функция позволяет программе автоматически получать актуальные курсы валют из внешнего источника (например, API) и использовать их для конвертации. Это сделает программу более полезной и актуальной, так как курсы валют постоянно меняются.

Описание новой функции: Автоматическое обновление курсов валют

Функция будет включать следующие возможности:

1. Получение актуальных курсов валют из внешнего API.
 2. Обновление курсов валют в программе через определённые интервалы времени (например, каждые 10 минут).
 3. Отображение текущих курсов валют на форме.
-

Функциональные требования

1. Получение курсов валют

- Программа должна получать актуальные курсы валют (например, USD/EUR) из внешнего API.
- Курсы должны обновляться через определённые интервалы времени.

2. Отображение текущих курсов

- На форме должна отображаться информация о текущих курсах валют.

3. Обработка ошибок

- Если получение курсов валют из API не удалось, программа должна использовать последние доступные курсы и уведомить пользователя об ошибке.

3. Управление задачами

В программу "Управление задачами" можно добавить новую функцию **"Категории задач"**. Эта функция позволит пользователю группировать задачи по категориям, что делает управление задачами более удобным и организованным. Например, можно создать категории "Работа", "Дом", "Учёба" и т.д.

Описание новой функции: Категории задач

Функция будет включать следующие возможности:

1. Создание и управление категориями задач.
 2. Назначение задач определённым категориям.
 3. Фильтрация задач по категориям.
 4. Сохранение и загрузка категорий вместе с задачами.
-

Функциональные требования

1. Создание категорий

- Пользователь должен иметь возможность создавать новые категории.
- Каждая категория должна иметь уникальное имя.

2. Назначение задач категориям

- При добавлении задачи пользователь должен иметь возможность выбрать категорию.
- Если категория не выбрана, задача будет относиться к категории "Без категории".

3. Фильтрация задач по категориям

- Пользователь должен иметь возможность фильтровать задачи по выбранной категории.

4. Сохранение и загрузка категорий

- Категории должны сохраняться вместе с задачами в файл.
- При запуске программы категории и задачи должны загружаться из файла.

4. Календарь событий

В программу "Календарь событий" можно добавить новую функцию **"Напоминания о событиях"**. Эта функция позволит пользователю устанавливать напоминания о предстоящих событиях, чтобы не пропустить

важные даты. Напоминания могут быть реализованы через уведомления на экране или звуковые сигналы.

Описание новой функции: Напоминания о событиях

Функция будет включать следующие возможности:

1. Установка напоминаний для событий (например, за 1 день, за 1 час или за 10 минут до события).
 2. Отображение уведомлений о предстоящих событиях.
 3. Возможность отключения или изменения напоминаний.
-

Функциональные требования

1. Установка напоминаний

- Пользователь должен иметь возможность устанавливать напоминания для каждого события.
- Напоминания могут быть установлены на определённое время до события (например, за 1 день, за 1 час, за 10 минут).

2. Отображение уведомлений

- Программа должна отображать уведомления о предстоящих событиях в установленное время.
- Уведомления могут быть в виде всплывающих окон или звуковых сигналов.

3. Управление напоминаниями

- Пользователь должен иметь возможность отключать или изменять напоминания для событий.

5. Управление книгами в библиотеке

В программу "Управление книгами" можно добавить новую функцию **"Категории книг"**. Эта функция позволит пользователю группировать книги по категориям, что делает управление библиотекой более удобным и организованным. Например, можно создать категории "Художественная литература", "Научная литература", "Техническая литература" и т.д.

Описание новой функции: Категории книг

Функция будет включать следующие возможности:

1. Создание и управление категориями книг.

2. Назначение книг определённым категориям.
 3. Фильтрация книг по категориям.
 4. Сохранение и загрузка категорий вместе с книгами.
-

Функциональные требования

1. Создание категорий

- Пользователь должен иметь возможность создавать новые категории.
- Каждая категория должна иметь уникальное имя.

2. Назначение книг категориям

- При добавлении книги пользователь должен иметь возможность выбрать категорию.
- Если категория не выбрана, книга будет относиться к категории "Без категории".

3. Фильтрация книг по категориям

- Пользователь должен иметь возможность фильтровать книги по выбранной категории.

4. Сохранение и загрузка категорий

- Категории должны сохраняться вместе с книгами в файл.
- При запуске программы категории и книги должны загружаться из файла.

6. Управление заметками

В программу "Управление заметками" можно добавить новую функцию **"Теги для заметок"**. Эта функция позволит пользователю добавлять теги к заметкам, что сделает их организацию и поиск более удобными. Например, можно добавлять теги "Работа", "Личное", "Идеи" и т.д.

Описание новой функции: Теги для заметок

Функция будет включать следующие возможности:

1. Добавление тегов к заметкам.
 2. Фильтрация заметок по тегам.
 3. Управление тегами (создание, удаление, редактирование).
 4. Сохранение и загрузка тегов вместе с заметками.
-

Функциональные требования

1. Добавление тегов

- Пользователь должен иметь возможность добавлять теги к заметкам.
- Каждая заметка может иметь несколько тегов.

2. Фильтрация заметок по тегам

- Пользователь должен иметь возможность фильтровать заметки по выбранному тегу.

3. Управление тегами

- Пользователь должен иметь возможность создавать, удалять и редактировать теги.

4. Сохранение и загрузка тегов

- Теги должны сохраняться вместе с заметками в файл.
- При запуске программы теги и заметки должны загружаться из файла.

7. Управление контактами

В программу "Управление контактами" можно добавить новую функцию "Группы контактов". Эта функция позволит пользователю группировать контакты по категориям, что сделает управление контактами более удобным и организованным. Например, можно создать группы "Семья", "Друзья", "Коллеги" и т.д.

Описание новой функции: Группы контактов

Функция будет включать следующие возможности:

1. Создание и управление группами контактов.
 2. Назначение контактов определённым группам.
 3. Фильтрация контактов по группам.
 4. Сохранение и загрузка групп вместе с контактами.
-

Функциональные требования

1. Создание групп

- Пользователь должен иметь возможность создавать новые группы.
- Каждая группа должна иметь уникальное имя.

2. Назначение контактов группам

- При добавлении контакта пользователь должен иметь возможность выбрать группу.
- Если группа не выбрана, контакт будет относиться к группе "Без группы".

3. Фильтрация контактов по группам

- Пользователь должен иметь возможность фильтровать контакты по выбранной группе.

4. Сохранение и загрузка групп

- Группы должны сохраняться вместе с контактами в файл.
- При запуске программы группы и контакты должны загружаться из файла.

8. Управление покупками

В программу "Управление покупками" можно добавить новую функцию **"Анализ расходов"**. Эта функция позволит пользователю анализировать свои расходы по категориям, периодам и другим параметрам. Это поможет лучше понимать, на что тратятся деньги, и принимать более обоснованные финансовые решения.

Описание новой функции: Анализ расходов

Функция будет включать следующие возможности:

1. **Статистика по категориям:**
 - Пользователь сможет видеть, сколько денег было потрачено в каждой категории (например, "Продукты", "Техника", "Одежда").
 2. **Статистика по периодам:**
 - Пользователь сможет анализировать расходы за определённый период (например, за месяц, неделю или год).
 3. **Визуализация данных:**
 - Данные будут представлены в виде графиков или диаграмм для удобства восприятия.
 4. **Экспорт данных:**
 - Пользователь сможет экспортировать данные в файл (например, CSV или Excel).
-

Функциональные требования

1. Статистика по категориям

- Программа должна рассчитывать общую сумму расходов для каждой категории.
- Пользователь должен иметь возможность просматривать статистику по категориям.

2. Статистика по периодам

- Программа должна рассчитывать общую сумму расходов за выбранный период.
- Пользователь должен иметь возможность выбирать период (например, месяц, неделя, год).

3. Визуализация данных

- Данные должны отображаться в виде графиков или диаграмм (например, круговая диаграмма для категорий, линейный график для периодов).

4. Экспорт данных

- Пользователь должен иметь возможность экспортировать данные в файл (например, CSV или Excel).

9. Управление задачами с приоритетом

В программу "Управление задачами с приоритетом" можно добавить новую функцию **"Уведомления о сроках выполнения задач"**. Эта функция позволит пользователю получать уведомления о приближающихся сроках выполнения задач, чтобы не пропустить важные задачи. Уведомления могут быть реализованы через всплывающие окна или звуковые сигналы.

Описание новой функции: Уведомления о сроках выполнения задач

Функция будет включать следующие возможности:

1. Настройка уведомлений:

- Пользователь сможет настроить, за сколько времени до дедлайна он хочет получать уведомления (например, за 1 день, за 1 час, за 10 минут).

2. Отображение уведомлений:

- Программа будет отображать уведомления о приближающихся сроках выполнения задач.

3. Повторяющиеся уведомления:

- Уведомления будут повторяться до тех пор, пока задача не будет отмечена как выполненная или пока пользователь не закроет уведомление.
-

Функциональные требования

1. Настройка уведомлений

- Пользователь должен иметь возможность настраивать время уведомления (например, за 1 день, за 1 час, за 10 минут до дедлайна).

2. Отображение уведомлений

- Программа должна отображать уведомления о приближающихся сроках выполнения задач.
- Уведомления должны содержать описание задачи, её приоритет и оставшееся время до дедлайна.

3. Повторяющиеся уведомления

- Уведомления должны повторяться до тех пор, пока задача не будет отмечена как выполненная или пока пользователь не закроет уведомление.

10. Управление продажами

В программу "Управление продажами" можно добавить новую функцию **"Анализ продаж по периодам"**. Эта функция позволит пользователю анализировать продажи за определённые периоды времени (например, за день, неделю, месяц или год). Это поможет лучше понимать динамику продаж и принимать более обоснованные бизнес-решения.

Описание новой функции: Анализ продаж по периодам

Функция будет включать следующие возможности:

- 1. Анализ продаж за день:**
 - Пользователь сможет видеть общий доход и количество проданных товаров за выбранный день.
- 2. Анализ продаж за неделю:**

- Пользователь сможет видеть общий доход и количество проданных товаров за выбранную неделю.
 - 3. **Анализ продаж за месяц:**
 - Пользователь сможет видеть общий доход и количество проданных товаров за выбранный месяц.
 - 4. **Анализ продаж за год:**
 - Пользователь сможет видеть общий доход и количество проданных товаров за выбранный год.
 - 5. **Визуализация данных:**
 - Данные будут представлены в виде графиков или диаграмм для удобства восприятия.
-

Функциональные требования

1. Анализ продаж за день

- Программа должна рассчитывать общий доход и количество проданных товаров за выбранный день.

2. Анализ продаж за неделю

- Программа должна рассчитывать общий доход и количество проданных товаров за выбранную неделю.

3. Анализ продаж за месяц

- Программа должна рассчитывать общий доход и количество проданных товаров за выбранный месяц.

4. Анализ продаж за год

- Программа должна рассчитывать общий доход и количество проданных товаров за выбранный год.

5. Визуализация данных

- Данные должны отображаться в виде графиков или диаграмм (например, линейный график для доходов, столбчатая диаграмма для количества проданных товаров).

11. Управление инвентарём

В программу "Управление инвентарём" можно добавить новую функцию **"Автоматическое уведомление о низком уровне запасов"**. Эта функция позволит пользователю получать уведомления, когда количество определённого товара становится ниже заданного порога. Это поможет

избежать ситуаций, когда товар заканчивается на складе, и своевременно пополнять запасы.

Описание новой функции: Уведомления о низком уровне запасов

Функция будет включать следующие возможности:

1. **Настройка порога уведомлений:**
 - Пользователь сможет задать минимальное количество товара, при котором будет срабатывать уведомление.
 2. **Отображение уведомлений:**
 - Программа будет отображать уведомления, если количество товара становится ниже заданного порога.
 3. **Повторяющиеся уведомления:**
 - Уведомления будут повторяться до тех пор, пока количество товара не будет увеличено выше порога.
-

Функциональные требования

1. Настройка порога уведомлений

- Пользователь должен иметь возможность задать минимальное количество товара для уведомления.

2. Отображение уведомлений

- Программа должна отображать уведомления, если количество товара становится ниже заданного порога.
- Уведомления должны содержать название товара и текущее количество.

3. Повторяющиеся уведомления

- Уведомления должны повторяться до тех пор, пока количество товара не будет увеличено выше порога.

12. Управление заказами

В программу "Управление заказами" можно добавить новую функцию **"Уведомления о статусе заказов"**. Эта функция позволит пользователю получать уведомления о важных изменениях в статусе заказов, таких как переход заказа в статус "Завершён" или "В обработке". Это поможет своевременно реагировать на изменения и улучшить управление заказами.

Описание новой функции: Уведомления о статусе заказов

Функция будет включать следующие возможности:

1. **Уведомления о завершении заказа:**
 - Пользователь будет получать уведомление, когда заказ переходит в статус "Завершён".
 2. **Уведомления о начале обработки заказа:**
 - Пользователь будет получать уведомление, когда заказ переходит в статус "В обработке".
 3. **Настройка уведомлений:**
 - Пользователь сможет настроить, для каких статусов он хочет получать уведомления.
-

Функциональные требования

1. Уведомления о завершении заказа

- Программа должна отображать уведомление, когда заказ переходит в статус "Завершён".

2. Уведомления о начале обработки заказа

- Программа должна отображать уведомление, когда заказ переходит в статус "В обработке".

3. Настройка уведомлений

- Пользователь должен иметь возможность включать или отключать уведомления для определённых статусов.

13. Управление сотрудниками

В программу "Управление сотрудниками" можно добавить новую функцию **"Уведомления о предстоящих отпусках"**. Эта функция позволит пользователю получать уведомления о приближающихся отпусках сотрудников, чтобы своевременно планировать рабочие процессы и замещения.

Описание новой функции: Уведомления о предстоящих отпусках

Функция будет включать следующие возможности:

1. **Уведомления о начале отпуска:**

- Пользователь будет получать уведомление за определённое время до начала отпуска сотрудника (например, за 3 дня).
 - 2. **Уведомления о завершении отпуска:**
 - Пользователь будет получать уведомление за определённое время до завершения отпуска сотрудника (например, за 1 день).
 - 3. **Настройка уведомлений:**
 - Пользователь сможет настраивать время уведомлений (например, за сколько дней до начала или завершения отпуска).
-

Функциональные требования

1. Уведомления о начале отпуска

- Программа должна отображать уведомление за определённое время до начала отпуска сотрудника.

2. Уведомления о завершении отпуска

- Программа должна отображать уведомление за определённое время до завершения отпуска сотрудника.

3. Настройка уведомлений

- Пользователь должен иметь возможность настраивать время уведомлений (например, за 3 дня до начала отпуска и за 1 день до завершения).

14. Управление проектами

В программу "Управление проектами" можно добавить новую функцию **"Уведомления о сроках выполнения проектов"**. Эта функция позволит пользователю получать уведомления о приближающихся сроках завершения проектов, чтобы своевременно принимать меры для их завершения.

Описание новой функции: Уведомления о сроках выполнения проектов

Функция будет включать следующие возможности:

1. **Уведомления о приближении срока завершения проекта:**
 - Пользователь будет получать уведомление за определённое время до окончания срока проекта (например, за 7 дней).
2. **Уведомления о просроченных проектах:**

- Пользователь будет получать уведомление, если проект не завершён к установленному сроку.

3. Настройка уведомлений:

- Пользователь сможет настраивать время уведомлений (например, за сколько дней до окончания срока).
-

Функциональные требования

1. Уведомления о приближении срока завершения проекта

- Программа должна отображать уведомление за определённое время до окончания срока проекта.

2. Уведомления о просроченных проектах

- Программа должна отображать уведомление, если проект не завершён к установленному сроку.

3. Настройка уведомлений

- Пользователь должен иметь возможность настраивать время уведомлений (например, за 7 дней до окончания срока).

15. Управление клиентами

В программу "Управление клиентами" можно добавить новую функцию **"Группировка клиентов по категориям"**. Эта функция позволит пользователю группировать клиентов по определённым категориям, например, "Постоянные клиенты", "Потенциальные клиенты", "VIP-клиенты" и т.д. Это упростит управление клиентской базой и улучшит её организацию.

Описание новой функции: Группировка клиентов по категориям

Функция будет включать следующие возможности:

1. Создание и управление категориями клиентов:

- Пользователь сможет создавать новые категории и назначать клиентов в эти категории.

2. Фильтрация клиентов по категориям:

- Пользователь сможет фильтровать клиентов по выбранной категории.

3. Сохранение и загрузка категорий:

- Категории будут сохраняться вместе с клиентами и загружаться при запуске программы.
-

Функциональные требования

1. Создание и управление категориями

- Пользователь должен иметь возможность создавать новые категории и назначать клиентов в эти категории.

2. Фильтрация клиентов по категориям

- Пользователь должен иметь возможность фильтровать клиентов по выбранной категории.

3. Сохранение и загрузка категорий

- Категории должны сохраняться вместе с клиентами в файл.
- При запуске программы категории и клиенты должны загружаться из файла.

16. Управление доставкой

В вашу программу "Управление доставками" можно добавить новую функцию **"Группировка доставок по статусам"**. Эта функция позволит пользователю группировать доставки по их текущему статусу (например, "Новый", "В пути", "Доставлен"). Это упростит управление доставками и улучшит их организацию.

Описание новой функции: Группировка доставок по статусам

Функция будет включать следующие возможности:

1. Фильтрация доставок по статусам:

- Пользователь сможет фильтровать доставки по выбранному статусу (например, показать только "Новые" или "Доставленные").

2. Сохранение и загрузка данных:

- Статусы доставок будут сохраняться вместе с остальными данными и загружаться при запуске программы.
-

Функциональные требования

1. Фильтрация доставок по статусам:

- Пользователь должен иметь возможность фильтровать доставки по выбранному статусу.
- 2. **Сохранение и загрузка данных:**
 - Статусы доставок должны сохраняться в файл и загружаться при запуске программы.

17. Управление отчётами

В вашу программу "Управление отчётами" можно добавить новую функцию **"Категоризация отчётов"**. Эта функция позволит пользователю группировать отчёты по категориям, что упростит их организацию и поиск. Например, можно создать категории "Финансовые отчёты", "Технические отчёты", "Маркетинговые отчёты" и т.д.

Описание новой функции: Категоризация отчётов

Функция будет включать следующие возможности:

1. **Создание и управление категориями:**
 - Пользователь сможет создавать новые категории и назначать отчёты в эти категории.
 2. **Фильтрация отчётов по категориям:**
 - Пользователь сможет фильтровать отчёты по выбранной категории.
 3. **Сохранение и загрузка категорий:**
 - Категории будут сохраняться вместе с отчётами и загружаться при запуске программы.
-

Функциональные требования

1. **Создание и управление категориями:**
 - Пользователь должен иметь возможность создавать новые категории и назначать отчёты в эти категории.
2. **Фильтрация отчётов по категориям:**
 - Пользователь должен иметь возможность фильтровать отчёты по выбранной категории.
3. **Сохранение и загрузка категорий:**
 - Категории должны сохраняться вместе с отчётами в файл и загружаться при запуске программы.

18. Управление резервированием

В вашу программу "Управление резервированием" можно добавить новую функцию "**Проверка доступности времени для резервирования**". Эта функция позволит пользователю проверять, свободно ли выбранное время для нового резервирования, чтобы избежать наложения бронирований.

Описание новой функции: Проверка доступности времени

Функция будет включать следующие возможности:

1. **Проверка наложения времени:**
 - При добавлении нового резервирования программа будет проверять, не пересекается ли выбранное время с уже существующими резервированиями.
 2. **Уведомление о конфликте:**
 - Если выбранное время занято, программа покажет сообщение с информацией о конфликте.
 3. **Отображение доступного времени:**
 - Пользователь сможет просматривать список свободных временных интервалов для выбора.
-

Функциональные требования

1. **Проверка наложения времени:**
 - Программа должна проверять, не пересекается ли новое резервирование с уже существующими.
2. **Уведомление о конфликте:**
 - Если время занято, программа должна уведомить пользователя.
3. **Отображение доступного времени:**
 - Пользователь должен видеть список свободных временных интервалов.

19. Управление бюджетом

В вашу программу "Управление бюджетом" можно добавить новую функцию "**Категоризация транзакций**". Эта функция позволит пользователю группировать транзакции по категориям, что упростит анализ расходов и доходов. Например, можно создать категории "Продукты", "Транспорт", "Зарплата", "Развлечения" и т.д.

Описание новой функции: Категоризация транзакций

Функция будет включать следующие возможности:

1. **Создание и управление категориями:**
 - Пользователь сможет создавать новые категории и назначать транзакции в эти категории.
 2. **Фильтрация транзакций по категориям:**
 - Пользователь сможет фильтровать транзакции по выбранной категории.
 3. **Сохранение и загрузка категорий:**
 - Категории будут сохраняться вместе с транзакциями и загружаться при запуске программы.
 4. **Анализ расходов и доходов по категориям:**
 - Пользователь сможет просматривать статистику по каждой категории (например, общая сумма расходов или доходов).
-

Функциональные требования

1. **Создание и управление категориями:**
 - Пользователь должен иметь возможность создавать новые категории и назначать транзакции в эти категории.
2. **Фильтрация транзакций по категориям:**
 - Пользователь должен иметь возможность фильтровать транзакции по выбранной категории.
3. **Сохранение и загрузка категорий:**
 - Категории должны сохраняться вместе с транзакциями в файл и загружаться при запуске программы.
4. **Анализ расходов и доходов по категориям:**
 - Пользователь должен видеть статистику по каждой категории (например, общая сумма расходов или доходов).

20. Управление файлами и папками

В вашу программу "Управление файлами и папками" можно добавить новую функцию **"Поиск файлов и папок"**. Эта функция позволит пользователю искать файлы и папки по имени или части имени, что упростит навигацию в больших директориях.

Описание новой функции: Поиск файлов и папок

Функция будет включать следующие возможности:

1. **Поиск по имени:**
 - Пользователь сможет вводить имя или часть имени файла или папки для поиска.
 2. **Отображение результатов поиска:**
 - Результаты поиска будут отображаться в `ListView`, как и при обычном просмотре файлов.
 3. **Сброс поиска:**
 - Пользователь сможет сбросить результаты поиска и вернуться к полному списку файлов и папок.
-

Функциональные требования

1. **Поиск по имени:**
 - Пользователь должен иметь возможность вводить имя или часть имени для поиска.
2. **Отображение результатов поиска:**
 - Результаты поиска должны отображаться в `ListView`.
3. **Сброс поиска:**
 - Пользователь должен иметь возможность сбросить результаты поиска и вернуться к полному списку файлов и папок.

21. Управление системой безопасности данных

В вашу программу "Управление безопасностью данных" можно добавить новую функцию "**Управление ключами шифрования**". Эта функция позволит пользователю создавать, сохранять и загружать ключи шифрования, что упростит процесс шифрования и расшифровки файлов, особенно при работе с большим количеством файлов или при необходимости обмена зашифрованными данными с другими пользователями.

Описание новой функции: Управление ключами шифрования

Функция будет включать следующие возможности:

1. **Генерация ключей шифрования:**
 - Пользователь сможет генерировать новые ключи шифрования и сохранять их в файл.
2. **Загрузка ключей шифрования:**

- Пользователь сможет загружать ранее сохранённые ключи для шифрования и расшифровки файлов.
 - 3. **Экспорт и импорт ключей:**
 - Пользователь сможет экспортировать ключи для передачи другим пользователям и импортировать ключи, полученные от других пользователей.
 - 4. **Хранение ключей в защищённом хранилище:**
 - Ключи будут храниться в зашифрованном виде для повышения безопасности.
-

Функциональные требования

1. **Генерация ключей:**
 - Пользователь должен иметь возможность генерировать новые ключи шифрования.
2. **Сохранение и загрузка ключей:**
 - Пользователь должен иметь возможность сохранять ключи в файл и загружать их из файла.
3. **Экспорт и импорт ключей:**
 - Пользователь должен иметь возможность экспортировать и импортировать ключи для обмена с другими пользователями.
4. **Защита ключей:**
 - Ключи должны храниться в зашифрованном виде для предотвращения несанкционированного доступа.

22. Управление здоровьем

В вашу программу "Управление здоровьем" можно добавить новую функцию **"Цели и рекомендации"**. Эта функция позволит пользователю устанавливать цели по активности, питанию и сну, а также получать рекомендации на основе текущих данных. Например, программа может рекомендовать увеличить продолжительность сна, если пользователь спит меньше нормы, или предложить добавить больше физической активности.

Описание новой функции: Цели и рекомендации

Функция будет включать следующие возможности:

1. **Установка целей:**

- Пользователь сможет устанавливать цели по активности (например, 30 минут в день), питанию (например, 2000 калорий в день) и сну (например, 8 часов в день).
 - 2. **Отслеживание прогресса:**
 - Программа будет сравнивать текущие данные с установленными целями и отображать прогресс.
 - 3. **Рекомендации:**
 - На основе анализа данных программа будет давать рекомендации, например, увеличить продолжительность сна или добавить больше физической активности.
 - 4. **Визуализация прогресса:**
 - Программа будет отображать прогресс в виде графиков или диаграмм.
-

Функциональные требования

1. **Установка целей:**
 - Пользователь должен иметь возможность устанавливать цели по активности, питанию и сну.
2. **Отслеживание прогресса:**
 - Программа должна сравнивать текущие данные с установленными целями и отображать прогресс.
3. **Рекомендации:**
 - Программа должна давать рекомендации на основе анализа данных.
4. **Визуализация прогресса:**
 - Программа должна отображать прогресс в виде графиков или диаграмм.

23. Управление музыкальной коллекцией

В вашу программу "Управление музыкальной коллекцией" можно добавить новую функцию "**Экспорт и импорт коллекции**". Эта функция позволит пользователю сохранять свою музыкальную коллекцию в файл и загружать её из файла, что упростит перенос данных между устройствами или создание резервных копий.

Описание новой функции: Экспорт и импорт коллекции

Функция будет включать следующие возможности:

1. **Экспорт коллекции:**
 - Пользователь сможет сохранить всю музыкальную коллекцию в файл (например, в формате JSON или CSV).
 2. **Импорт коллекции:**
 - Пользователь сможет загрузить музыкальную коллекцию из файла, чтобы восстановить данные или добавить новые треки.
 3. **Форматы файлов:**
 - Поддержка популярных форматов, таких как JSON или CSV, для удобства работы с данными.
 4. **Резервное копирование:**
 - Пользователь сможет создавать резервные копии своей коллекции для защиты от потери данных.
-

Функциональные требования

1. **Экспорт коллекции:**
 - Пользователь должен иметь возможность сохранить коллекцию в файл.
2. **Импорт коллекции:**
 - Пользователь должен иметь возможность загрузить коллекцию из файла.
3. **Форматы файлов:**
 - Поддержка JSON или CSV для экспорта и импорта.
4. **Резервное копирование:**
 - Пользователь должен иметь возможность создавать резервные копии коллекции.

24. Управление путешествиями

В вашу программу "Управление путешествием" можно добавить аналогичную функцию "Уведомления о сроках выполнения задач". Эта функция позволит пользователю получать уведомления о приближающихся сроках завершения задач, связанных с путешествием, чтобы своевременно принимать меры для их выполнения.

Описание новой функции: Уведомления о сроках выполнения задач

Функция будет включать следующие возможности:

1. **Уведомления о приближении срока завершения задачи:**
 - Пользователь будет получать уведомление за определённое время до окончания срока задачи (например, за 3 дня).

2. Уведомления о просроченных задачах:

- Пользователь будет получать уведомление, если задача не выполнена к установленному сроку.

3. Настройка уведомлений:

- Пользователь сможет настраивать время уведомлений (например, за сколько дней до окончания срока).

Функциональные требования

1. Уведомления о приближении срока завершения задачи:

- Программа должна отображать уведомление за определённое время до окончания срока задачи.

2. Уведомления о просроченных задачах:

- Программа должна отображать уведомление, если задача не выполнена к установленному сроку.

3. Настройка уведомлений:

- Пользователь должен иметь возможность настраивать время уведомлений (например, за 3 дня до окончания срока).

25. Управление фотографиями

Эта функция позволит пользователю просматривать выбранную фотографию в отдельном окне с возможностью увеличения и уменьшения масштаба.

Описание новой функции: Просмотр фотографий

Функция будет включать следующие возможности:

1. Просмотр выбранной фотографии:

- Пользователь сможет открыть выбранную фотографию в отдельном окне для детального просмотра.

2. Масштабирование фотографии:

- Пользователь сможет увеличивать и уменьшать масштаб фотографии для удобного просмотра.
-

Функциональные требования

1. Просмотр фотографии:

- Программа должна открывать выбранную фотографию в отдельном окне.

2. Масштабирование:

- Пользователь должен иметь возможность увеличивать и уменьшать масштаб фотографии.

26. Управление рецептами и планированием меню

В вашу программу "Планирование меню" можно добавить новую функцию **"Генерация списка покупок"**. Эта функция позволит пользователю автоматически создавать список необходимых ингредиентов для всех рецептов, запланированных на определённый период (например, на неделю). Это упростит процесс покупки продуктов и поможет избежать забытых ингредиентов.

Описание новой функции: Генерация списка покупок

Функция будет включать следующие возможности:

1. **Автоматическое создание списка покупок:**
 - Программа будет анализировать ингредиенты всех рецептов, запланированных на выбранный период, и создавать единый список.
2. **Группировка ингредиентов:**
 - Ингредиенты будут группироваться по категориям (например, овощи, молочные продукты, мясо и т.д.).
3. **Экспорт списка покупок:**
 - Пользователь сможет экспортировать список покупок в текстовый файл или PDF для удобства использования.
4. **Редактирование списка:**
 - Пользователь сможет вручную редактировать список покупок, добавлять или удалять ингредиенты.

27. Управление обучением и курсами

В вашу программу "Управление курсами" можно добавить новую функцию **"Отслеживание прогресса и статистика"**. Эта функция позволит пользователю отслеживать общий прогресс по курсу, анализировать успеваемость и получать статистику по завершённым модулям и темам. Это поможет пользователю лучше понимать, насколько он продвинулся в изучении курса.

Описание новой функции: Отслеживание прогресса и статистика

Функция будет включать следующие возможности:

1. **Общий прогресс по курсу:**
 - Пользователь сможет видеть общий прогресс по курсу, основанный на прогрессе всех модулей.
2. **Статистика по модулям:**
 - Пользователь сможет просматривать статистику по каждому модулю (например, количество завершённых тем, общий прогресс).
3. **Графическое отображение прогресса:**
 - Программа будет отображать прогресс в виде диаграмм или графиков для наглядности.
4. **Уведомления о завершении модулей:**
 - Пользователь будет получать уведомления, когда модуль или курс завершён.

28. Управление резюме и поиском работы

В вашу программу "Управление резюме и поиском работы" можно добавить новую функцию "**Анализ соответствия резюме вакансиям**". Эта функция позволит пользователю автоматически проверять, насколько его резюме соответствует требованиям определённой вакансии. Это поможет пользователю лучше подготовиться к собеседованиям и улучшить свои шансы на трудоустройство.

Описание новой функции: Анализ соответствия резюме вакансиям

Функция будет включать следующие возможности:

1. **Сравнение навыков:**
 - Программа будет сравнивать навыки, указанные в резюме, с требованиями вакансии.
2. **Оценка соответствия:**
 - Программа будет оценивать, насколько резюме соответствует вакансии, и выводить процент соответствия.
3. **Рекомендации по улучшению:**
 - Программа будет давать рекомендации по улучшению резюме (например, добавление недостающих навыков).
4. **Сохранение результатов анализа:**
 - Пользователь сможет сохранять результаты анализа для дальнейшего использования.

29. Управление встречами и мероприятиями

В вашу программу "Управление встречами и мероприятиями" можно добавить новую функцию **"Интеграция с календарём и уведомлениями"**. Эта функция позволит пользователю синхронизировать события с внешними календарями (например, Google Calendar, Outlook) и получать уведомления о предстоящих событиях через электронную почту или push-уведомления.

Описание новой функции: Интеграция с календарём и уведомлениями

Функция будет включать следующие возможности:

1. **Синхронизация с календарём:**
 - Пользователь сможет экспортировать события в популярные календари (Google Calendar, Outlook и т.д.).
2. **Уведомления о событиях:**
 - Пользователь будет получать уведомления о предстоящих событиях за определённое время до их начала (например, за 1 час, за 1 день).
3. **Настройка уведомлений:**
 - Пользователь сможет настраивать способ получения уведомлений (электронная почта, push-уведомления) и время напоминания.
4. **Автоматическое обновление календаря:**
 - При изменении события в программе (например, изменение времени или места) календарь будет автоматически обновляться.

30. Управление техникой и оборудованием

В программу "Управление оборудованием" можно добавить новую функцию **"Уведомления о необходимости обслуживания оборудования"**. Эта функция позволит пользователю получать уведомления о том, что оборудование требует обслуживания, чтобы своевременно проводить профилактические работы и избежать поломок.

Описание новой функции: Уведомления о необходимости обслуживания оборудования

Функция будет включать следующие возможности:

1. **Уведомления о необходимости обслуживания:**
 - Пользователь будет получать уведомление, если с момента последнего обслуживания прошло больше определённого времени (например, 6 месяцев).
 2. **Настройка уведомлений:**
 - Пользователь сможет настраивать интервал времени, через который нужно уведомлять о необходимости обслуживания (например, 3, 6 или 12 месяцев).
 3. **Уведомления о просроченном обслуживании:**
 - Пользователь будет получать уведомление, если оборудование не обслуживалось дольше установленного срока.
-

Функциональные требования

1. **Уведомления о необходимости обслуживания:**
 - Программа должна отображать уведомление, если с момента последнего обслуживания прошло больше установленного времени.
2. **Уведомления о просроченном обслуживании:**
 - Программа должна отображать уведомление, если оборудование не обслуживалось дольше установленного срока.
3. **Настройка уведомлений:**
 - Пользователь должен иметь возможность настраивать интервал времени для уведомлений (например, 3, 6 или 12 месяцев).

31. Управление автомобилем и его обслуживанием

В программу "Управление автомобилем" можно добавить новую функцию **"Уведомления о плановом обслуживании"**. Эта функция позволит пользователю получать напоминания о предстоящем плановом обслуживании автомобиля, чтобы избежать пропуска важных технических работ.

Описание новой функции: Уведомления о плановом обслуживании

Функция будет включать следующие возможности:

1. **Уведомления о предстоящем обслуживании:**
 - Пользователь будет получать уведомление за определённое время до даты планового обслуживания (например, за 7 дней).

2. **Уведомления о просроченном обслуживании:**
 - Пользователь будет получать уведомление, если плановое обслуживание не выполнено к установленной дате.
 3. **Настройка уведомлений:**
 - Пользователь сможет настраивать время уведомлений (например, за сколько дней до обслуживания).
-

Функциональные требования

1. **Уведомления о предстоящем обслуживании:**
 - Программа должна отображать уведомление за определённое время до даты планового обслуживания.
2. **Уведомления о просроченном обслуживании:**
 - Программа должна отображать уведомление, если плановое обслуживание не выполнено к установленной дате.
3. **Настройка уведомлений:**
 - Пользователь должен иметь возможность настраивать время уведомлений (например, за 7 дней до обслуживания).

32. Управление энергопотреблением

В программу "Управление энергопотреблением" можно добавить новую функцию **"Анализ энергопотребления и рекомендации по экономии"**. Эта функция позволит пользователю анализировать данные о потреблении энергии и получать рекомендации по снижению затрат на электроэнергию.

Описание новой функции: Анализ энергопотребления и рекомендации по экономии

Функция будет включать следующие возможности:

1. **Анализ энергопотребления:**
 - Программа будет анализировать данные о потреблении энергии для каждого устройства и выводить статистику (например, общее потребление, среднее потребление в час).
2. **Рекомендации по экономии:**
 - На основе анализа программа будет предлагать рекомендации по снижению энергопотребления (например, отключение неиспользуемых устройств, замена на энергоэффективные приборы).
3. **Визуализация данных:**

- Программа будет отображать графики или диаграммы для наглядного представления данных о потреблении.
-

Функциональные требования

- 1. Анализ энергопотребления:**
 - Программа должна рассчитывать общее потребление энергии для каждого устройства и выводить статистику.
- 2. Рекомендации по экономии:**
 - Программа должна предлагать рекомендации на основе анализа данных.
- 3. Визуализация данных:**
 - Программа должна отображать данные в виде графиков или диаграмм.

33. Управление личными целями и задачами

В программу "Управление личными целями" можно добавить новую функцию **"Анализ прогресса и рекомендации по достижению целей"**. Эта функция позволит пользователю анализировать прогресс выполнения целей, получать рекомендации по их достижению и визуализировать данные для лучшего понимания.

Описание новой функции: Анализ прогресса и рекомендации по достижению целей

Функция будет включать следующие возможности:

- 1. Анализ прогресса:**
 - Программа будет анализировать прогресс выполнения целей и выводить статистику (например, средний прогресс, количество завершённых целей).
- 2. Рекомендации по достижению целей:**
 - На основе анализа программа будет предлагать рекомендации (например, разбить цель на подзадачи, установить более реалистичные сроки).
- 3. Визуализация данных:**
 - Программа будет отображать данные в виде графиков или диаграмм (например, прогресс по целям, распределение целей по срокам).

Функциональные требования

1. Анализ прогресса:

- Программа должна рассчитывать средний прогресс по всем целям и выводить статистику.

2. Рекомендации по достижению целей:

- Программа должна предлагать рекомендации на основе анализа данных.

3. Визуализация данных:

- Программа должна отображать данные в виде графиков или диаграмм.

34. Управление коллекцией игр

В программу "Управление коллекцией игр" можно добавить новую функцию **"Анализ коллекции игр и рекомендации"**. Эта функция позволит пользователю анализировать свою коллекцию игр, получать статистику и рекомендации (например, по жанрам, платформам или рейтингу), а также визуализировать данные для лучшего понимания.

Описание новой функции: Анализ коллекции игр и рекомендации

Функция будет включать следующие возможности:

1. Анализ коллекции:

- Программа будет анализировать коллекцию игр и выводить статистику (например, количество игр по жанрам, платформам, средний рейтинг).

2. Рекомендации:

- На основе анализа программа будет предлагать рекомендации (например, попробовать игры с высоким рейтингом, изучить новые жанры).

3. Визуализация данных:

- Программа будет отображать данные в виде графиков или диаграмм (например, распределение игр по жанрам, платформам).

Функциональные требования

1. **Анализ коллекции:**

- Программа должна рассчитывать статистику по коллекции игр (например, количество игр по жанрам, платформам, средний рейтинг).

2. **Рекомендации:**

- Программа должна предлагать рекомендации на основе анализа данных.

3. **Визуализация данных:**

- Программа должна отображать данные в виде графиков или диаграмм.

35. Управление доставкой

В программу "Управление процессом доставки" можно добавить новую функцию **"Уведомления о статусе доставки и аналитика"**. Эта функция позволит пользователю получать уведомления об изменениях статуса доставки, а также анализировать данные о доставках для улучшения процесса.

Описание новой функции: Уведомления о статусе доставки и аналитика

Функция будет включать следующие возможности:

1. **Уведомления о статусе доставки:**

- Пользователь будет получать уведомления при изменении статуса доставки (например, "Отправка в пути", "Отправка задержана", "Отправка доставлена").

2. **Аналитика доставок:**

- Программа будет анализировать данные о доставках и выводить статистику (например, количество доставок по статусам, среднее время доставки).

3. **Рекомендации по улучшению процесса:**

- На основе анализа программа будет предлагать рекомендации (например, оптимизация маршрутов, улучшение логистики).
-

Функциональные требования

1. **Уведомления о статусе доставки:**

- Программа должна отправлять уведомления при изменении статуса доставки.

2. Аналитика доставок:

- Программа должна рассчитывать статистику по доставкам (например, количество доставок по статусам, среднее время доставки).

3. Рекомендации по улучшению процесса:

- Программа должна предлагать рекомендации на основе анализа данных.