

Министерство образования Республики Беларусь
Учреждение образования
**«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»**
Институт информационных технологий

Специальность: Программная инженерия

Курсовая работа
По дисциплине «Конструирование программного обеспечения»

Тема: Программа учета заявок на обмен квартир и поиска вариантов обмена

Студент 1 курса, ЗФО
Группа №481571
ФИО Зеневич Александр Олегович

Минск, 2025

СОДЕРЖАНИЕ

Введение	3
1 Постановка задачи	4
2 Программная реализация	5
2.1 Краткая характеристика инструментальной программной среды	5
2.2 Структура программы	5
2.3 Описание программных модулей	6
3 Руководство пользователя	7
Заключение	8
Список использованных источников	9
Приложение А Программный код	10
Приложение Б Алгоритм работы программы	23
Приложение В Схема ресурсов программного средства	25

Введение

В современном мире рынок недвижимости, включая обмен квартирами, представляет собой сложную систему, где эффективное управление заявками играет ключевую роль. Автоматизация учета заявок на обмен позволяет минимизировать ручной труд, снизить ошибки и ускорить поиск подходящих вариантов. Курсовая работа посвящена разработке программы для учета заявок на обмен квартир и поиска вариантов обмена.

Цели проекта включают закрепление теоретических знаний по объектно-ориентированному программированию, освоение языка C# и среды Visual Studio, а также развитие навыков создания пользовательского интерфейса на базе Windows Forms. Задачи охватывают анализ предметной области, проектирование классов, реализацию логики хранения данных с использованием контейнера HashSet<T> (эквивалент set в C# для обеспечения уникальности), обеспечение контроля ввода и тестирование программы.

Проект реализован с учетом принципов ООП: инкапсуляция (закрытые поля классов), наследование (не требуется явно, но используется в структуре форм), полиморфизм (в обработке событий). Хранение данных в памяти с использованием HashSet<Application> гарантирует отсутствие дубликатов заявок (по критерию Equals/GetHashCode на основе фамилии и характеристик квартир).

1 Постановка задачи

Постановка задачи основана на необходимости автоматизации процесса учета и поиска заявок на обмен квартирами. В предметной области обмен квартир – это процесс, где заявитель предлагает свою квартиру и указывает желаемые характеристики новой. Ключевые проблемы ручного учета: дубликаты заявок, медленный поиск, ошибки в расчетах (например, сравнение площадей).

Функциональные требования:

- Ввод заявки: Пользователь вводит фамилию и инициалы, характеристики имеющейся и требуемой квартир (комнаты: целое >0 , площадь: вещественное >0 , этаж: целое >0 , район: строка). Контроль: валидация на пустые поля, положительные числа, целостность.
- Поиск варианта: Ввод новой заявки, поиск в картотеке по критериям: совпадение комнат (имеющаяся текущей = требуемая в картотеке, и наоборот), совпадение этажности, разница площадей $\leq 10\% (|area1 - area2| / \max(area1, area2) \leq 0.1)$. Если найдено — вывести совпадающую заявку, удалить её из картотеки (симуляция обмена). Иначе добавить новую в картотеку.
- Вывод картотеки: Отображение всех заявок в таблице или списке с полными данными.

Нефункциональные требования:

- Интерфейс: Windows Forms, меню или кнопки для навигации.
- Хранение: HashSet<Application> для уникальности (переопределены Equals и GetHashCode для сравнения по всем полям).
- Обработка ошибок: MessageBox для уведомлений (некорректный ввод, пустая картотека).
- Тестирование: Контрольные примеры (3-5 сценариев).

Ограничения: Данные в памяти (без БД), уникальность заявок по содержимому.

2 Программная реализация

2.1 Краткая характеристика инструментальной программной среды

Для реализации проекта выбрана кроссплатформенная интегрированная среда разработки JetBrains Rider – мощный инструмент для создания приложений на .NET. JetBrains Rider поддерживает язык C# (версия 13.0 в проекте), фреймворк .NET 9.0, и компоненты Windows Forms для GUI.

Характеристики среды:

- Язык программирования: C# – объектно-ориентированный, с сильной типизацией, автоматическим управлением памятью (GC), поддержкой LINQ для запросов к коллекциям. В проекте использованы классы, свойства, события, исключения.
- Фреймворк: .NET 9.0 — кроссплатформенный, с библиотеками System.Windows.Forms для форм, System.Collections.Generic для HashSet.
- Интерфейс разработки: Дизайнер форм для визуального создания UI (кнопки, TextBox, DataGridView), IntelliSense для автодополнения, отладчик с breakpoints.
- Дополнительные инструменты: NuGet для пакетов (не использованы), Git для версионного контроля.
- Преимущества: Бесплатная версия, интеграция с Windows, простота развертывания (exe-файл).
- Ограничения: Зависимость от Windows для Forms (альтернатива — WinUI или MAUI для кроссплатформы, но не требуется).

Проект создан как Windows Forms App (.NET), с target framework .NET 9.0. Код скомпилирован в release-режиме для тестирования.

2.2 Структура программы

Программа структурирована по принципам ООП и MVC-подобной модели: модель (классы данных), вид (формы), контроллер (логика в обработчиках событий).

Классы данных:

- Apartment: Свойства Rooms (int), Area (double), Floor (int), District (string). Конструктор, ToString.
- Application: Свойства SurnameInitials (string), OwnedApartment (Apartment), RequiredApartment (Apartment). Конструктор, переопределены Equals/GetHashCode для HashSet (сравнение по всем полям). ToString для вывода.

Хранение: Статический HashSet<Application> в главной форме (для глобального доступа).

Формы:

- MainForm: Главная форма с кнопками «Ввод заявки», «Поиск варианта», «Вывод картотеки», «Выход». DataGridView для вывода.
- InputForm: Диалоговая форма для ввода заявки (TextBox для полей, Button "OK"/"Cancel"). Валидация в обработчике.

Структура проекта:

- Program.cs: Точка входа, запуск MainForm.
- MainForm.cs / MainForm.Designer.cs: Логика и дизайн главной формы.
- InputForm.cs / InputForm.Designer.cs: Логика и дизайн формы ввода.
- Apartment.cs: Класс квартиры.
- Application.cs: Класс заявки.

Поток выполнения: Запуск → Главная форма → Выбор функции → Обработка (ввод/поиск/вывод) → Возврат.

2.3 Описание программных модулей

Модуль данных (Apartment.cs, Application.cs):

Определяют структуру данных. Apartment инкапсулирует характеристики квартиры. Application – заявку, с методами для сравнения (Equals использует последовательное сравнение полей; GetHashCode комбинирует хэши).

Модуль интерфейса (Формы):

- MainForm: Обработчики кнопок: InputButton_Click открывает InputForm для ввода, добавляет в HashSet. SearchButton_Click открывает InputForm для новой заявки, ищет совпадение (LINQ-запрос), если найдено — MessageBox с деталями, удаление; иначе добавление. OutputButton_Click заполняет DataGridView (DataSource = list from HashSet).

- InputForm: Load устанавливает заголовок. OKButton_Click валидирует (try-parse для чисел, проверки >0, не пустые), создает Application, возвращает DialogResult.OK.

Валидация: В InputForm – проверка на int,double, положительность, не пустоту. Ошибки — MessageBox.

Поиск: Для новой заявки newApp:

```
var match = cardIndex.FirstOrDefault(app => app.OwnedApartment.Rooms ==  
newApp.RequiredApartment.Rooms && app.OwnedApartment.Floor ==  
newApp.RequiredApartment.Floor && app.RequiredApartment.Floor ==  
newApp.OwnedApartment.Floor && Math.Abs(app.OwnedApartment.Area -  
newApp.RequiredApartment.Area) / Math.Max(app.OwnedApartment.Area,  
newApp.RequiredApartment.Area) <= 0.1 && Math.Abs(app.RequiredApartment.Area -  
newApp.OwnedApartment.Area) / Math.Max(app.RequiredApartment.Area,  
newApp.OwnedApartment.Area) <= 0.1 );
```

Если match != null – вывод и удаление; else добавить newApp.

Вывод: DataGridView с колонками для всех полей (используя binding).

3 Руководство пользователя

Установка: Скопировать exe из bin/Release. Запуск на Windows 10+.

Интерфейс:

– Главная форма: Кнопки «Ввод заявки», «Поиск варианта», «Вывод картотеки», «Выход».

– Форма ввода: Поля для фамилии, характеристик квартир. «OK» – сохранить, «Cancel» – отмена.

Сценарии:

1. Ввод: Нажать «Ввод», заполнить поля (пример: Иванов И.И., имеющаяся: 2 комнаты, 50 м², 3 этаж, Центр; требуемая: 3 комнаты, 60 м², 5 этаж, Заводской). OK – добавлено.

2. Поиск: Нажать «Поиск», ввести новую. Если совпадение – в сообщение с деталями, удалено; иначе добавлено.

3. Вывод: Нажать «Вывод» – таблица с заявками.

4. Ошибки: Некорректный ввод – сообщение. Пустая картотека – уведомление.

Тестирование:

Пример 1: Ввод двух заявок с совпадением – поиск находит, удаляет.

Пример 2: Несовпадение – добавляет.

Пример 3: Вывод пустой/полней картотеки.

Заключение

Курсовой проект позволил закрепить навыки ООП в C#, работы с Windows Forms и коллекциями. Разработанная программа полностью соответствует задаче, обеспечивает надежный учет и поиск заявок. Возможные улучшения: добавление БД (SQLite), сериализация в файл, расширенный поиск.

Список использованных источников

- 1 Биллиг, В. А. Основы программирования на С#; Интернет-университет информационных технологий, Бином. Лаборатория знаний - М., 2012. - 488 с.
- 2 Арсеновски, Даниэль Рефакторинг в С# и ASP.NET для профессионалов; Вильямс - М., 2010. - 528 с.
- 3 Евдокимов, П. С# на примерах; ИЛ - Москва, 2016. - 865 с.
- 4 Колесников, А. П. Методы численного анализа, изложенные на языке формул и алгоритмическом языке С#; Высшая школа - Москва, 2017. - 414 с.
- 5 Мартин, Р. С., Мартин М. Принципы, паттерны и методики гибкой разработки на языке С#; Символ-плюс - М., 2011. - 768 с.
- 6 Вирт, Н. Алгоритмы и структуры данных / Н. Вирт. – СПб.: «Невский диалект», 2005.
- 7 Бишоп, Дж. С# в кратком изложении; Бином. Лаборатория знаний - М., 2015. - **234** с.
- 8 Арсеновски, Даниэль Рефакторинг в С# и ASP.NET для профессионалов; Вильямс - М., 2010. - 528 с.
- 9 Кристиан, Нейгел Visual C# 2010. Полный курс; Диалектика / Вильямс - М., 2011. - **575** с.
- 10 Мартин, Р. С., Мартин М. Принципы, паттерны и методики гибкой разработки на языке С#; Символ-плюс - М., 2011. - 768 с.
- 11 Дэвис, Алекс Асинхронное программирование в С# 5.0; ДМК Пресс - М., 2015. - 120 с.

Приложение А (обязательное) Программный код

```
// Файл Apartment.cs
namespace Cw01;

/// <summary>
/// Класс, представляющий квартиру с характеристиками: количество комнат, площадь, этаж, район.
/// </summary>
public class Apartment
{
    public int Rooms { get; set; } // Количество комнат
    public double Area { get; set; } // Площадь в м2
    public int Floor { get; set; } // Этаж
    public string District { get; set; } // Район

    /// <summary>
    /// Конструктор для инициализации квартиры.
    /// </summary>
    /// <param name="rooms">Количество комнат.</param>
    /// <param name="area">Площадь.</param>
    /// <param name="floor">Этаж.</param>
    /// <param name="district">Район.</param>
    public Apartment(int rooms, double area, int floor, string district)
    {
        Rooms = rooms;
        Area = area;
        Floor = floor;
        District = district;
    }

    /// <summary>
    /// Переопределение ToString для вывода характеристик квартиры.
    /// </summary>
    /// <returns>Строка с данными квартиры.</returns>
    public override string ToString() => $"Комнат: {Rooms}, Площадь: {Area} м2, Этаж: {Floor}, Район: {District}";
}

// Файл ExchangeApplication.cs
namespace Cw01;

/// <summary>
/// Класс, представляющий заявку на обмен: фамилия, имеющаяся и требуемая квартиры.
/// </summary>
public class ExchangeApplication
{
    public string SurnameInitials { get; set; } // Фамилия и инициалы
    public Apartment OwnedApartment { get; set; } // Имеющаяся квартира
    public Apartment RequiredApartment { get; set; } // Требуемая квартира

    /// <summary>
    /// Конструктор для инициализации заявки.
    /// </summary>
    /// <param name="surnameInitials">Фамилия и инициалы.</param>
    /// <param name="owned">Имеющаяся квартира.</param>
    /// <param name="required">Требуемая квартира.</param>
```

```

    public ExchangeApplication(string surnameInitials, Apartment owned, Apartment
required)
{
    SurnameInitials = surnameInitials;
    OwnedApartment = owned;
    RequiredApartment = required;
}

/// <summary>
/// Переопределение Equals для сравнения заявок по всем полям.
/// </summary>
/// <param name="obj">Объект для сравнения.</param>
/// <returns>True, если заявки равны.</returns>
public override bool Equals(object obj)
{
    if (obj is ExchangeApplication other)
    {
        return SurnameInitials == other.SurnameInitials &&
            OwnedApartment.Rooms == other.OwnedApartment.Rooms &&
            OwnedApartment.Area == other.OwnedApartment.Area &&
            OwnedApartment.Floor == other.OwnedApartment.Floor &&
            OwnedApartment.District == other.OwnedApartment.District &&
            RequiredApartment.Rooms == other.RequiredApartment.Rooms &&
            RequiredApartment.Area == other.RequiredApartment.Area &&
            RequiredApartment.Floor == other.RequiredApartment.Floor &&
            RequiredApartment.District == other.RequiredApartment.District;
    }
    return false;
}

/// <summary>
/// Переопределение GetHashCode для использования в HashSet.
/// </summary>
/// <returns>Хэш-код заявки.</returns>
public override int GetHashCode()
{
    var hasher = new HashCode();
    hasher.Add(SurnameInitials);
    hasher.Add(OwnedApartment.Rooms);
    hasher.Add(OwnedApartment.Area);
    hasher.Add(OwnedApartment.Floor);
    hasher.Add(OwnedApartment.District);
    hasher.Add(RequiredApartment.Rooms);
    hasher.Add(RequiredApartment.Area);
    hasher.Add(RequiredApartment.Floor);
    hasher.Add(RequiredApartment.District);
    return hasher.ToHashCode();
}

/// <summary>
/// Переопределение ToString для вывода заявки.
/// </summary>
/// <returns>Строка с данными заявки.</returns>
public override string ToString()
    => $"Заявитель: {SurnameInitials}\nИмеющаяся: {OwnedApartment}\nТребуемая:
{RequiredApartment}";
}

// Файл InputForm.cs
namespace Cw01;

public partial class InputForm : Form

```

```

{
    public InputForm(string title)
    {
        InitializeComponent();
        Text = title;
    }

    /// <summary>
    /// Получение созданной заявки.
    /// </summary>
    /// <returns>Объект ExchangeApplication.</returns>
    public ExchangeApplication GetApplication()
    {
        var ownedRooms = int.Parse(ownedRoomsTextBox.Text);
        var ownedArea = double.Parse(ownedAreaTextBox.Text);
        var ownedFloor = int.Parse(ownedFloorTextBox.Text);
        var ownedDistrict = ownedDistrictTextBox.Text;

        var requiredRooms = int.Parse(requiredRoomsTextBox.Text);
        var requiredArea = double.Parse(requiredAreaTextBox.Text);
        var requiredFloor = int.Parse(requiredFloorTextBox.Text);
        var requiredDistrict = requiredDistrictTextBox.Text;

        var owned = new Apartment(ownedRooms, ownedArea, ownedFloor, ownedDistrict);
        var required = new Apartment(requiredRooms, requiredArea, requiredFloor,
        requiredDistrict);

        return new ExchangeApplication(surnameTextBox.Text, owned, required);
    }

    // Обработчик кнопки "OK"
    private void OKButton_Click(object sender, EventArgs e)
    {
        try
        {
            // Валидация
            if (string.IsNullOrWhiteSpace(surnameTextBox.Text) ||
                string.IsNullOrWhiteSpace(ownedDistrictTextBox.Text) ||
                string.IsNullOrWhiteSpace(requiredDistrictTextBox.Text))
            {
                throw new Exception("Все текстовые поля должны быть заполнены.");
            }

            var ownedRooms = int.Parse(ownedRoomsTextBox.Text);
            var ownedArea = double.Parse(ownedAreaTextBox.Text);
            var ownedFloor = int.Parse(ownedFloorTextBox.Text);

            var requiredRooms = int.Parse(requiredRoomsTextBox.Text);
            var requiredArea = double.Parse(requiredAreaTextBox.Text);
            var requiredFloor = int.Parse(requiredFloorTextBox.Text);

            if (ownedRooms <= 0 || ownedArea <= 0 || ownedFloor <= 0 ||
                requiredRooms <= 0 || requiredArea <= 0 || requiredFloor <= 0)
            {
                throw new Exception("Числовые значения должны быть положительными.");
            }

            DialogResult = DialogResult.OK;
            Close();
        }
        catch (FormatException)
        {

```

```

        MessageBox.Show(
            "Некорректный формат чисел. Используйте целые для комнат/этажа,
вещественные для площади.",
            "Ошибка",
            MessageBoxButtons.OK,
            MessageBoxIcon.Error);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Ошибка", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }
}

// Обработчик кнопки "Cancel"
private void CancelButton_Click(object sender, EventArgs e)
{
    DialogResult = DialogResult.Cancel;
    Close();
}
}

// Файл MainForm.cs
namespace Cw01;

public partial class MainForm : Form
{
    private static readonly HashSet<ExchangeApplication> CardIndex = []; // Картотека
(set)

    public MainForm()
    {
        InitializeComponent();
    }

    // Обработчик кнопки "Ввод заявки"
    private void InputButton_Click(object sender, EventArgs e)
    {
        using (var inputForm = new InputForm("Ввод новой заявки"))
        {
            if (inputForm.ShowDialog() == DialogResult.OK)
            {
                var newApp = inputForm.GetApplication();
                if (CardIndex.Add(newApp))
                {
                    MessageBox.Show(
                        "Заявка добавлена успешно.",
                        "Успех",
                        MessageBoxButtons.OK,
                        MessageBoxIcon.Information);
                }
                else
                {
                    MessageBox.Show(
                        "Такая заявка уже существует.",
                        "Ошибка",
                        MessageBoxButtons.OK,
                        MessageBoxIcon.Warning);
                }
            }
        }
    }
}

```

```

// Обработчик кнопки "Поиск варианта"
private void SearchButton_Click(object sender, EventArgs e)
{
    using (var inputForm = new InputForm("Ввод заявки для поиска"))
    {
        if (inputForm.ShowDialog() == DialogResult.OK)
        {
            var newApp = inputForm.GetApplication();

            // Поиск совпадения
            var match = CardIndex.FirstOrDefault(app =>
                app.OwnedApartment.Rooms == newApp.RequiredApartment.Rooms &&
                app.RequiredApartment.Rooms == newApp.OwnedApartment.Rooms &&
                app.OwnedApartment.Floor == newApp.RequiredApartment.Floor &&
                app.RequiredApartment.Floor == newApp.OwnedApartment.Floor &&
                Math.Abs(app.OwnedApartment.Area - newApp.RequiredApartment.Area) /
                Math.Max(app.OwnedApartment.Area, newApp.RequiredApartment.Area) <= 0.1 &&
                Math.Abs(app.RequiredApartment.Area - newApp.OwnedApartment.Area) /
                Math.Max(app.RequiredApartment.Area, newApp.OwnedApartment.Area) <= 0.1
            );

            if (match != null)
            {
                MessageBox.Show(
                    $"Найден подходящий вариант:\n{match}\nЗаявка удалена из
карточки.",

                    "Совпадение найдено",
                    MessageBoxButtons.OK,
                    MessageBoxIcon.Information);
                CardIndex.Remove(match);
            }
            else
            {
                if (CardIndex.Add(newApp))
                {
                    MessageBox.Show(
                        "Совпадение не найдено. Заявка добавлена в картотеку.",
                        "Добавлено",
                        MessageBoxButtons.OK,
                        MessageBoxIcon.Information);
                }
                else
                {
                    MessageBox.Show(
                        "Такая заявка уже существует.",
                        "Ошибка",
                        MessageBoxButtons.OK,
                        MessageBoxIcon.Warning);
                }
            }
        }
    }
}

// Обработчик кнопки "Вывод картотеки"
private void OutputButton_Click(object sender, EventArgs e)
{
    if (CardIndex.Count == 0)
    {
        MessageBox.Show("Картотека пуста.", "Информация", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
}

```

```

        return;
    }

    dataGridView.Rows.Clear();
    dataGridView.Columns.Clear();

    // Настройка колонок
    dataGridView.Columns.Add("Surname", "Фамилия и инициалы");
    dataGridView.Columns.Add("OwnedRooms", "Имеющаяся: Комнаты");
    dataGridView.Columns.Add("OwnedArea", "Имеющаяся: Площадь");
    dataGridView.Columns.Add("OwnedFloor", "Имеющаяся: Этаж");
    dataGridView.Columns.Add("OwnedDistrict", "Имеющаяся: Район");
    dataGridView.Columns.Add("RequiredRooms", "Требуемая: Комнаты");
    dataGridView.Columns.Add("RequiredArea", "Требуемая: Площадь");
    dataGridView.Columns.Add("RequiredFloor", "Требуемая: Этаж");
    dataGridView.Columns.Add("RequiredDistrict", "Требуемая: Район");

    foreach (var app in CardIndex)
    {
        dataGridView.Rows.Add(app.SurnameInitials,
            app.OwnedApartment.Rooms,
            app.OwnedApartment.Area,
            app.OwnedApartment.Floor,
            app.OwnedApartment.District,
            app.RequiredApartment.Rooms,
            app.RequiredApartment.Area,
            app.RequiredApartment.Floor,
            app.RequiredApartment.District);
    }
}

// Обработчик кнопки "Выход"
private void ExitButton_Click(object sender, EventArgs e)
{
    Close();
}
}

// Файл Program.cs
namespace Cw01;

internal static class Program
{
    /// <summary>
    /// The main entry point for the application.
    /// </summary>
    [STAThread]
    private static void Main()
    {
        // To customize application configuration such as set high DPI settings or
        default font,
        // see https://aka.ms/applicationconfiguration.
        ApplicationConfiguration.Initialize();
        Application.Run(new MainForm());
    }
}

// Файл MainForm.Designer.cs
namespace Cw01
{
    partial class MainForm
    {

```

```

/// <summary>
/// Required designer variable.
/// </summary>
private System.ComponentModel.IContainer components = null;

/// <summary>
/// Clean up any resources being used.
/// </summary>
/// <param name="disposing">true if managed resources should be disposed;
otherwise, false.</param>
protected override void Dispose(bool disposing)
{
    if (disposing && (components != null))
    {
        components.Dispose();
    }
    base.Dispose(disposing);
}

#region Windows Form Designer generated code

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    this.InputButton = new System.Windows.Forms.Button();
    this.SearchButton = new System.Windows.Forms.Button();
    this.OutputButton = new System.Windows.Forms.Button();
    this.ExitButton = new System.Windows.Forms.Button();
    this.dataGridView = new System.Windows.Forms.DataGridView();
    this.SuspendLayout();
    //
    // InputButton
    //
    this.InputButton.Location = new System.Drawing.Point(12, 12);
    this.InputButton.Name = "InputButton";
    this.InputButton.Size = new System.Drawing.Size(150, 30);
    this.InputButton.TabIndex = 0;
    this.InputButton.Text = "Ввод заявки";
    this.InputButton.UseVisualStyleBackColor = true;
    this.InputButton.Click += new
System.EventHandler(this.InputButton_Click);
    //
    // SearchButton
    //
    this.SearchButton.Location = new System.Drawing.Point(168, 12);
    this.SearchButton.Name = "SearchButton";
    this.SearchButton.Size = new System.Drawing.Size(150, 30);
    this.SearchButton.TabIndex = 1;
    this.SearchButton.Text = "Поиск варианта";
    this.SearchButton.UseVisualStyleBackColor = true;
    this.SearchButton.Click += new
System.EventHandler(this.SearchButton_Click);
    //
    // OutputButton
    //
    this.OutputButton.Location = new System.Drawing.Point(324, 12);
    this.OutputButton.Name = "OutputButton";
    this.OutputButton.Size = new System.Drawing.Size(150, 30);
    this.OutputButton.TabIndex = 2;
}

```

```

        this.OutputButton.Text = "Вывод картотеки";
        this.OutputButton.UseVisualStyleBackColor = true;
        this.OutputButton.Click += new
System.EventHandler(this.OutputButton_Click);
        //
        // ExitButton
        //
        this.ExitButton.Location = new System.Drawing.Point(480, 12);
        this.ExitButton.Name = "ExitButton";
        this.ExitButton.Size = new System.Drawing.Size(150, 30);
        this.ExitButton.TabIndex = 3;
        this.ExitButton.Text = "Выход";
        this.ExitButton.UseVisualStyleBackColor = true;
        this.ExitButton.Click += new System.EventHandler(this.ExitButton_Click);
        //
        // dataGridView
        //
        this.dataGridView.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
        this.dataGridView.Location = new System.Drawing.Point(12, 48);
        this.dataGridView.Name = "dataGridView";
        this.dataGridView.RowHeadersWidth = 51;
        this.dataGridView.RowTemplate.Height = 29;
        this.dataGridView.Size = new System.Drawing.Size(618, 300);
        this.dataGridView.TabIndex = 4;
        //
        // MainForm
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 15F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(642, 360);
        this.Controls.Add(this.dataGridView);
        this.Controls.Add(this.ExitButton);
        this.Controls.Add(this.OutputButton);
        this.Controls.Add(this.SearchButton);
        this.Controls.Add(this.InputButton);
        this.Name = "MainForm";
        this.Text = "Учет заявок на обмен квартир";
        this.ResumeLayout(false);
    }

}

#endregion

private Button InputButton;
private Button SearchButton;
private Button OutputButton;
private Button ExitButton;
private DataGridView dataGridView;
}

}

// Файл InputForm.Designer.cs
namespace Cw01
{
    partial class InputForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

```

```

/// <summary>
/// Clean up any resources being used.
/// </summary>
/// <param name="disposing">true if managed resources should be disposed;
otherwise, false.</param>
protected override void Dispose(bool disposing)
{
    if (disposing && (components != null))
    {
        components.Dispose();
    }
    base.Dispose(disposing);
}

#region Windows Form Designer generated code

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    this.surnameLabel = new System.Windows.Forms.Label();
    this.surnameTextBox = new System.Windows.Forms.TextBox();
    this.ownedGroupBox = new System.Windows.Forms.GroupBox();
    this.ownedRoomsLabel = new System.Windows.Forms.Label();
    this.ownedRoomsTextBox = new System.Windows.Forms.TextBox();
    this.ownedAreaLabel = new System.Windows.Forms.Label();
    this.ownedAreaTextBox = new System.Windows.Forms.TextBox();
    this.ownedFloorLabel = new System.Windows.Forms.Label();
    this.ownedFloorTextBox = new System.Windows.Forms.TextBox();
    this.ownedDistrictLabel = new System.Windows.Forms.Label();
    this.ownedDistrictTextBox = new System.Windows.Forms.TextBox();
    this.requiredGroupBox = new System.Windows.Forms.GroupBox();
    this.requiredRoomsLabel = new System.Windows.Forms.Label();
    this.requiredRoomsTextBox = new System.Windows.Forms.TextBox();
    this.requiredAreaLabel = new System.Windows.Forms.Label();
    this.requiredAreaTextBox = new System.Windows.Forms.TextBox();
    this.requiredFloorLabel = new System.Windows.Forms.Label();
    this.requiredFloorTextBox = new System.Windows.Forms.TextBox();
    this.requiredDistrictLabel = new System.Windows.Forms.Label();
    this.requiredDistrictTextBox = new System.Windows.Forms.TextBox();
    this.OKButton = new System.Windows.Forms.Button();
    this.CancelButton = new System.Windows.Forms.Button();
    this.ownedGroupBox.SuspendLayout();
    this.requiredGroupBox.SuspendLayout();
    this.SuspendLayout();
    //
    // surnameLabel
    //
    this.surnameLabel.AutoSize = true;
    this.surnameLabel.Location = new System.Drawing.Point(12, 12);
    this.surnameLabel.Name = "surnameLabel";
    this.surnameLabel.Size = new System.Drawing.Size(116, 15);
    this.surnameLabel.TabIndex = 0;
    this.surnameLabel.Text = "Фамилия и инициалы:";
    //
    // surnameTextBox
    //
    this.surnameTextBox.Location = new System.Drawing.Point(134, 9);
    this.surnameTextBox.Name = "surnameTextBox";
    this.surnameTextBox.Size = new System.Drawing.Size(200, 23);
}

```

```

this.surnameTextBox.TabIndex = 1;
//
// ownedGroupBox
//
this.ownedGroupBox.Controls.Add(this.ownedRoomsLabel);
this.ownedGroupBox.Controls.Add(this.ownedRoomsTextBox);
this.ownedGroupBox.Controls.Add(this.ownedAreaLabel);
this.ownedGroupBox.Controls.Add(this.ownedAreaTextBox);
this.ownedGroupBox.Controls.Add(this.ownedFloorLabel);
this.ownedGroupBox.Controls.Add(this.ownedFloorTextBox);
this.ownedGroupBox.Controls.Add(this.ownedDistrictLabel);
this.ownedGroupBox.Controls.Add(this.ownedDistrictTextBox);
this.ownedGroupBox.Location = new System.Drawing.Point(12, 38);
this.ownedGroupBox.Name = "ownedGroupBox";
this.ownedGroupBox.Size = new System.Drawing.Size(322, 150);
this.ownedGroupBox.TabIndex = 2;
this.ownedGroupBox.TabStop = false;
this.ownedGroupBox.Text = "Имеющаяся квартира";
//
// ownedRoomsLabel
//
this.ownedRoomsLabel.AutoSize = true;
this.ownedRoomsLabel.Location = new System.Drawing.Point(6, 25);
this.ownedRoomsLabel.Name = "ownedRoomsLabel";
this.ownedRoomsLabel.Size = new System.Drawing.Size(110, 15);
this.ownedRoomsLabel.TabIndex = 0;
this.ownedRoomsLabel.Text = "Количество комнат:";
//
// ownedRoomsTextBox
//
this.ownedRoomsTextBox.Location = new System.Drawing.Point(122, 22);
this.ownedRoomsTextBox.Name = "ownedRoomsTextBox";
this.ownedRoomsTextBox.Size = new System.Drawing.Size(194, 23);
this.ownedRoomsTextBox.TabIndex = 1;
//
// ownedAreaLabel
//
this.ownedAreaLabel.AutoSize = true;
this.ownedAreaLabel.Location = new System.Drawing.Point(6, 54);
this.ownedAreaLabel.Name = "ownedAreaLabel";
this.ownedAreaLabel.Size = new System.Drawing.Size(59, 15);
this.ownedAreaLabel.TabIndex = 2;
this.ownedAreaLabel.Text = "Площадь:";
//
// ownedAreaTextBox
//
this.ownedAreaTextBox.Location = new System.Drawing.Point(122, 51);
this.ownedAreaTextBox.Name = "ownedAreaTextBox";
this.ownedAreaTextBox.Size = new System.Drawing.Size(194, 23);
this.ownedAreaTextBox.TabIndex = 3;
//
// ownedFloorLabel
//
this.ownedFloorLabel.AutoSize = true;
this.ownedFloorLabel.Location = new System.Drawing.Point(6, 83);
this.ownedFloorLabel.Name = "ownedFloorLabel";
this.ownedFloorLabel.Size = new System.Drawing.Size(41, 15);
this.ownedFloorLabel.TabIndex = 4;
this.ownedFloorLabel.Text = "Этаж:";
//
// ownedFloorTextBox
//

```

```

this.ownedFloorTextBox.Location = new System.Drawing.Point(122, 80);
this.ownedFloorTextBox.Name = "ownedFloorTextBox";
this.ownedFloorTextBox.Size = new System.Drawing.Size(194, 23);
this.ownedFloorTextBox.TabIndex = 5;
//
// ownedDistrictLabel
//
this.ownedDistrictLabel.AutoSize = true;
this.ownedDistrictLabel.Location = new System.Drawing.Point(6, 112);
this.ownedDistrictLabel.Name = "ownedDistrictLabel";
this.ownedDistrictLabel.Size = new System.Drawing.Size(48, 15);
this.ownedDistrictLabel.TabIndex = 6;
this.ownedDistrictLabel.Text = "Район:";
//
// ownedDistrictTextBox
//
this.ownedDistrictTextBox.Location = new System.Drawing.Point(122, 109);
this.ownedDistrictTextBox.Name = "ownedDistrictTextBox";
this.ownedDistrictTextBox.Size = new System.Drawing.Size(194, 23);
this.ownedDistrictTextBox.TabIndex = 7;
//
// requiredGroupBox
//
this.requiredGroupBox.Controls.Add(this.requiredRoomsLabel);
this.requiredGroupBox.Controls.Add(this.requiredRoomsTextBox);
this.requiredGroupBox.Controls.Add(this.requiredAreaLabel);
this.requiredGroupBox.Controls.Add(this.requiredAreaTextBox);
this.requiredGroupBox.Controls.Add(this.requiredFloorLabel);
this.requiredGroupBox.Controls.Add(this.requiredFloorTextBox);
this.requiredGroupBox.Controls.Add(this.requiredDistrictLabel);
this.requiredGroupBox.Controls.Add(this.requiredDistrictTextBox);
this.requiredGroupBox.Location = new System.Drawing.Point(12, 194);
this.requiredGroupBox.Name = "requiredGroupBox";
this.requiredGroupBox.Size = new System.Drawing.Size(322, 150);
this.requiredGroupBox.TabIndex = 3;
this.requiredGroupBox.TabStop = false;
this.requiredGroupBox.Text = "Требуемая квартира";
//
// requiredRoomsLabel
//
this.requiredRoomsLabel.AutoSize = true;
this.requiredRoomsLabel.Location = new System.Drawing.Point(6, 25);
this.requiredRoomsLabel.Name = "requiredRoomsLabel";
this.requiredRoomsLabel.Size = new System.Drawing.Size(110, 15);
this.requiredRoomsLabel.TabIndex = 0;
this.requiredRoomsLabel.Text = "Количество комнат:";
//
// requiredRoomsTextBox
//
this.requiredRoomsTextBox.Location = new System.Drawing.Point(122, 22);
this.requiredRoomsTextBox.Name = "requiredRoomsTextBox";
this.requiredRoomsTextBox.Size = new System.Drawing.Size(194, 23);
this.requiredRoomsTextBox.TabIndex = 1;
//
// requiredAreaLabel
//
this.requiredAreaLabel.AutoSize = true;
this.requiredAreaLabel.Location = new System.Drawing.Point(6, 54);
this.requiredAreaLabel.Name = "requiredAreaLabel";
this.requiredAreaLabel.Size = new System.Drawing.Size(59, 15);
this.requiredAreaLabel.TabIndex = 2;
this.requiredAreaLabel.Text = "Площадь:";
```

```

//  

// requiredAreaTextBox  

//  

this.requiredAreaTextBox.Location = new System.Drawing.Point(122, 51);  

this.requiredAreaTextBox.Name = "requiredAreaTextBox";  

this.requiredAreaTextBox.Size = new System.Drawing.Size(194, 23);  

this.requiredAreaTextBox.TabIndex = 3;  

//  

// requiredFloorLabel  

//  

this.requiredFloorLabel.AutoSize = true;  

this.requiredFloorLabel.Location = new System.Drawing.Point(6, 83);  

this.requiredFloorLabel.Name = "requiredFloorLabel";  

this.requiredFloorLabel.Size = new System.Drawing.Size(41, 15);  

this.requiredFloorLabel.TabIndex = 4;  

this.requiredFloorLabel.Text = "Этаж:";  

//  

// requiredFloorTextBox  

//  

this.requiredFloorTextBox.Location = new System.Drawing.Point(122, 80);  

this.requiredFloorTextBox.Name = "requiredFloorTextBox";  

this.requiredFloorTextBox.Size = new System.Drawing.Size(194, 23);  

this.requiredFloorTextBox.TabIndex = 5;  

//  

// requiredDistrictLabel  

//  

this.requiredDistrictLabel.AutoSize = true;  

this.requiredDistrictLabel.Location = new System.Drawing.Point(6, 112);  

this.requiredDistrictLabel.Name = "requiredDistrictLabel";  

this.requiredDistrictLabel.Size = new System.Drawing.Size(48, 15);  

this.requiredDistrictLabel.TabIndex = 6;  

this.requiredDistrictLabel.Text = "Район:";  

//  

// requiredDistrictTextBox  

//  

this.requiredDistrictTextBox.Location = new System.Drawing.Point(122,  

109);  

this.requiredDistrictTextBox.Name = "requiredDistrictTextBox";  

this.requiredDistrictTextBox.Size = new System.Drawing.Size(194, 23);  

this.requiredDistrictTextBox.TabIndex = 7;  

//  

// OKButton  

//  

this.OKButton.Location = new System.Drawing.Point(178, 350);  

this.OKButton.Name = "OKButton";  

this.OKButton.Size = new System.Drawing.Size(75, 30);  

this.OKButton.TabIndex = 4;  

this.OKButton.Text = "OK";  

this.OKButton.UseVisualStyleBackColor = true;  

this.OKButton.Click += new System.EventHandler(this.OKButton_Click);  

//  

// CancelButton  

//  

this.CancelButton.Location = new System.Drawing.Point(259, 350);  

this.CancelButton.Name = "CancelButton";  

this.CancelButton.Size = new System.Drawing.Size(75, 30);  

this.CancelButton.TabIndex = 5;  

this.CancelButton.Text = "Cancel";  

this.CancelButton.UseVisualStyleBackColor = true;  

this.CancelButton.Click += new  

System.EventHandler(this.CancelButton_Click);  

//

```

```

// InputForm
//
this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 15F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(346, 392);
this.Controls.Add(this.CancelButton);
this.Controls.Add(this.OKButton);
this.Controls.Add(this.requiredGroupBox);
this.Controls.Add(this.ownedGroupBox);
this.Controls.Add(this.surnameTextBox);
this.Controls.Add(this.surnameLabel);
this.Name = "InputForm";
this.Text = "Ввод заявки";
this.ownedGroupBox.ResumeLayout(false);
this.ownedGroupBox.PerformLayout();
this.requiredGroupBox.ResumeLayout(false);
this.requiredGroupBox.PerformLayout();
this.ResumeLayout(false);
this.PerformLayout();

}

#endifregion

private Label surnameLabel;
private TextBox surnameTextBox;
private GroupBox ownedGroupBox;
private Label ownedRoomsLabel;
private TextBox ownedRoomsTextBox;
private Label ownedAreaLabel;
private TextBox ownedAreaTextBox;
private Label ownedFloorLabel;
private TextBox ownedFloorTextBox;
private Label ownedDistrictLabel;
private TextBox ownedDistrictTextBox;
private GroupBox requiredGroupBox;
private Label requiredRoomsLabel;
private TextBox requiredRoomsTextBox;
private Label requiredAreaLabel;
private TextBox requiredAreaTextBox;
private Label requiredFloorLabel;
private TextBox requiredFloorTextBox;
private Label requiredDistrictLabel;
private TextBox requiredDistrictTextBox;
private Button OKButton;
private Button CancelButton;
}
}

```

Приложение Б

Алгоритм работы программы

Псевдокод алгоритма

1. Главный цикл:

Начало

 Инициализировать HashSet<ExchangeApplication> cardIndex

 Пока форма открыта:

 Ждать события нажатия кнопки

 Если "Ввод заявки":

 Вызвать InputForm("Ввод новой заявки")

 Если DialogResult.OK:

 Получить ExchangeApplication newApp

 Если cardIndex.Add(newApp):

 Вывести "Заявка добавлена"

 Иначе:

 Вывести "Заявка уже существует"

 Если "Поиск варианта":

 Вызвать InputForm("Ввод заявки для поиска")

 Если DialogResult.OK:

 Получить ExchangeApplication newApp

 Найти match в cardIndex по критериям:

 app.Owned.Rooms == newApp.Required.Rooms AND

 app.Required.Rooms == newApp.Owned.Rooms AND

 app.Owned.Floor == newApp.Required.Floor AND

 app.Required.Floor == newApp.Owned.Floor AND

 |app.Owned.Area - newApp.Required.Area| / max(...) <= 0.1

 AND

 |app.Required.Area - newApp.Owned.Area| / max(...) <= 0.1

 Если match найден:

 Вывести детали match

 cardIndex.Remove(match)

 Иначе:

 cardIndex.Add(newApp)

 Вывести "Добавлено в картотеку"

 Если "Вывод картотеки":

 Если cardIndex пуст:

 Вывести "Картотека пуста"

 Иначе:

 Очистить DataGridView

 Добавить колонки (Фамилия, Имеющаяся: Комнаты, ...)

 Для каждой app в cardIndex:

 Добавить строку с данными app

 Если "Выход":

Закрыть форму

Конец

2. Подпрограмма ввода:

Начало

Инициализировать поля TextBox

Ждать нажатия "OK" или "Cancel"

Если "OK":

Попытка:

Парсить числа (rooms, area, floor)

Проверить: не пустые строки, положительные числа

Создать Apartment owned и required

Создать ExchangeApplication

Установить DialogResult.OK

Исключение:

Вывести ошибку (формат, отрицательные)

Если "Cancel":

DialogResult.Cancel

Конец

3. Алгоритм поиска совпадения:

Вход: newApp

Для каждой app в cardIndex:

Если совпадение комнат, этажей и площадь в пределах 10%:

Вернуть app (match)

Вернуть null (добавить newApp)

Приложение В

Схема ресурсов программного средства

Файловая структура проекта:

Проект организован в JetBrains Rider как Windows Forms App:
ApartmentExchange.sln –Файл решения.

ApartmentExchange.csproj –Файл проекта (XML с зависимостями:
TargetFramework=net9.0-windows, UseWindowsForms=true).

bin/ и obj/ – Автогенерируемые (компилированные файлы, не в source control).

Program.cs –Точка входа.

Apartment.cs –Класс квартиры.

ExchangeApplication.cs –Класс заявки.

MainForm.cs и MainForm.Designer.cs –Главная форма и её дизайн.

InputForm.cs и InputForm.Designer.cs –Форма ввода и её дизайн.

Properties/ –Resources.resx (если есть ресурсы, но в проекте пусто),
Settings.settings.

На картинке В.1 изображена схема ресурсов программного средства.

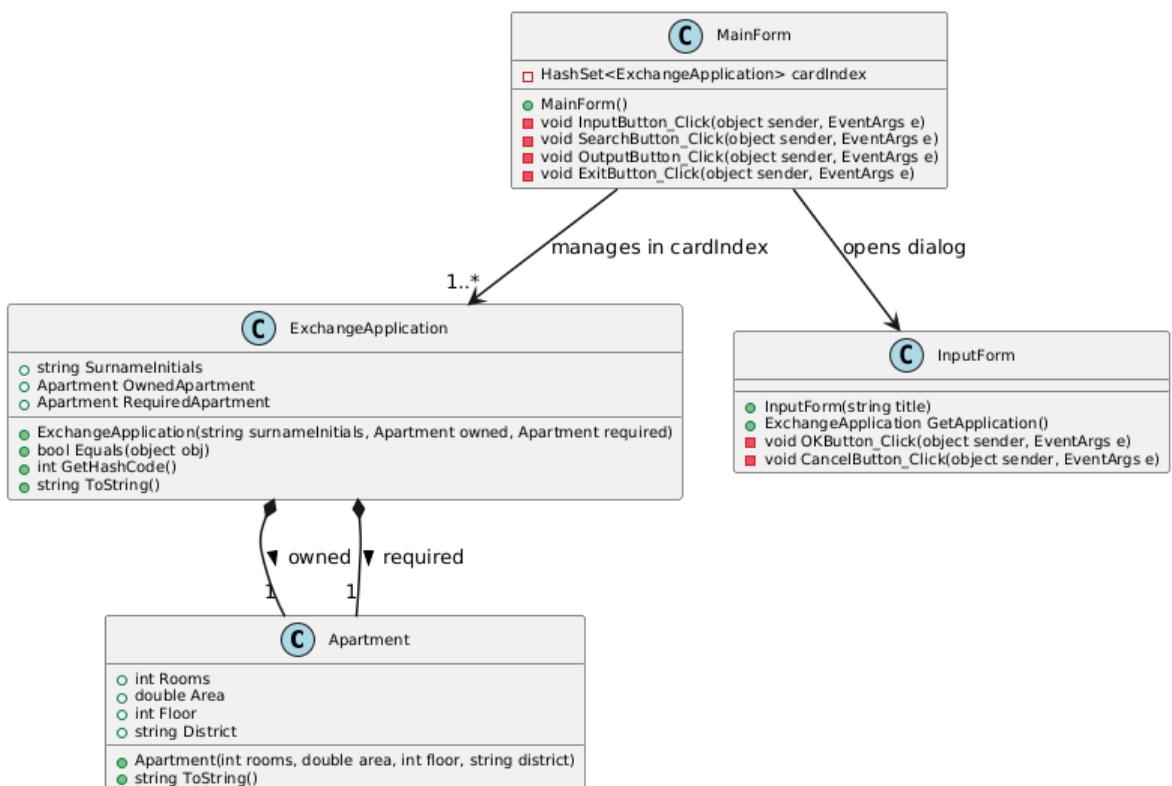


Рисунок В.1 – Схема ресурсов программного средства