



**«Московский государственный технический университет
имени Н.Э. Баумана»
(национальный исследовательский университет)
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

Отчет

по лабораторной работе № 5

Название лабораторной работы: Коллекции. Создание приложений с графическим интерфейсом.

Дисциплина: Алгоритмизация и программирование

Студент гр. ИУ6-24Б

20.04.2024

А.С. Воеводин

(Подпись, дата)

(И.О. Фамилия)

Преподаватель

20.04.2024

О.А. Веселовская

(Подпись, дата)

(И.О. Фамилия)

Цель работы

Ознакомление с механизмом работы коллекций в языке C#, их особенностями и различиями, сравнение их в работе. Также написание программы с графическим интерфейсом, которая реализует функционал из прошлой лабораторной.

Задание

Разработать приложения, выполняющие любое задание лабораторной работы 4, с использованием коллекций и графическим интерфейсом, выполнить тестирование и отладку. В отчете обосновать выбор коллекции, привести диаграмму классов, текст программы и результаты тестирования.

Ход работы:

- Выбор используемой коллекции и обоснование выбора
- Создание дизайна приложения
- Написание кода приложения
- Тестирование
- Диаграмма классов
- Вывод

В данной лабораторной работе будет выполнено графическое представление первой части предыдущей лабораторной работы. Для представления данных в программе была выбрана коллекция списка `List<T>`, так как это самый удобный и быстрый вид представить однотипные данные, размер который может меняться. Теперь представим дизайн, который потом будем реализовывать:

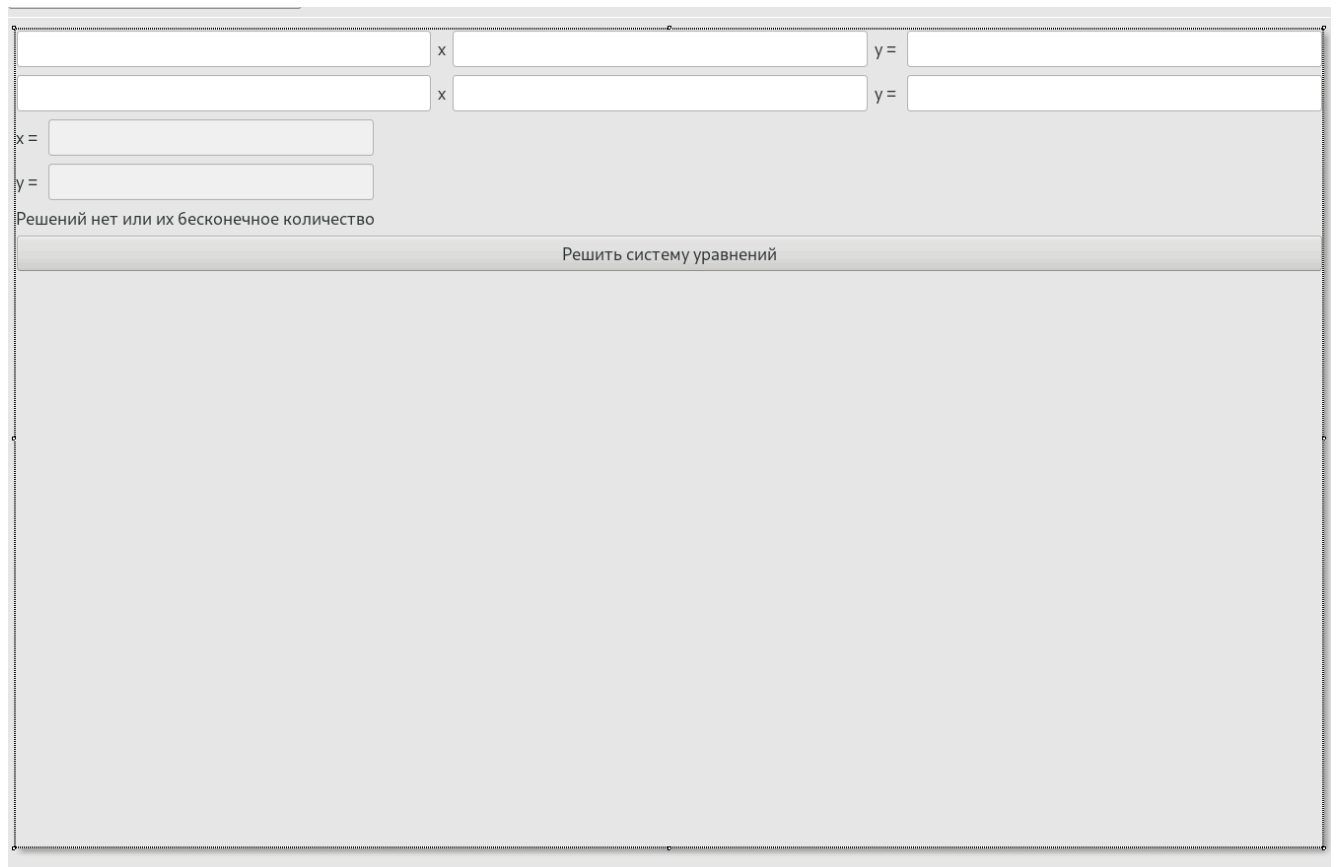


Рисунок 1 – Дизайн приложения

Теперь напомним код программы:

Листинг 1 – MainWindow.cs

```
using System;
using Gtk;
using System.Collections.Generic;

public partial class MainWindow: Gtk.Window
{
    private double Determinant(int m1, int m2, List<List<double>>
arr)
    {
        return (arr[0][m1] * arr[1][m2] - arr[1][m1] *
arr[0][m2]);
    }

    private List<double> solve(double a, double b, double c, double
d, double e, double f)
    {
        List<List<double>> kramer = new List<List<double>>();
```

```

        kramer.Add (new List<double>{ a, b, c });
        kramer.Add (new List<double>{ d, e, f });

        if (Determinant(0, 1, kramer) == 0)
        {
            return null;
        }
        else
        {
            return new List<double>{Determinant(2, 1, kramer) /
Determinant(0, 1, kramer), Determinant(0, 2, kramer) /
Determinant(0, 1, kramer)};
        }
    }

    public MainWindow () : base (Gtk.WindowType.Toplevel)
    {
        Build ();
        label1.Visible = false;
    }

    protected void OnDeleteEvent (object sender, DeleteEventArgs a)
    {
        Application.Quit ();
        a.RetVal = true;
    }

    protected void OnButton2Clicked (object sender, EventArgs e)
    {
        try
        {
            List<double> solution =
solve(double.Parse(entry_x1.Text), double.Parse(entry_y1.Text),
        double.Parse(entry_a.Text),
double.Parse(entry_x2.Text),

```

```

        double.Parse(entry_y2.Text),
double.Parse(entry_b.Text));
        if (solution != null)
        {
            entry4.Text = solution[0].ToString();
            entry5.Text = solution[1].ToString();
            label11.Visible = false;
            entry4.Visible = true;
            entry5.Visible = true;
            label20.Visible = true;
            label21.Visible = true;
        }
        else
        {
            label11.Visible = true;
            entry4.Visible = false;
            entry5.Visible = false;
            label20.Visible = false;
            label21.Visible = false;
        }
    }
    catch
    {
        MessageBox md = new MessageBox (null,
DialogFlags.DestroyWithParent, MessageType.Error,
        ButtonsType.Ok, "Ошибка: введены нечисловые
данные. Пожалуйста, попробуйте ещё раз.");
        md.Run ();
        md.Destroy();

    }
}
}
}

```

Указывать код основного файла не имеет смысла, так как он не изменялся после создания проекта. Теперь протестируем получившееся приложение:

MainWindow

1	x	2	y =	3
6	x	7	y =	8

x =

y =

Решить систему уравнений

Рисунок 2 – Корректные данные 1

MainWindow

1	x	3	y =	4
76	x	87	y =	48

x =

y =

Решить систему уравнений

Рисунок 3 – Корректные данные 2

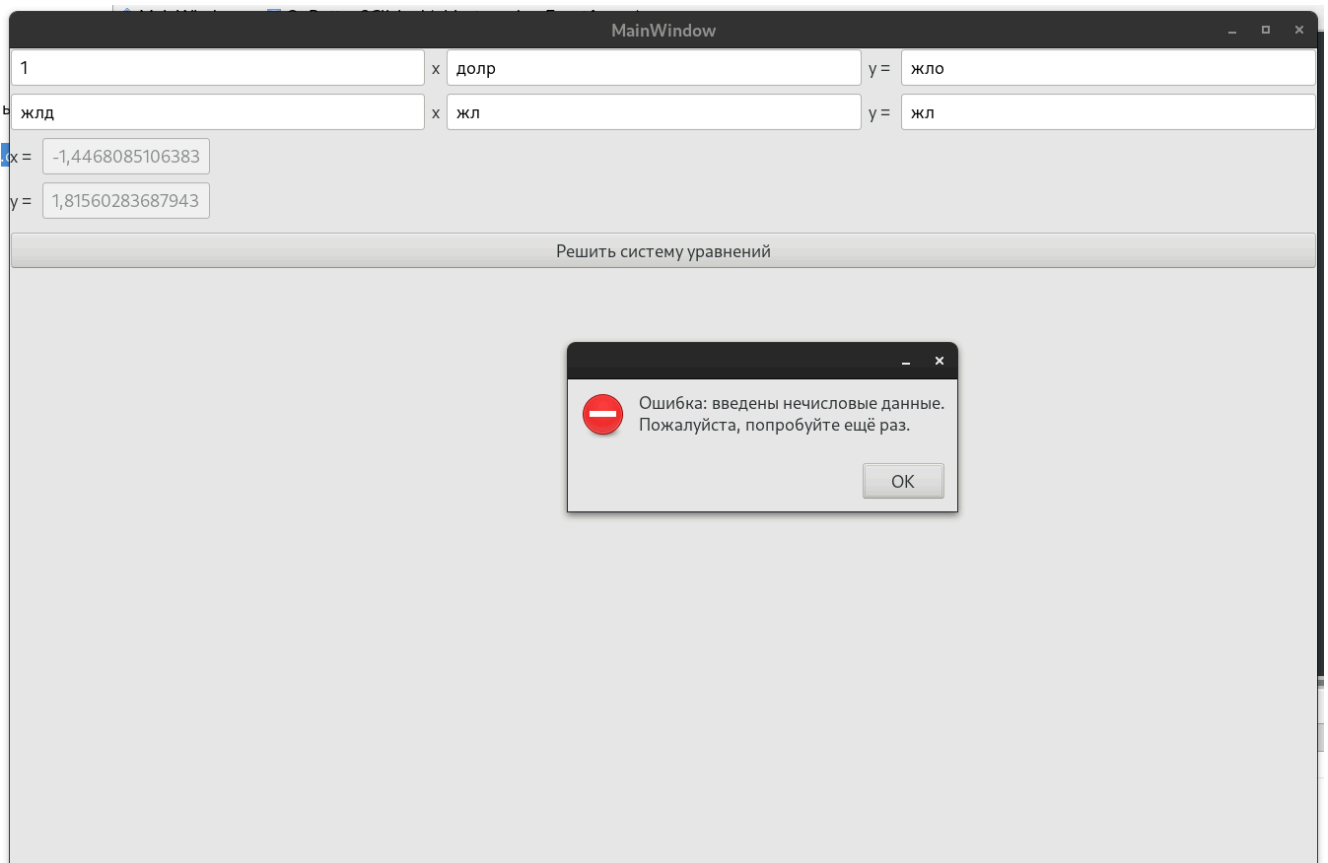


Рисунок 4 – Нечисловые тестовые данные

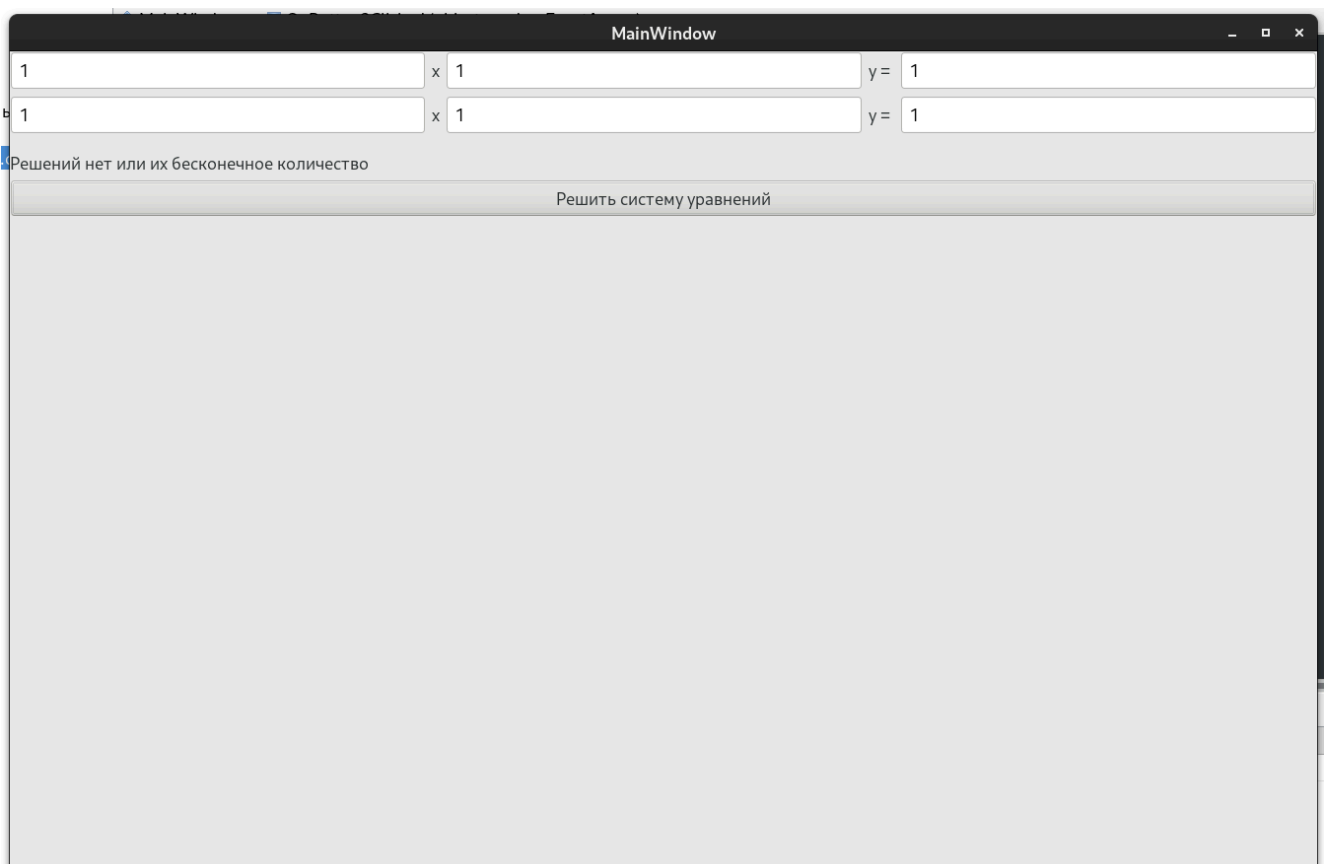


Рисунок 5 – Пример отсутствия решений

Несложно проверить и решить систему уравнений чтобы проверить корни. Теперь изобразим диаграмму классов:

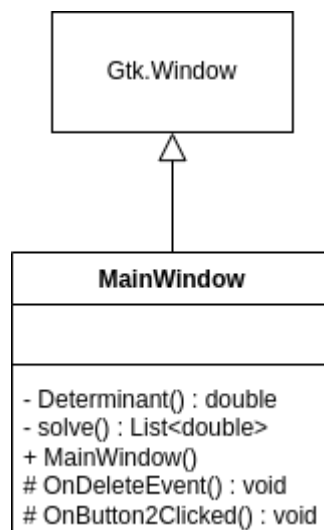


Рисунок 6 – Диаграмма классов

Вывод

В ходе лабораторной работы были получены навыки работы с коллекциями на языке C#, а также разработка графического интерфейса с помощью библиотеки GTK# в профессиональной среде разработки Monodevelop и встроенного дизайнера.