**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

**Centrul de Excelenţă în Informatică şi Tehnologii Informaţionale**

**RAPORT**

**LA PRACTICA DE INSTRUIRE**

**SPECIALITATEA "Programare şi analiza produselor program"**

**TEMA : "Terenuri Cultivate"**

A elaborat elevul

Vicev Alexandr, P1723r

(nume, prenume, grupa)

Conducătorul practicii

Şarapanovscaia Irina

(nume, prenume)

Chişinău 2019

Содержание

[**Введение** 3](#_Toc11756762)

[**Условие задания** 4](#_Toc11756763)

[**Входные данные**: 6](#_Toc11756764)

[**Общее описание приложения** 7](#_Toc11756765)

[**Детальное описание функционала приложения** 8](#_Toc11756766)

[**Код программы** 29](#_Toc11756767)

[**Основная форма** 29](#_Toc11756768)

[**Форма справки** 57](#_Toc11756769)

[**Форма загрузки** 60](#_Toc11756770)

[**Вывод** 61](#_Toc11756771)

[**Библиография** 62](#_Toc11756772)

# **Введение**

Практика это один из важных шагов в становление профессионалом в своей профессии. Моя практика проходила в колледже информатики CEITI с 3 июня по 21 июня. Мы получили свои задания, которые помогли нам усовершенствовать свои знания в области программирования. Нашу основную программу мы разрабатывали на VisualStudio в С# Forms.

Благодаря данной практике я получил бесценный опыт, который поможет мне в будущем, а также осталось много положительных эмоций. Отдельно хочу поблагодарить колледж CEITI за предоставленную возможность, а также большое спасибо нашей учительнице по программированию – Шарапановской Ирине Александровне.

Все визуальные элементы моей программы, являются моими собственными разработками, а различные иконки и .PNG изображения взяты из открытых источников, которые позволяют использовать свои труды.

Далее я хочу представить свою работу за эти три недели.

# **Условие задания**

**Обрабатываемые земли.** В прямоугольной системе координат **XOY** нарисованы прямоугольники, моделирующие участки земли, засаженные культурами одного и того же вида. Стороны участков параллельны осям координат и у вершин прямоугольников целочисленные координаты. Участки могут иметь общие точки только по сторонам. Каждому виду культуры поставлен определённый цвет с кодом **Ср** .(см. рис. 1 и рис. 2)

Подробная информация о рассматриваемых прямоугольниках записана в текстовый файл **Teren.in**, который содержит в первой строке натуральное число **n**, представляющее собой число прямоугольников. Каждая из следующих **n** строк данного файла содержит по 5 целых чисел, отделённых одним пробелом: **X1, Y1, X2, Y2, CP**.

**(X1, Y1)** и **(X2, Y2)** являются координатами двух противоположных вершин прямоугольника, а **Ср** содержит код цвета прямоугольника.

Разработать приложение, которое с помощью меню и подпрограмм выполняет по выбору пользователя следующие действия:

1. Дописывает в текстовый файл **Teren.in**, прямоугольник с характеристиками, описанными ранее.

2. Удаляет из текстового файла **Teren.in**, прямоугольник, координаты двух противоположных сторон которого вводятся с клавиатуры.

3. Строит прямоугольник минимальной площади со сторонами параллельными осям координат, содержащий все участки, описанные в входном файле.

4. Находит общую площадь

5. Определяет культуру (цвет), занимающий наибольшую площадь из всех участков.

6. Создаёт файл **ColColumn.txt** копируя те строки входного файла **Teren.in**, которые соответствуют коду цвета (Сp), введённого с клавиатуры.

7. Перераспределяет строки входного файла в порядке возрастания кодов цвета культур.

8. **Решите задачу**. Мариука впервые летит на самолёте. С самолёта она видит земли, на которых выращиваются культуры, описанные ранее. Будучи любопытной, маленькая девочка хотела бы определить площадь самого большого прямоугольника, занятого культурами одного и того же цвета. Разработайте подпрограмму, которая поможет Мариуке решить свою задачу.

**Входные данные**. Файл **Teren.in**, описанный ранее.

**Выходные данные**. Выходной файл **Teren.out** содержит одно единственное значение – площадь самого большого прямоугольника, занятого культурами одного и того же цвета.

Таблица . Пример заполнения полей с культурами.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **9** |  |  |  |  |  |  |  | **9** |  |  |  |  |  |  |  |  |  |  |  |
| **8** |  |  |  |  |  |  |  | **8** | **3** | **3** | **3** | **3** | **3** | **3** | **3** | **3** | **3** | **3** |  |
| **7** |  |  |  |  |  |  |  | **7** | **3** | **3** | **3** | **3** | **3** | **3** | **3** | **3** | **3** | **3** |  |
| **6** |  |  |  |  |  |  |  | **6** | **3** | **3** | **3** | **3** | **3** | **3** | **2** | **2** | **2** |  |  |
| **5** |  | **1** | **1** | **1** | **1** |  |  | **5** | **2** | **2** | **2** | **2** | **2** | **2** | **2** | **2** | **2** |  |  |
| **4** |  | **1** | **1** | **1** | **1** |  |  | **4** | **2** | **2** | **2** | **2** | **2** | **2** | **2** | **2** | **2** |  |  |
| **3** | **2** | **2** | **2** | **2** | **2** | **2** |  | **3** | **2** | **2** | **2** | **2** | **2** | **2** | **2** | **2** | **2** |  |  |
| **2** |  | **4** | **4** | **1** | **1** |  |  | **2** | **1** | **1** | **1** | **1** | **5** | **5** | **2** | **2** | **2** |  |  |
| **1** |  | **4** | **4** | **1** | **1** |  |  | **1** | **1** | **1** | **1** | **1** | **5** | **5** |  | **2** |  |  |  |
|  |  |  |  |  |  |  |  |  | **1** | **1** | **1** | **1** | **5** | **5** |  | **2** |  |  |  |
| **(0,0)** | **1** | **2** | **3** | **4** | **5** | **6** |  | **(0,0)** | **1 2 3 4 5 6 7 8 9 10** | | | | | | | | | | | |
|  | **Fig.1 (Exemplul 1)** | | | | | | |  | **Fig. 2 (Exemplul 2)** | | | | | | | | | | | |

Таблица . Контрольные данные

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ограничения** | **Пример** (для рисунков 1 и 2) | | | |
| **1** | | **2** | |
| 1 **<= *n* <= 2500 1 <= Cp <= 100** **0<=X1,Y1, X2,Y2<=1000000000  Максимальное время выполнения**: **1.5 секунды** | **teren.in** | **teren.out** | **teren.in** | **teren.out** |
| **5  1 1 3 3 4  3 1 5 3 1  1 4 3 6 1  3 4 5 6 1  0 3 6 4 2** | **8** | **7  0 0 4 3 1  6 2 9 7 2  6 7 10 9 3  4 0 6 3 5  0 6 6 9 3  0 3 6 6 2  7 0 8 2 2** | **27** |

# **Входные данные**:

Для примера работы моей программы, я взял второй пример, так как он является более сложным:

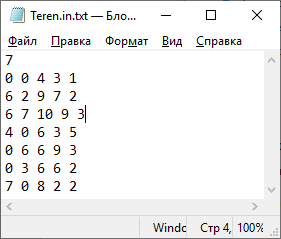


Рисунок . Входной файл Teren.in.

Второй пример.

# **Общее описание приложения**

Программа «Cultivated Land» является многофункциональным помощником для работы с участками территорий, которые засеяны различными культурами (в программе представленные различными цифрами и соответствующими им цветам).

«Cultivated Land» представляет из себя лишь тестовое задание, которое не предназначено для профессионального использования.

Основной функционал предоставляет пользователю: просматривать участки, удалять определённые участки, добавлять новые, очищать все поля одним нажатием, а также сохранять файл с координатами определённых участков на неограниченный срок.

Дополнительный функционал программы предоставляет пользователю: найти координаты минимального прямоугольника, в который вмешаются все засеянные территории, найти общую площадь, занятую всему участками, определить культуры, территории которой занимают наибольшую площадь, создать новый файл, в котором будет введена информация только оп определённой культуре, сортировка культур по их номеру цвета, а также поиск самого большого цельного участка, который засеян одинаковой культурой.

Для удобного использования также есть справка на трёх языках: румынский, русский и английский.

# **Детальное описание функционала приложения**

Запуск приложения начинается со стартового логотипа:



Рисунок . Стартовый логотип

Реализация очень проста: создание новой формы, на которой помещён PictureBox и изменён цвет формы, также добавлен таймер, код формы:

public Loading()

{

InitializeComponent();

timer1.Start();

}

private void Timer1\_Tick(object sender, EventArgs e)

{

this.Opacity += .03;

if(this.Opacity==1)

{

timer1.Stop();

}

}

Также необходимо изменить код в Program.cs на следующее:

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Loading first = new Loading();

DateTime end = DateTime.Now + TimeSpan.FromSeconds(5);

first.Show();

while(end>DateTime.Now)

{

Application.DoEvents();

}

first.Close();

first.Dispose();

Application.Run(new MainForm());

}

}

После загрузки перед нами появляется рабочая область программы:

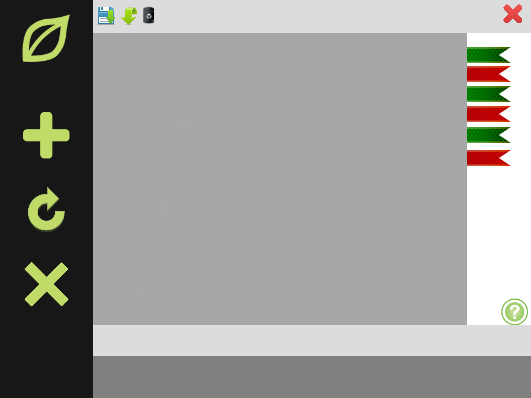


Рисунок . Рабочая область программы

Для начала работы с программой, необходимо обновить данные, которые считываются из файла Teren.in.txt и сохраняются в DataGridView1, а далее DataGridView2 выводит всю информацию на экран в виде прямоугольной системы координат XOY:

Считывание информации из файла происходит при нажатии боковой клавиши в левом меню, которая находится между добавлением (+) и удалением (х):

private DataTable ReadFromFile(string name)

{

DataTable dt = new DataTable();

int nn = 0;

try

{

using (StreamReader sr = new StreamReader(name))

{

string line;

string[] ar;

line = sr.ReadLine();

num = Convert.ToInt32(line);

while ((line = sr.ReadLine()) != null)

{

ar = line.Split(' ');

for (int i = 0; i < ar.Length; i++)

if (nn == 0) dt.Columns.Add(Convert.ToString(i + 1));

dt.Rows.Add(ar);

nn++;

}

sr.Close();

}

}

catch (Exception ex)

{

Console.WriteLine("Файл не может быть прочитан:");

Console.WriteLine(ex.Message);

}

return dt;

}

После считывания информации из файла, она сразу сохраняется в DataGridView1 и DataGridView2 выводит её на экран:

private DataTable VisualDataGrid(DataGridView dataGridView)

{

DataTable visualDataTable = new DataTable();

int MaxElement = MaxElem(dataGridView1);

for (int i = 0; i <= MaxElement; i++)

visualDataTable.Columns.Add(Convert.ToString(i));

for (int j = 0; j <= MaxElement; j++)

visualDataTable.Rows.Add("");

for (int i = 0; i <= MaxElement; i++)

for (int j = 0; j <= MaxElement; j++)

visualDataTable.Rows[j][i] = 0;

for (int i = 0; i < num; i++)

for (int j = Convert.ToInt32(dataGridView.Rows[i].Cells[1].Value); j < Convert.ToInt32(dataGridView.Rows[i].Cells[3].Value); j++)

for (int k = Convert.ToInt32(dataGridView.Rows[i].Cells[0].Value); k < Convert.ToInt32(dataGridView.Rows[i].Cells[2].Value); k++)

visualDataTable.Rows[j][k] = Convert.ToInt32(dataGridView.Rows[i].Cells[4].Value);

return visualDataTable;

}

Также, чтобы DataGridView2 отображал цвета, не мог быть изменён через поля, отсортирован, то есть, чтобы пользователь никак не мог нарушить его работу, добавлена защита:

1. Получение максимального значения из DataGridView1, для правильного построения прямоугольной системы координат:

private int MaxElem(DataGridView dataGridView)

{

int maxE = 0;

for (int j = 0; j < dataGridView.Rows.Count; j++)

{

int Tx2 = Convert.ToInt32(dataGridView.Rows[j].Cells[2].Value);

int Ty2 = Convert.ToInt32(dataGridView.Rows[j].Cells[3].Value);

if (Tx2 > maxE) maxE = Tx2;

if (Ty2 > maxE) maxE = Ty2;

}

return ++maxE;

}

1. Цвета, соответствующие номеру культуры:

private void ColorGridView()

{

int MaxElement = MaxElem(dataGridView1);

for (int i = 0; i < MaxElement; i++)

for (int j = 0; j < MaxElement; j++)

{

if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == -1)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.Black;

else if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 0)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.White;

else if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 1)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.Blue;

else if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 2)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.Green;

else if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 3)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.Red;

else if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 4)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.Pink;

else if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 5)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.Purple;

else if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 6)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.Silver;

else if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 7)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.Yellow;

else if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 8)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.Wheat;

else if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 9)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.Tan;

}

}

1. Запрет на выделение ячеек, запрет на изменение размеров столбцов/строк, запрет на сортировку по столбцам, запрет на добавление/удаление/изменение элементов, цифры в боковой стороне.

//Запрет на выделение ячеек

private void DataGridView2\_SelectionChanged(object sender, EventArgs e)

{

dataGridView2.ClearSelection();

}

//Запрет на изменение размеров столбцов/строк

private void NoSizebale(DataGridView dataGridView)

{

dataGridView.AllowUserToResizeColumns = false;

dataGridView.AllowUserToResizeRows = false;

}

//Запрет на сортировку по столбцу

private void NoSortable(DataGridView dataGridView)

{

int MaxElement = MaxElem(dataGridView1);

for (int i = 0; i <= MaxElement; i++)

dataGridView.Columns[i].SortMode = DataGridViewColumnSortMode.NotSortable;

}

//Запрет на добавление/изменение/удаление через DataGridView

private void ForbideRead(DataGridView dg)

{

int MaxElement = MaxElem(dataGridView1);

for (int i = 0; i < MaxElement; i++)

for (int j = 0; j < MaxElement; j++)

dg.Rows[i].Cells[j].ReadOnly = true;

}

//Вывол боковых колонок DataGridView2=====

private void FillRecordNo(DataGridView dg)

{

for (int i = 0; i < dg.Rows.Count; i++)

{

dg.Rows[i].HeaderCell.Style.Alignment = DataGridViewContentAlignment.MiddleCenter;

dg.Rows[i].HeaderCell.Value = (i).ToString();

}

}

Все это программа выполняет во время каждого обновления рабочей области DataGridView2. После всех этих действий, на экран выводится следующее:

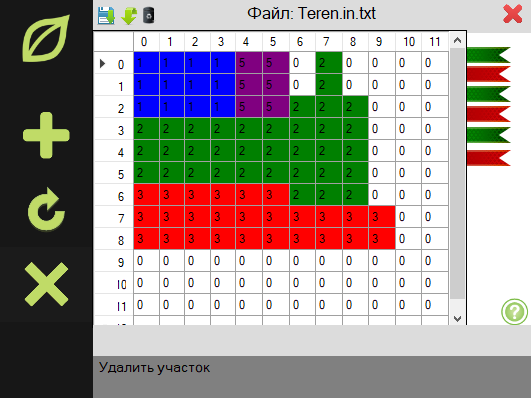


Рисунок . Рабочая область. Обновление значений.

Теперь попробуем добавить новое значение в таблицу, делается это нажатием на кнопку плюс в левой боковой панели, далее вводятся координаты по X1, Y1, X2, Y2 и цифра, то есть культура:

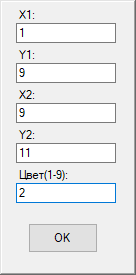


Рисунок . Новые данные для добавления.



Рисунок . Новые данные в DataGridView2

Теперь рассмотрим, как этот процесс происходит на уровне кода:

1. Вызывается диалоговое окно, которое запрашивает у пользователя пять новых значений:

static Tuple<string, string, string, string, string> CreateDlgBox2()

{

Form Form2 = new Form

{

ControlBox = false,

Text = String.Empty,

FormBorderStyle = FormBorderStyle.FixedDialog

};

TextBox TextBox1 = new TextBox();

TextBox TextBox2 = new TextBox();

TextBox TextBox3 = new TextBox();

TextBox TextBox4 = new TextBox();

TextBox TextBox5 = new TextBox();

Label label1 = new Label();

Label label2 = new Label();

Label label3 = new Label();

Label label4 = new Label();

Label label5 = new Label();

Button btn1 = new Button();

string x1, x2;

string y1, y2;

string Cp;

TextBox1.Parent = Form2;

TextBox2.Parent = Form2;

TextBox3.Parent = Form2;

TextBox4.Parent = Form2;

TextBox5.Parent = Form2;

label1.Parent = Form2;

label2.Parent = Form2;

label3.Parent = Form2;

label4.Parent = Form2;

label5.Parent = Form2;

btn1.Parent = Form2;

btn1.Text = "OK";

label1.Text = "X1:";

label2.Text = "Y1:";

label3.Text = "X2:";

label4.Text = "Y2:";

label5.Text = "Цвет(1-9):";

label1.Bounds = new Rectangle(13, 5, 100, 20);

TextBox1.Bounds = new Rectangle(13, 20, 100, 20);

label2.Bounds = new Rectangle(13, 45, 100, 20);

TextBox2.Bounds = new Rectangle(13, 60, 100, 20);

label3.Bounds = new Rectangle(13, 85, 100, 20);

TextBox3.Bounds = new Rectangle(13, 100, 100, 20);

label4.Bounds = new Rectangle(13, 125, 100, 20);

TextBox4.Bounds = new Rectangle(13, 140, 100, 20);

label5.Bounds = new Rectangle(13, 165, 100, 20);

TextBox5.Bounds = new Rectangle(13, 180, 100, 20);

btn1.Bounds = new Rectangle(25, 220, 70, 30);

Form2.Bounds = new Rectangle(0, 0, 70, 275);

Form2.StartPosition = FormStartPosition.CenterParent;

btn1.DialogResult = DialogResult.OK;

Form2.ShowDialog();

x1 = TextBox1.Text;

y1 = TextBox2.Text;

x2 = TextBox3.Text;

y2 = TextBox4.Text;

Cp = TextBox5.Text;

Form2.AcceptButton = btn1;

return Tuple.Create<string, string, string, string, string>(x1, y1, x2, y2, Cp);

}

1. После этого идёт добавление:

private void AddRow(string x1, string y1, string x2, string y2, string Cp, DataGridView dataGridView)

{

int MaxElement = MaxElem(dataGridView1);

try

{

if (x1 == "" && y1 == "" && x2 == "" && y2 == "" && Cp == "")

throw new Exception("Вы ничего не ввели!");

if (x1 == "" || y1 == "" || x2 == "" || y2 == "" || Cp == "")

throw new Exception("Одно из значений не заполнено");

if (x1 == x2 || y1 == y2)

throw new Exception("Значения x1, x2 или y1=y2 равны между собой!");

int LocalX1 = Convert.ToInt32(x1);

int LocalY1 = Convert.ToInt32(y1);

int LocalX2 = Convert.ToInt32(x2);

int LocalY2 = Convert.ToInt32(y2);

int LocalCp = Convert.ToInt32(Cp);

bool WasFound = true;

if (LocalX1 > LocalX2 || LocalY1 > LocalY2)

{

if (LocalX1 > LocalX2)

{

int temp;

temp = LocalX1;

LocalX1 = LocalX2;

LocalX2 = temp;

}

if (LocalY1 > LocalY2)

{

int temp;

temp = LocalY1;

LocalY1 = LocalY2;

LocalY2 = temp;

}

MessageBox.Show("Координаты были введены неверно.\nПрограмма всё исправила!");

}

if (LocalX1 < 0 || LocalX1 > 1000000000 ||

LocalY1 < 0 || LocalY1 > 1000000000 ||

LocalX2 < 0 || LocalX2 > 1000000000 ||

LocalY2 < 0 || LocalY2 > 1000000000)

throw new Exception("Одно из значений не входит в промежуток (0 - 1000000000)");

if (LocalCp > 100 || LocalCp < 0)

throw new Exception("Значение цвета не входит в промежуток (1 - 100)");

bool ErrorZeroPoint = Check();

if (!(LocalX1 > MaxElement && LocalX2 > MaxElement && LocalY1 > MaxElement && LocalY2 > MaxElement))

{

for (int i = 0; i < MaxElement && WasFound; i++)

{

for (int j = 0; j < MaxElement && WasFound; j++)

{

if (LocalX1 <= j && j < LocalX2 && LocalY1 <= i && i < LocalY2)

{

if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 0)

WasFound = true;

else WasFound = false;

}

}

}

}

if (WasFound)

{

DataTable dt = (DataTable)(dataGridView.DataSource);

DataRow newRow = dt.NewRow();

num++;

newRow[0] = Convert.ToString(LocalX1);

newRow[1] = Convert.ToString(LocalY1);

newRow[2] = Convert.ToString(LocalX2);

newRow[3] = Convert.ToString(LocalY2);

newRow[4] = Cp;

dt.Rows.InsertAt(newRow, dt.Rows.Count);

if (ErrorZeroPoint)

{

dataGridView.Rows.RemoveAt(0);

num--;

}

}

else throw new Exception("Прямоугольники пересекаются!");

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка: {ex.Message}\nЗначения не были добавлены.");

}

1. Хотя, значения добавились в DataGridView2 и мы их видим, но в файл эти значения ещё не сохранились, для этого необходимо нажать кнопку сохранения в верхней панели:

private void SaveToFile(DataGridView dg, String fileName)

{

bool flag = true;

if (num == 0)

flag = CreateDlgBox4();

if (flag)

{

FileStream fs = new FileStream(@fileName, FileMode.Create);

StreamWriter streamWriter = new StreamWriter(fs);

streamWriter.Write(num);

streamWriter.WriteLine();

try

{

for (int j = 0; j < dg.Rows.Count; j++)

{

for (int i = 0; i < dg.Rows[j].Cells.Count; i++)

{

streamWriter.Write(dg.Rows[j].Cells[i].Value);

if (i < dg.Rows[j].Cells.Count - 1 && j < dg.Rows.Count - 1) streamWriter.Write(" ");

}

if (j < dg.Rows.Count - 1) streamWriter.WriteLine();

}

streamWriter.Close();

fs.Close();

label2.Visible = true;

label2.Text = "Файл успешно сохранен";

}

catch

{

MessageBox.Show("Ошибка при сохранении файла!");

}

}

}

Процесс удаления необходимой территории очень схож с процессом добавления: также запрашиваются данные прямоугольника у пользователя, после чего, происходит проверка и если прямоугольник был найден, он будет удалён, в противном случае, программа сообщит, что прямоугольник не был найден. После всех этих действий необходимо сохранение:

private void DelRow(string x1, string y1, string x2, string y2, DataGridView dataGridView)

{

int MaxElement = MaxElem(dataGridView);

try

{

if (x1 == "" && y1 == "" && x2 == "" && y2 == "")

throw new Exception("Вы ничего не ввели!");

if (x1 == "" || y1 == "" || x2 == "" || y2 == "")

throw new Exception("Одно из значений не заполнено");

if (x1 == x2 || y1 == y2)

throw new Exception("Значения x1, x2 или y1=y2 равны между собой!");

int LocalX1 = Convert.ToInt32(x1);

int LocalY1 = Convert.ToInt32(y1);

int LocalX2 = Convert.ToInt32(x2);

int LocalY2 = Convert.ToInt32(y2);

bool WasFound = true;

if (LocalX1 > LocalX2 || LocalY1 > LocalY2)

{

if (LocalX1 > LocalX2)

{

int temp;

temp = LocalX1;

LocalX1 = LocalX2;

LocalX2 = temp;

}

if (LocalY1 > LocalY2)

{

int temp;

temp = LocalY1;

LocalY1 = LocalY2;

LocalY2 = temp;

}

MessageBox.Show("Координаты были введены неверно.\nПрограмма всё исправила!");

}

if (LocalX1 < 0 || LocalX1 > 1000000000 ||

LocalY1 < 0 || LocalY1 > 1000000000 ||

LocalX2 < 0 || LocalX2 > 1000000000 ||

LocalY2 < 0 || LocalY2 > 1000000000)

throw new Exception("Одно из значений не входит в промежуток (0 - 1000000000)");

for (int i = 0; i < MaxElement && WasFound; i++)

{

for (int j = 0; j < MaxElement && WasFound; j++)

{

if (Convert.ToInt32(dataGridView.Rows[i].Cells[0].Value) == LocalX1

&& Convert.ToInt32(dataGridView.Rows[i].Cells[1].Value) == LocalY1

&& Convert.ToInt32(dataGridView.Rows[i].Cells[2].Value) == LocalX2

&& Convert.ToInt32(dataGridView.Rows[i].Cells[3].Value) == LocalY2)

{

dataGridView.Rows.RemoveAt(i);

WasFound = false;

num--;

}

}

}

if (WasFound) throw new Exception("Прямоугольник по вашим данным не найден");

else MessageBox.Show("Прямоугольник был удалён!");

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка: {ex.Message}\nЗначения не были добавлены.");

}

}

Теперь разберёмся с дополнительными функциями, для их вызова необходимо нажать на флажки, которые выдвигаются с правой стороны от DataGridView2, все результаты выполнения программы выводятся на нижнею панель (на самой нижней панели находятся подсказки, которые появляются при наведении на кнопки):

1. Строит прямоугольник минимальной площади со сторонами параллельными осям координат, содержащий все участки, описанные в входном файле:

static Tuple<int, int, int, int> MinimalPream(DataGridView dataGridView, int numOf)

{

int x1 = 1000000000, y1 = 1000000000, x2 = 0, y2 = 0;

for (int i = 0; i < numOf; i++)

{

if (Convert.ToInt32(dataGridView.Rows[i].Cells[0].Value) < x1) x1 = Convert.ToInt32(dataGridView.Rows[i].Cells[0].Value);

if (Convert.ToInt32(dataGridView.Rows[i].Cells[1].Value) < y1) y1 = Convert.ToInt32(dataGridView.Rows[i].Cells[1].Value);

if (Convert.ToInt32(dataGridView.Rows[i].Cells[2].Value) > x2) x2 = Convert.ToInt32(dataGridView.Rows[i].Cells[2].Value);

if (Convert.ToInt32(dataGridView.Rows[i].Cells[3].Value) > y2) y2 = Convert.ToInt32(dataGridView.Rows[i].Cells[3].Value);

}

return Tuple.Create<int, int, int, int>(x1, y1, x2, y2);

}



Рисунок . Результаты выполнения первой программы.

1. Находит общую площадь, занятую всеми участками.

private int TotalArea(DataGridView dataGridView)

{

int Area = 0;

int MaxElement = MaxElem(dataGridView1);

for (int i = 0; i < MaxElement; i++)

{

for (int j = 0; j < MaxElement; j++)

{

if (Convert.ToInt32(dataGridView.Rows[i].Cells[j].Value) != 0) Area++;

}

}

return Area;

}

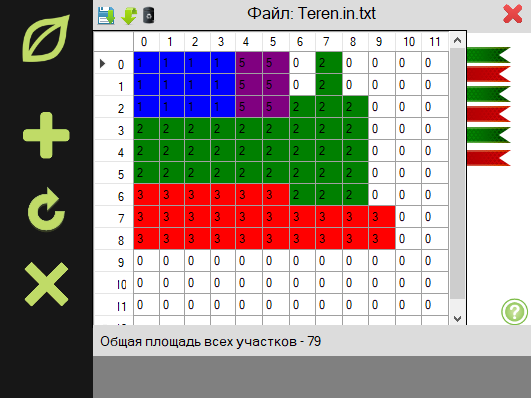


Рисунок . Результат выполнения второй программы.

1. Определяет культуру (цвет), занимающий наибольшую площадь из всех участков.

private int ColorArea(DataGridView dataGridView)

{

int[] Area = new int[100];

int max = 0;

int maxValue = 0;

int MaxElement = MaxElem(dataGridView1);

for (int i = 0; i < 100; i++)

Area[i] = 0;

for (int i = 0; i < MaxElement; i++)

for (int j = 0; j < MaxElement; j++)

{

if (Convert.ToInt32(dataGridView.Rows[i].Cells[j].Value) != 0)

Area[Convert.ToInt32(dataGridView.Rows[i].Cells[j].Value)]++;

}

for (int i = 0; i < 100; i++)

if (Area[i] > maxValue)

{

max = i;

maxValue = Area[i];

}

return max;

}

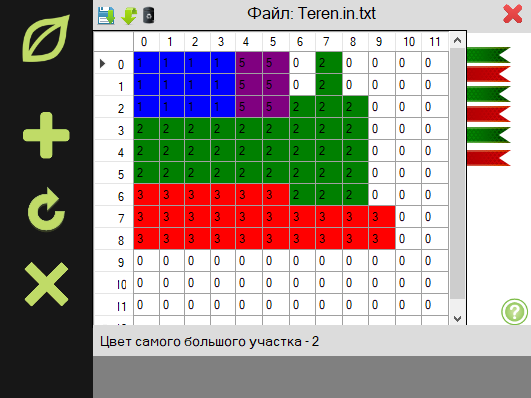


Рисунок . Результат выполнения третей программы.

1. Создаёт файл ColColumn.txt копируя те строки входного файла Teren.in, которые соответствуют коду цвета (Ср), введённого с клавиатуры.

* Делаем запрос через новую форму и узнаем цвет, который он хочет вывести в отдельный файл.

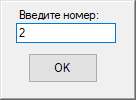


Рисунок . Запрос к пользователю.

* Ищем подходящие поля в DataGridView1 и записываем результат в ColColumn.txt

public static void CreateFile(string fileName, DataGridView dg, string numCp)

{

FileStream fs = new FileStream(@fileName, FileMode.Create);

StreamWriter streamWriter = new StreamWriter(fs);

bool WasWritte;

bool WasFound = false;

try

{

if (numCp == "") throw new Exception("Вы ничего не ввели.");

if (Convert.ToInt32(numCp) > 100 || Convert.ToInt32(numCp) < 0) throw new Exception("Значение цвета не входит в промежуток (1 - 100)");

for (int j = 0; j < dg.Rows.Count; j++)

{

WasWritte = false;

for (int i = 0; i < dg.Rows[j].Cells.Count; i++)

{

if (Convert.ToString(dg.Rows[j].Cells[4].Value) == numCp)

{

streamWriter.Write(dg.Rows[j].Cells[i].Value);

if (i < dg.Rows[j].Cells.Count - 1 && j < dg.Rows.Count - 1) streamWriter.Write(" ");

{

WasWritte = true;

WasFound = true;

}

}

}

if (WasWritte)

if (j < dg.Rows.Count - 1) streamWriter.WriteLine();

}

streamWriter.Close();

fs.Close();

if (WasFound)

MessageBox.Show("Файл успешно сохранен");

else

MessageBox.Show("Участка такого цвета не найдено.\nФайл пуст!");

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка: {ex.Message}\nЗначения не были сохранены.");

}

}

* Результат выполнения в файле ColColumn.txt

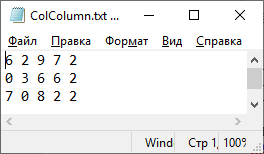


Рисунок . Результат выполнения программы

1. Перераспределяет строки входного файла в порядке возрастания кодов цвета культур.

* Сортируем DataGridView1

private void Sort(DataGridView dataGridView)

{

DataGridViewColumn newColumn = dataGridView.Columns[4];

dataGridView.Sort(newColumn, ListSortDirection.Ascending);

}

* Используем уже ранее написанную функцию сохранения в файл.

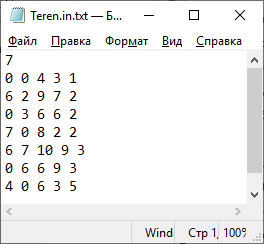


Рисунок . Результат сортировки входного файла

1. **Решите задачу**. Мариука впервые летит на самолёте. С самолёта она видит земли, на которых выращиваются культуры, описанные ранее. Будучи любопытной, маленькая девочка хотела бы определить площадь самого большого прямоугольника, занятого культурами одного и того же цвета. Разработайте подпрограмму, которая поможет Мариуке решить свою задачу.

* Для решения данной задачи мы должны понимать, что такой прямоугольник может быть как вертикальный, так и горизонтальный, по этой причине пишем две функции:

//Поиск максимальной территории горизонтально

static Tuple<int, int, int, int, int> FindMaxGorizontal(DataGridView dataGridView, int maxElement)

{

int Value;

int firstX, secondX;

int firstY, secondY = 0;

int k = -1;

int[,] Matrix = new int[1000, 4];

int[] Arr = new int[1000];

bool Continue;

bool Midle;

for (int i = 0; i < maxElement; i++)

for (int j = 0; j < maxElement; j++)

{

Continue = true;

Midle = true;

if (Convert.ToInt32(dataGridView.Rows[i].Cells[j].Value) != 0)

{

if (j == 0)

Midle = true;

else

if (Convert.ToInt32(dataGridView.Rows[i].Cells[j].Value) == Convert.ToInt32(dataGridView.Rows[i].Cells[j - 1].Value))

Midle = false;

if (Midle)

{

k++;

Value = Convert.ToInt32(dataGridView.Rows[i].Cells[j].Value);

int TempIndex = j + 1;

firstX = j;

firstY = i;

while (Convert.ToInt32(dataGridView.Rows[i].Cells[TempIndex].Value) == Value)

TempIndex++;

secondX = TempIndex;

while (Continue)

for (int p = firstY; Continue; p++)

{

for (int q = firstX; q < secondX; q++)

if (Convert.ToInt32(dataGridView.Rows[p].Cells[q].Value) != Value)

Continue = false;

secondY = p;

}

Matrix[k, 0] = firstX;

Matrix[k, 1] = firstY;

Matrix[k, 2] = secondX;

Matrix[k, 3] = secondY;

Arr[k] = (secondX - firstX) \* (secondY - firstY);

}

}

}

int max = 0;

int index = 0;

for (int i = 0; i < Arr.Length; i++)

if (Arr[i] > max)

{

max = Arr[i];

index = i;

}

return Tuple.Create<int, int, int, int, int>(Matrix[index, 0], Matrix[index, 1], Matrix[index, 2], Matrix[index, 3], max);

}

//Поиск максимальной территории вертикально

static Tuple<int, int, int, int, int> FindMaxVertical(DataGridView dataGridView, int maxElement)

{

int Value;

int firstX, secondX = 0;

int firstY, secondY;

int k = -1;

int[,] Matrix = new int[1000, 4];

int[] Arr = new int[1000];

bool Continue;

bool Midle;

for (int i = 0; i < maxElement; i++)

for (int j = 0; j < maxElement; j++)

{

Continue = true;

Midle = true;

if (Convert.ToInt32(dataGridView.Rows[i].Cells[j].Value) != 0)

{

if (i == 0)

Midle = true;

else

if (Convert.ToInt32(dataGridView.Rows[i - 1].Cells[j].Value) == Convert.ToInt32(dataGridView.Rows[i].Cells[j].Value))

Midle = false;

if (Midle)

{

k++;

Value = Convert.ToInt32(dataGridView.Rows[i].Cells[j].Value);

int TempIndex = i + 1;

firstX = j;

firstY = i;

while (Convert.ToInt32(dataGridView.Rows[TempIndex].Cells[j].Value) == Value)

TempIndex++;

secondY = TempIndex;

while (Continue)

for (int p = firstX; Continue; p++)

{

for (int q = firstY; q < secondY; q++)

if (Convert.ToInt32(dataGridView.Rows[q].Cells[p].Value) != Value)

Continue = false;

secondX = p;

}

Matrix[k, 0] = firstX;

Matrix[k, 1] = firstY;

Matrix[k, 2] = secondX;

Matrix[k, 3] = secondY;

Arr[k] = (secondX - firstX) \* (secondY - firstY);

}

}

}

int max = 0;

int index = 0;

for (int i = 0; i < Arr.Length; i++)

if (Arr[i] > max)

{

max = Arr[i];

index = i;

}

return Tuple.Create<int, int, int, int, int>(Matrix[index, 0], Matrix[index, 1], Matrix[index, 2], Matrix[index, 3], max);

}

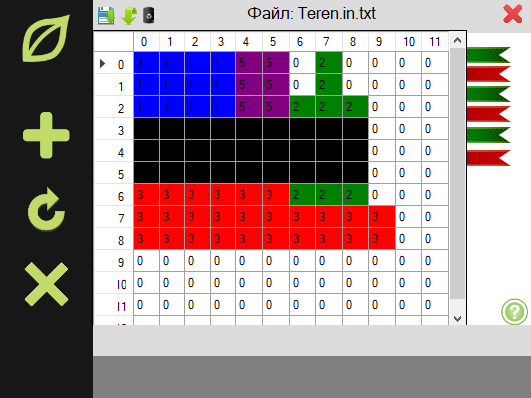


Рисунок . Результат выполнения шестой программы.

Это был основной функционал программы, который доступен пользователю. Стоит отметить, что через верхнюю панель возможно сохранить текущие участки, скрыть DataGridView2, а также очистить все участки с таблицы.

Значок вопроса откроет справку программы, которая доступна на трёх языках.

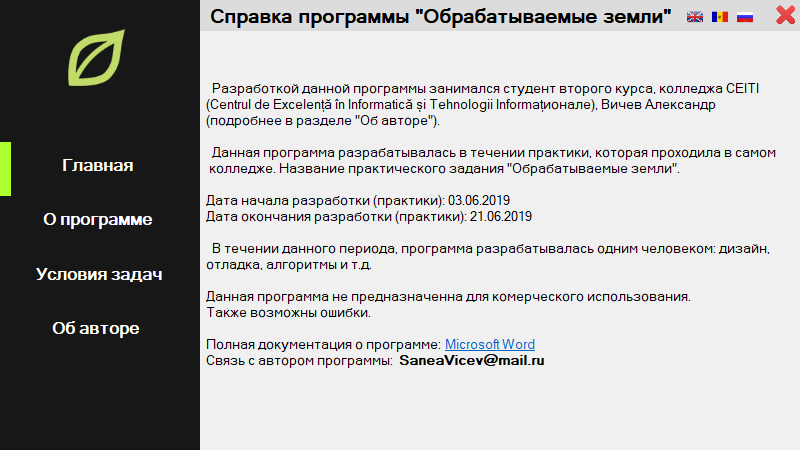


Рисунок . Справка программы.

# **Код программы**

**Основная форма:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

namespace Practic

{

public partial class MainForm : Form

{

int num = 0;

//Начальные настройки элементов формы=

public MainForm()

{

InitializeComponent();

label1.Visible = false;

label2.Visible = false;

label3.Visible = false;

dataGridView2.Visible = false;

}

//===============

//Действия по нажатию на кнопки формы.

//===============

//Button1\_Click============

//Добавление новой территории

private void Button1\_Click(object sender, EventArgs e)

{

label3.Visible = false;

if (dataGridView1.Visible == false)

{

dataGridView1.DataSource = ReadFromFile(@"E:\Всё для колледжа\Задачи по программированию\Задачи (основные)\Проекты\Практика 2 курс (2019)\Practic\bin\Debug\Teren.in.txt");

FillRecordNo(dataGridView1);

}

var sm = CreateDlgBox2();

AddRow(sm.Item1, sm.Item2, sm.Item3, sm.Item4, sm.Item5, dataGridView1);

label3.Visible = false;

dataGridView2.DataSource = VisualDataGrid(dataGridView1);

label1.Visible = true;

dataGridView2.Visible = true;

label1.Text = "Файл: " + "Teren.in.txt";

ForbideRead(dataGridView2);

FillRecordNo(dataGridView2);

ColorGridView();

NoSortable(dataGridView2);

NoSizebale(dataGridView2);

CheckZero();

}

//Button2\_Click============

//Чтение из файла/Обновление информации=================

private void Button2\_Click(object sender, EventArgs e)

{

label3.Visible = false;

dataGridView1.DataSource = ReadFromFile(@"E:\Всё для колледжа\Задачи по программированию\Задачи (основные)\Проекты\Практика 2 курс (2019)\Practic\bin\Debug\Teren.in.txt");

dataGridView2.DataSource = VisualDataGrid(dataGridView1);

label1.Visible = true;

dataGridView2.Visible = true;

label1.Text = "Файл: " + "Teren.in.txt";

ForbideRead(dataGridView2);

FillRecordNo(dataGridView2);

ColorGridView();

NoSortable(dataGridView2);

NoSizebale(dataGridView2);

CheckZero();

}

//Button3\_Click============

//Удаление участка=========

private void Button3\_Click(object sender, EventArgs e)

{

label3.Visible = false;

if (dataGridView1.Visible == false)

{

dataGridView1.DataSource = ReadFromFile(@"E:\Всё для колледжа\Задачи по программированию\Задачи (основные)\Проекты\Практика 2 курс (2019)\Practic\bin\Debug\Teren.in.txt");

FillRecordNo(dataGridView1);

}

var sm = CreateDlgBox3();

DelRow(sm.Item1, sm.Item2, sm.Item3, sm.Item4, dataGridView1);

label3.Visible = false;

dataGridView2.DataSource = VisualDataGrid(dataGridView1);

label1.Visible = true;

dataGridView2.Visible = true;

label1.Text = "Файл: " + "Teren.in.txt";

ForbideRead(dataGridView2);

FillRecordNo(dataGridView2);

ColorGridView();

NoSortable(dataGridView2);

NoSizebale(dataGridView2);

CheckZero();

}

//Button4\_Click============

//Поиск минимального прямоугольника===

private void Button4\_Click(object sender, EventArgs e)

{

if (dataGridView2.Visible == false)

{

label3.Visible = false;

dataGridView1.DataSource = ReadFromFile(@"E:\Всё для колледжа\Задачи по программированию\Задачи (основные)\Проекты\Практика 2 курс (2019)\Practic\bin\Debug\Teren.in.txt");

dataGridView2.DataSource = VisualDataGrid(dataGridView1);

label1.Visible = true;

dataGridView2.Visible = true;

label1.Text = "Файл: " + "Teren.in.txt";

ForbideRead(dataGridView2);

FillRecordNo(dataGridView2);

ColorGridView();

NoSortable(dataGridView2);

NoSizebale(dataGridView2);

}

CheckZero();

if (num != 0)

label2.Text = "Минимальный треугольник: x1 = " + Convert.ToString(MinimalPream(dataGridView1, num).Item1) +

", y1 = " + Convert.ToString(MinimalPream(dataGridView1, num).Item2) +

", x2 = " + Convert.ToString(MinimalPream(dataGridView1, num).Item3) +

", y2 = " + Convert.ToString(MinimalPream(dataGridView1, num).Item4);

else label2.Text = "Точек не существует";

label2.Visible = true;

}

//Button4\_Click============

//Сохранение в файл========

private void Button5\_Click(object sender, EventArgs e)

{

CheckZero();

SaveToFile(dataGridView1, @"Teren.in.txt");

}

//Button6\_Click============

//Цвет самого большого участка========

private void Button6\_Click(object sender, EventArgs e)

{

if (dataGridView2.Visible == false)

{

label3.Visible = false;

dataGridView1.DataSource = ReadFromFile(@"E:\Всё для колледжа\Задачи по программированию\Задачи (основные)\Проекты\Практика 2 курс (2019)\Practic\bin\Debug\Teren.in.txt");

dataGridView2.DataSource = VisualDataGrid(dataGridView1);

label1.Visible = true;

dataGridView2.Visible = true;

label1.Text = "Файл: " + "Teren.in.txt";

ForbideRead(dataGridView2);

FillRecordNo(dataGridView2);

ColorGridView();

NoSortable(dataGridView2);

NoSizebale(dataGridView2);

}

ForbideRead(dataGridView2);

FillRecordNo(dataGridView2);

ColorGridView();

NoSortable(dataGridView2);

NoSizebale(dataGridView2);

CheckZero();

label2.Text = "Цвет самого большого участка - " + Convert.ToString(ColorArea(dataGridView2));

label2.Visible = true;

}

//Button7\_Click============

//Общая площадь всех участков

private void Button7\_Click(object sender, EventArgs e)

{

if (dataGridView2.Visible == false)

{

label3.Visible = false;

dataGridView1.DataSource = ReadFromFile(@"E:\Всё для колледжа\Задачи по программированию\Задачи (основные)\Проекты\Практика 2 курс (2019)\Practic\bin\Debug\Teren.in.txt");

dataGridView2.DataSource = VisualDataGrid(dataGridView1);

label1.Visible = true;

dataGridView2.Visible = true;

label1.Text = "Файл: " + "Teren.in.txt";

ForbideRead(dataGridView2);

FillRecordNo(dataGridView2);

ColorGridView();

NoSortable(dataGridView2);

NoSizebale(dataGridView2);

}

CheckZero();

label2.Text = "Общая площадь всех участков - " + Convert.ToString(TotalArea(dataGridView2));

label2.Visible = true;

}

//Button8\_Click============

//Создание файла с цветами определённого участка========

private void Button8\_Click(object sender, EventArgs e)

{

if (dataGridView2.Visible == false)

{

label3.Visible = false;

dataGridView1.DataSource = ReadFromFile(@"E:\Всё для колледжа\Задачи по программированию\Задачи (основные)\Проекты\Практика 2 курс (2019)\Practic\bin\Debug\Teren.in.txt");

dataGridView2.DataSource = VisualDataGrid(dataGridView1);

label1.Visible = true;

dataGridView2.Visible = true;

label1.Text = "Файл: " + "Teren.in.txt";

ForbideRead(dataGridView2);

FillRecordNo(dataGridView2);

ColorGridView();

NoSortable(dataGridView2);

NoSizebale(dataGridView2);

}

CheckZero();

string numCp = CreateDlgBox();

CreateFile(@"ColColumn.txt", dataGridView1, numCp);

label2.Text = "Файл был сохранён.";

label2.Visible = true;

}

//Button9\_Click============

//Сортировка входного файла по цветам=

private void Button9\_Click(object sender, EventArgs e)

{

if (dataGridView2.Visible == false)

{

label3.Visible = false;

dataGridView1.DataSource = ReadFromFile(@"E:\Всё для колледжа\Задачи по программированию\Задачи (основные)\Проекты\Практика 2 курс (2019)\Practic\bin\Debug\Teren.in.txt");

dataGridView2.DataSource = VisualDataGrid(dataGridView1);

label1.Visible = true;

dataGridView2.Visible = true;

label1.Text = "Файл: " + "Teren.in.txt";

ForbideRead(dataGridView2);

FillRecordNo(dataGridView2);

ColorGridView();

NoSortable(dataGridView2);

NoSizebale(dataGridView2);

}

CheckZero();

if (num != 0)

{

Sort(dataGridView1);

SaveToFile(dataGridView1, @"Teren.in.txt");

}

else label2.Text = "Файл пуст!";

}

//Button10\_Click============

//Самая большая территория одного цвета

private void Button10\_Click(object sender, EventArgs e)

{

if (dataGridView2.Visible == false)

{

label3.Visible = false;

dataGridView1.DataSource = ReadFromFile(@"E:\Всё для колледжа\Задачи по программированию\Задачи (основные)\Проекты\Практика 2 курс (2019)\Practic\bin\Debug\Teren.in.txt");

dataGridView2.DataSource = VisualDataGrid(dataGridView1);

label1.Visible = true;

dataGridView2.Visible = true;

label1.Text = "Файл: " + "Teren.in.txt";

ForbideRead(dataGridView2);

FillRecordNo(dataGridView2);

ColorGridView();

NoSortable(dataGridView2);

NoSizebale(dataGridView2);

}

CheckZero();

int maxElement = MaxElem(dataGridView1);

var sm1 = FindMaxGorizontal(dataGridView2, maxElement);

var sm2 = FindMaxVertical(dataGridView2, maxElement);

dataGridView2.DataSource = VisualDataGrid(dataGridView1);

FillRecordNo(dataGridView2);

if (sm1.Item5 > sm2.Item5)

{

for (int i = sm1.Item2; i < sm1.Item4; i++)

{

for (int j = sm1.Item1; j < sm1.Item3; j++)

{

dataGridView2.Rows[i].Cells[j].Value = -1;

}

}

ColorGridView();

CreateFileMaxArea(@"E:\Всё для колледжа\Задачи по программированию\Задачи (основные)\Проекты\Практика 2 курс (2019)\Practic\bin\Debug\Teren.out.txt", Convert.ToString(sm1.Item5));

}

else if (sm1.Item5 < sm2.Item5)

{

for (int i = sm2.Item2; i < sm2.Item4; i++)

{

for (int j = sm2.Item1; j < sm2.Item3; j++)

{

dataGridView2.Rows[i].Cells[j].Value = -1;

}

}

ColorGridView();

CreateFileMaxArea(@"E:\Всё для колледжа\Задачи по программированию\Задачи (основные)\Проекты\Практика 2 курс (2019)\Practic\bin\Debug\Teren.out.txt", Convert.ToString(sm2.Item5));

}

else if (sm1.Item5 == sm2.Item5)

{

for (int i = sm1.Item2; i < sm1.Item4; i++)

{

for (int j = sm1.Item1; j < sm1.Item3; j++)

{

dataGridView2.Rows[i].Cells[j].Value = -1;

}

}

for (int i = sm2.Item2; i < sm2.Item4; i++)

{

for (int j = sm2.Item1; j < sm2.Item3; j++)

{

dataGridView2.Rows[i].Cells[j].Value = -1;

}

}

ColorGridView();

CreateFileMaxArea(@"E:\Всё для колледжа\Задачи по программированию\Задачи (основные)\Проекты\Практика 2 курс (2019)\Practic\bin\Debug\Teren.out.txt", Convert.ToString(sm2.Item5));

}

}

//Button11\_Click============

//Выход из программы========

private void Button11\_Click(object sender, EventArgs e)

{

CheckZero();

Application.Exit();

}

//Button12\_Click============

//Скрыть DataGridView2=======

private void Button12\_Click(object sender, EventArgs e)

{

CheckZero();

label1.Visible = false;

label2.Visible = false;

dataGridView2.Visible = false;

}

//Button13\_Click============

//Очистка всех территорий===

private void Button13\_Click(object sender, EventArgs e)

{

dataGridView1.SelectAll();

dataGridView1.ClearSelection();

dataGridView1.Columns.Clear();

num = 0;

dataGridView2.SelectAll();

dataGridView2.ClearSelection();

dataGridView2.Columns.Clear();

}

//Button14\_Click============

//Справка=

private void Button14\_Click(object sender, EventArgs e)

{

Form fm = new FAQ();

fm.ShowDialog();

}

//===============

//Дизайн, анимация и подсказки на кнопках

//===============

//===============

//Mouse Enter

//===============

//Button1\_MouseEnter============

private void Button1\_MouseEnter(object sender, EventArgs e)

{

label3.Text = "Добавить новый участок.";

label3.Visible = true;

}

//Button2\_MouseEnter============

private void Button2\_MouseEnter(object sender, EventArgs e)

{

label3.Text = "Обновить участки.";

label3.Visible = true;

}

//Button3\_MouseEnter============

private void Button3\_MouseEnter(object sender, EventArgs e)

{

label3.Text = "Удалить участок";

label3.Visible = true;

}

//Button4\_MouseEnter============

private void Button4\_MouseEnter(object sender, EventArgs e)

{

label3.Text = "Координаты прямоугольника минимальной площади " + "\nсодержащего все участки";

label3.Visible = true;

for (int i = 0; i < 10; i++)

button4.Location = new Point(button4.Location.X + 1, button4.Location.Y);

}

//Button5\_MouseEnter============

private void Button5\_MouseEnter(object sender, EventArgs e)

{

label3.Text = "Сохранить участки в файл.";

label3.Visible = true;

}

//Button6\_MouseEnter============

private void Button6\_MouseEnter(object sender, EventArgs e)

{

label3.Text = "Культура(цвет) занимающая большую площадь из всех участков.";

label3.Visible = true;

for (int i = 0; i < 10; i++)

button6.Location = new Point(button6.Location.X + 1, button6.Location.Y);

}

//Button7\_MouseEnter============

private void Button7\_MouseEnter(object sender, EventArgs e)

{

label3.Text = "Общая площадь, занятая всему участками.";

label3.Visible = true;

for (int i = 0; i < 10; i++)

button7.Location = new Point(button7.Location.X + 1, button7.Location.Y);

}

//Button8\_MouseEnter============

private void Button8\_MouseEnter(object sender, EventArgs e)

{

label3.Text = "Создание файла Colum.in.txt со строками координат, которые\n" +

"соответсвуют коду цвета, введённого с клавиатуры.";

label3.Visible = true;

for (int i = 0; i < 10; i++)

button8.Location = new Point(button8.Location.X + 1, button8.Location.Y);

}

//Button9\_MouseEnter============

private void Button9\_MouseEnter(object sender, EventArgs e)

{

label3.Text = "Сортировка строк в файле по значению цвета в порядке\nвозрастания.";

label3.Visible = true;

for (int i = 0; i < 10; i++)

button9.Location = new Point(button9.Location.X + 1, button9.Location.Y);

}

//Button10\_MouseEnter===========

private void Button10\_MouseEnter(object sender, EventArgs e)

{

label3.Text = "Определение площади самого большого прямоугольника, \nзанятого " +

"культурами одного и того же цвета.";

label3.Visible = true;

for (int i = 0; i < 10; i++)

button10.Location = new Point(button10.Location.X + 1, button10.Location.Y);

}

//Button11\_MouseEnter===========

private void Button11\_MouseEnter(object sender, EventArgs e)

{

label3.Text = "Закрыть программу.";

label3.Visible = true;

}

//Button12\_MouseEnter===========

private void Button12\_MouseEnter(object sender, EventArgs e)

{

label3.Text = "Очистить рабочую область.";

label3.Visible = true;

}

//Button13\_MouseEnter===========

private void Button13\_MouseEnter(object sender, EventArgs e)

{

label3.Text = "Очистить таблицу.";

label3.Visible = true;

}

//Button14\_MouseEnter===========

private void Button14\_MouseEnter(object sender, EventArgs e)

{

label3.Text = "Справка.";

label3.Visible = true;

}

//===============

//Mouse Enter

//===============

//Button1\_MouseLeave============

private void Button1\_MouseLeave(object sender, EventArgs e)

{

label3.Visible = false;

}

//Button2\_MouseLeave============

private void Button2\_MouseLeave(object sender, EventArgs e)

{

label3.Visible = false;

}

//Button3\_MouseLeave============

private void Button3\_MouseLeave(object sender, EventArgs e)

{

label3.Visible = false;

}

//Button4\_MouseLeave============

private void Button4\_MouseLeave(object sender, EventArgs e)

{

label3.Visible = false;

for (int i = 0; i < 10; i++)

button4.Location = new Point(button4.Location.X - 1, button4.Location.Y);

}

//Button5\_MouseLeave============

private void Button5\_MouseLeave(object sender, EventArgs e)

{

label3.Visible = false;

}

//Button6\_MouseLeave============

private void Button6\_MouseLeave(object sender, EventArgs e)

{

label3.Visible = false;

for (int i = 0; i < 10; i++)

button6.Location = new Point(button6.Location.X - 1, button6.Location.Y);

}

//Button7\_MouseLeave============

private void Button7\_MouseLeave(object sender, EventArgs e)

{

label3.Visible = false;

for (int i = 0; i < 10; i++)

button7.Location = new Point(button7.Location.X - 1, button7.Location.Y);

}

//===Button8\_MouseLeave==========

private void Button8\_MouseLeave(object sender, EventArgs e)

{

label3.Visible = false;

for (int i = 0; i < 10; i++)

button8.Location = new Point(button8.Location.X - 1, button8.Location.Y);

}

//Button9\_MouseLeave============

private void Button9\_MouseLeave(object sender, EventArgs e)

{

label3.Visible = false;

for (int i = 0; i < 10; i++)

button9.Location = new Point(button9.Location.X - 1, button9.Location.Y);

}

//Button10\_MouseLeave===========

private void Button10\_MouseLeave(object sender, EventArgs e)

{

label3.Visible = false;

for (int i = 0; i < 10; i++)

button10.Location = new Point(button10.Location.X - 1, button10.Location.Y);

}

//Button11\_MouseLeave===========

private void Button11\_MouseLeave(object sender, EventArgs e)

{

label3.Visible = false;

}

//Button12\_MouseLeave===========

private void Button12\_MouseLeave(object sender, EventArgs e)

{

label3.Visible = false;

}

//Button13\_MouseLeave===========

private void Button13\_MouseLeave(object sender, EventArgs e)

{

label3.Visible = false;

}

//Button14\_MouseLeave===========

private void Button14\_MouseLeave(object sender, EventArgs e)

{

label3.Visible = false;

}

//===============

//Визуальное оформление DataGridView2

//===============

//Запрет на выделение ячеек===

private void DataGridView2\_SelectionChanged(object sender, EventArgs e)

{

dataGridView2.ClearSelection();

}

//Запрет на изменение размеров столбцов/строк===============

private void NoSizebale(DataGridView dataGridView)

{

dataGridView.AllowUserToResizeColumns = false;

dataGridView.AllowUserToResizeRows = false;

}

//Запрет на сортировку по столбцу

private void NoSortable(DataGridView dataGridView)

{

int MaxElement = MaxElem(dataGridView1);

for (int i = 0; i <= MaxElement; i++)

dataGridView.Columns[i].SortMode = DataGridViewColumnSortMode.NotSortable;

}

//Запрет на добавление/изменение/удаление через DataGridView===========

private void ForbideRead(DataGridView dg)

{

int MaxElement = MaxElem(dataGridView1);

for (int i = 0; i < MaxElement; i++)

for (int j = 0; j < MaxElement; j++)

dg.Rows[i].Cells[j].ReadOnly = true;

}

//Вывол боковых колонок DataGridView2=====

private void FillRecordNo(DataGridView dg)

{

for (int i = 0; i < dg.Rows.Count; i++)

{

dg.Rows[i].HeaderCell.Style.Alignment = DataGridViewContentAlignment.MiddleCenter;

dg.Rows[i].HeaderCell.Value = (i).ToString();

}

}

//Раскрашивание ячеек DataGridView2========

private void ColorGridView()

{

int MaxElement = MaxElem(dataGridView1);

for (int i = 0; i < MaxElement; i++)

for (int j = 0; j < MaxElement; j++)

{

if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == -1)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.Black;

else if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 0)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.White;

else if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 1)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.Blue;

else if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 2)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.Green;

else if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 3)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.Red;

else if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 4)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.Pink;

else if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 5)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.Purple;

else if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 6)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.Silver;

else if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 7)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.Yellow;

else if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 8)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.Wheat;

else if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 9)

dataGridView2.Rows[i].Cells[j].Style.BackColor = Color.Tan;

}

}

//Создание DataGridView2 на основе DataGridView1======

private DataTable VisualDataGrid(DataGridView dataGridView)

{

DataTable visualDataTable = new DataTable();

int MaxElement = MaxElem(dataGridView1);

for (int i = 0; i <= MaxElement; i++)

visualDataTable.Columns.Add(Convert.ToString(i));

for (int j = 0; j <= MaxElement; j++)

visualDataTable.Rows.Add("");

for (int i = 0; i <= MaxElement; i++)

for (int j = 0; j <= MaxElement; j++)

visualDataTable.Rows[j][i] = 0;

for (int i = 0; i < num; i++)

for (int j = Convert.ToInt32(dataGridView.Rows[i].Cells[1].Value); j < Convert.ToInt32(dataGridView.Rows[i].Cells[3].Value); j++)

for (int k = Convert.ToInt32(dataGridView.Rows[i].Cells[0].Value); k < Convert.ToInt32(dataGridView.Rows[i].Cells[2].Value); k++)

visualDataTable.Rows[j][k] = Convert.ToInt32(dataGridView.Rows[i].Cells[4].Value);

return visualDataTable;

}

//===============

//Чтение из файла, сохранение файлов

//===============

//Чтение DataGridView1 из файла============

private DataTable ReadFromFile(string name)

{

DataTable dt = new DataTable();

int nn = 0;

try

{

using (StreamReader sr = new StreamReader(name))

{

string line;

string[] ar;

line = sr.ReadLine();

num = Convert.ToInt32(line);

while ((line = sr.ReadLine()) != null)

{

ar = line.Split(' ');

for (int i = 0; i < ar.Length; i++)

if (nn == 0) dt.Columns.Add(Convert.ToString(i + 1));

dt.Rows.Add(ar);

nn++;

}

sr.Close();

}

}

catch (Exception ex)

{

Console.WriteLine("Файл не может быть прочитан:");

Console.WriteLine(ex.Message);

}

return dt;

}

//Сохранение данных в файл Teren.in.txt====

private void SaveToFile(DataGridView dg, String fileName)

{

bool flag = true;

if (num == 0)

flag = CreateDlgBox4();

if (flag)

{

FileStream fs = new FileStream(@fileName, FileMode.Create);

StreamWriter streamWriter = new StreamWriter(fs);

streamWriter.Write(num);

streamWriter.WriteLine();

try

{

for (int j = 0; j < dg.Rows.Count; j++)

{

for (int i = 0; i < dg.Rows[j].Cells.Count; i++)

{

streamWriter.Write(dg.Rows[j].Cells[i].Value);

if (i < dg.Rows[j].Cells.Count - 1 && j < dg.Rows.Count - 1) streamWriter.Write(" ");

}

if (j < dg.Rows.Count - 1) streamWriter.WriteLine();

}

streamWriter.Close();

fs.Close();

label2.Visible = true;

label2.Text = "Файл успешно сохранен";

}

catch

{

MessageBox.Show("Ошибка при сохранении файла!");

}

}

}

//Создание файла со строками, соответствующими определённому цвету======

public static void CreateFile(string fileName, DataGridView dg, string numCp)

{

FileStream fs = new FileStream(@fileName, FileMode.Create);

StreamWriter streamWriter = new StreamWriter(fs);

bool WasWritte;

bool WasFound = false;

try

{

if (numCp == "") throw new Exception("Вы ничего не ввели.");

if (Convert.ToInt32(numCp) > 100 || Convert.ToInt32(numCp) < 0) throw new Exception("Значение цвета не входит в промежуток (1 - 100)");

for (int j = 0; j < dg.Rows.Count; j++)

{

WasWritte = false;

for (int i = 0; i < dg.Rows[j].Cells.Count; i++)

{

if (Convert.ToString(dg.Rows[j].Cells[4].Value) == numCp)

{

streamWriter.Write(dg.Rows[j].Cells[i].Value);

if (i < dg.Rows[j].Cells.Count - 1 && j < dg.Rows.Count - 1) streamWriter.Write(" ");

{

WasWritte = true;

WasFound = true;

}

}

}

if (WasWritte)

if (j < dg.Rows.Count - 1) streamWriter.WriteLine();

}

streamWriter.Close();

fs.Close();

if (WasFound)

MessageBox.Show("Файл успешно сохранен");

else

MessageBox.Show("Участка такого цвета не найдено.\nФайл пуст!");

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка: {ex.Message}\nЗначения не были сохранены.");

}

}

//Создание файла ColColumn.txt(максимальная территория)=================

public static void CreateFileMaxArea(string fileName, string MaxArea)

{

FileStream fs = new FileStream(@fileName, FileMode.Create);

StreamWriter streamWriter = new StreamWriter(fs);

try

{

streamWriter.Write(MaxArea);

streamWriter.Close();

fs.Close();

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка: {ex.Message}\nЗначения не были сохранены.");

}

}

//===============

//Создание диалоговых окон

//===============

//Запрос на цвет территории для записи в ColColumn.txt

public string CreateDlgBox()

{

Form Form2 = new Form

{

ControlBox = false,

Text = String.Empty,

FormBorderStyle = FormBorderStyle.FixedDialog

};

TextBox TextBox1 = new TextBox();

Label label1 = new Label();

Button btn1 = new Button();

string str;

TextBox1.Parent = Form2;

label1.Parent = Form2;

btn1.Parent = Form2;

btn1.Text = "OK";

label1.Text = "Введите номер: ";

label1.Bounds = new Rectangle(13, 5, 100, 20);

TextBox1.Bounds = new Rectangle(13, 20, 100, 20);

btn1.Bounds = new Rectangle(25, 50, 70, 30);

Form2.Bounds = new Rectangle(0, 0, 100, 100);

Form2.StartPosition = FormStartPosition.CenterParent;

btn1.DialogResult = DialogResult.OK;

Form2.ShowDialog();

str = TextBox1.Text;

Form2.AcceptButton = btn1;

return str;

}

//Запрос данных для добавления новой территории=======

static Tuple<string, string, string, string, string> CreateDlgBox2()

{

Form Form2 = new Form

{

ControlBox = false,

Text = String.Empty,

FormBorderStyle = FormBorderStyle.FixedDialog

};

TextBox TextBox1 = new TextBox();

TextBox TextBox2 = new TextBox();

TextBox TextBox3 = new TextBox();

TextBox TextBox4 = new TextBox();

TextBox TextBox5 = new TextBox();

Label label1 = new Label();

Label label2 = new Label();

Label label3 = new Label();

Label label4 = new Label();

Label label5 = new Label();

Button btn1 = new Button();

string x1, x2;

string y1, y2;

string Cp;

TextBox1.Parent = Form2;

TextBox2.Parent = Form2;

TextBox3.Parent = Form2;

TextBox4.Parent = Form2;

TextBox5.Parent = Form2;

label1.Parent = Form2;

label2.Parent = Form2;

label3.Parent = Form2;

label4.Parent = Form2;

label5.Parent = Form2;

btn1.Parent = Form2;

btn1.Text = "OK";

label1.Text = "X1:";

label2.Text = "Y1:";

label3.Text = "X2:";

label4.Text = "Y2:";

label5.Text = "Цвет(1-9):";

label1.Bounds = new Rectangle(13, 5, 100, 20);

TextBox1.Bounds = new Rectangle(13, 20, 100, 20);

label2.Bounds = new Rectangle(13, 45, 100, 20);

TextBox2.Bounds = new Rectangle(13, 60, 100, 20);

label3.Bounds = new Rectangle(13, 85, 100, 20);

TextBox3.Bounds = new Rectangle(13, 100, 100, 20);

label4.Bounds = new Rectangle(13, 125, 100, 20);

TextBox4.Bounds = new Rectangle(13, 140, 100, 20);

label5.Bounds = new Rectangle(13, 165, 100, 20);

TextBox5.Bounds = new Rectangle(13, 180, 100, 20);

btn1.Bounds = new Rectangle(25, 220, 70, 30);

Form2.Bounds = new Rectangle(0, 0, 70, 275);

Form2.StartPosition = FormStartPosition.CenterParent;

btn1.DialogResult = DialogResult.OK;

Form2.ShowDialog();

x1 = TextBox1.Text;

y1 = TextBox2.Text;

x2 = TextBox3.Text;

y2 = TextBox4.Text;

Cp = TextBox5.Text;

Form2.AcceptButton = btn1;

return Tuple.Create<string, string, string, string, string>(x1, y1, x2, y2, Cp);

}

//Запрос данных для удаления территории====

static Tuple<string, string, string, string> CreateDlgBox3()

{

Form Form2 = new Form

{

ControlBox = false,

Text = String.Empty,

FormBorderStyle = FormBorderStyle.FixedDialog

};

TextBox TextBox1 = new TextBox();

TextBox TextBox2 = new TextBox();

TextBox TextBox3 = new TextBox();

TextBox TextBox4 = new TextBox();

Label label1 = new Label();

Label label2 = new Label();

Label label3 = new Label();

Label label4 = new Label();

Button btn1 = new Button();

string x1, x2;

string y1, y2;

TextBox1.Parent = Form2;

TextBox2.Parent = Form2;

TextBox3.Parent = Form2;

TextBox4.Parent = Form2;

label1.Parent = Form2;

label2.Parent = Form2;

label3.Parent = Form2;

label4.Parent = Form2;

btn1.Parent = Form2;

btn1.Text = "OK";

label1.Text = "X1:";

label2.Text = "Y1:";

label3.Text = "X2:";

label4.Text = "Y2:";

label1.Bounds = new Rectangle(13, 5, 100, 20);

TextBox1.Bounds = new Rectangle(13, 20, 100, 20);

label2.Bounds = new Rectangle(13, 45, 100, 20);

TextBox2.Bounds = new Rectangle(13, 60, 100, 20);

label3.Bounds = new Rectangle(13, 85, 100, 20);

TextBox3.Bounds = new Rectangle(13, 100, 100, 20);

label4.Bounds = new Rectangle(13, 125, 100, 20);

TextBox4.Bounds = new Rectangle(13, 140, 100, 20);

btn1.Bounds = new Rectangle(25, 220, 70, 30);

Form2.Bounds = new Rectangle(0, 0, 70, 275);

Form2.StartPosition = FormStartPosition.CenterParent;

btn1.DialogResult = DialogResult.OK;

Form2.ShowDialog();

x1 = TextBox1.Text;

y1 = TextBox2.Text;

x2 = TextBox3.Text;

y2 = TextBox4.Text;

Form2.AcceptButton = btn1;

return Tuple.Create<string, string, string, string>(x1, y1, x2, y2);

}

//Запрос на сохранение файла, если num==0 =

public bool CreateDlgBox4()

{

Form Form2 = new Form

{

ControlBox = false,

Text = String.Empty,

FormBorderStyle = FormBorderStyle.FixedDialog

};

Label label1 = new Label();

Label label2 = new Label();

Label label3 = new Label();

bool flag = false;

Button btn1 = new Button();

Button btn2 = new Button();

label1.Parent = Form2;

label2.Parent = Form2;

label3.Parent = Form2;

btn1.Parent = Form2;

btn2.Parent = Form2;

btn1.Text = "Да";

btn2.Text = "Нет";

label1.Text = "Все поля пусты.";

label2.Text = "Файл будет пуст.";

label3.Text = "Продолжить?";

label1.Bounds = new Rectangle(13, 5, 100, 20);

label2.Bounds = new Rectangle(13, 23, 100, 20);

label3.Bounds = new Rectangle(13, 41, 100, 20);

btn1.Bounds = new Rectangle(0, 70, 55, 30);

btn2.Bounds = new Rectangle(74, 70, 55, 30);

Form2.Bounds = new Rectangle(0, 0, 100, 110);

Form2.StartPosition = FormStartPosition.CenterParent;

btn1.DialogResult = DialogResult.OK;

btn2.DialogResult = DialogResult.Cancel;

btn1.Click += new EventHandler(myhandler);

void myhandler(object sender, EventArgs e)

{

flag = true;

}

Form2.ShowDialog();

return flag;

}

//===============

//Функции проверок, возврата данных.

//===============

//Проверка: если num==0 =

private bool Check()

{

if (num == 0)

{

FileStream fs = new FileStream(@"E:\Всё для колледжа\Задачи по программированию\Задачи (основные)\Проекты\Практика 2 курс (2019)\Practic\bin\Debug\Teren.in.txt", FileMode.Create);

StreamWriter streamWriter = new StreamWriter(fs);

streamWriter.Write("1");

streamWriter.WriteLine();

streamWriter.Write("0 0 0 0 0");

streamWriter.Close();

fs.Close();

dataGridView1.DataSource = ReadFromFile(@"E:\Всё для колледжа\Задачи по программированию\Задачи (основные)\Проекты\Практика 2 курс (2019)\Practic\bin\Debug\Teren.in.txt");

return true;

}

return false;

}

//Возврат максимального элемента DataGridView1

private int MaxElem(DataGridView dataGridView)

{

int maxE = 0;

for (int j = 0; j < dataGridView.Rows.Count; j++)

{

int Tx2 = Convert.ToInt32(dataGridView.Rows[j].Cells[2].Value);

int Ty2 = Convert.ToInt32(dataGridView.Rows[j].Cells[3].Value);

if (Tx2 > maxE) maxE = Tx2;

if (Ty2 > maxE) maxE = Ty2;

}

return ++maxE;

}

//Проверка на нулевые координаты============

private void CheckZero()

{

int LocalNum=0;

string[] ar = { "1", "1", "1", "1", "1"};

try

{

using (StreamReader sr = new StreamReader(@"E:\Всё для колледжа\Задачи по программированию\Задачи (основные)\Проекты\Практика 2 курс (2019)\Practic\bin\Debug\Teren.in.txt"))

{

string line;

line = sr.ReadLine();

LocalNum = Convert.ToInt32(line);

line = sr.ReadLine();

ar = line.Split(' ');

sr.Close();

}

}

catch (Exception ex)

{

Console.WriteLine("Файл не может быть прочитан:");

Console.WriteLine(ex.Message);

}

if(LocalNum==1)

{

if(ar[0]=="0" && ar[1]=="0" && ar[2]=="0" && ar[3]=="0" && ar[4]=="0")

{

FileStream fs = new FileStream(@"E:\Всё для колледжа\Задачи по программированию\Задачи (основные)\Проекты\Практика 2 курс (2019)\Practic\bin\Debug\Teren.in.txt", FileMode.Create);

StreamWriter streamWriter = new StreamWriter(fs);

streamWriter.Write("0");

streamWriter.WriteLine();

streamWriter.Close();

fs.Close();

}

}

}

//===============

//Основные функции

//===============

//Добавление новой территории==============

private void AddRow(string x1, string y1, string x2, string y2, string Cp, DataGridView dataGridView)

{

int MaxElement = MaxElem(dataGridView1);

try

{

if (x1 == "" && y1 == "" && x2 == "" && y2 == "" && Cp == "")

throw new Exception("Вы ничего не ввели!");

if (x1 == "" || y1 == "" || x2 == "" || y2 == "" || Cp == "")

throw new Exception("Одно из значений не заполнено");

if (x1 == x2 || y1 == y2)

throw new Exception("Значения x1, x2 или y1=y2 равны между собой!");

int LocalX1 = Convert.ToInt32(x1);

int LocalY1 = Convert.ToInt32(y1);

int LocalX2 = Convert.ToInt32(x2);

int LocalY2 = Convert.ToInt32(y2);

int LocalCp = Convert.ToInt32(Cp);

bool WasFound = true;

if (LocalX1 > LocalX2 || LocalY1 > LocalY2)

{

if (LocalX1 > LocalX2)

{

int temp;

temp = LocalX1;

LocalX1 = LocalX2;

LocalX2 = temp;

}

if (LocalY1 > LocalY2)

{

int temp;

temp = LocalY1;

LocalY1 = LocalY2;

LocalY2 = temp;

}

MessageBox.Show("Координаты были введены неверно.\nПрограмма всё исправила!");

}

if (LocalX1 < 0 || LocalX1 > 1000000000 ||

LocalY1 < 0 || LocalY1 > 1000000000 ||

LocalX2 < 0 || LocalX2 > 1000000000 ||

LocalY2 < 0 || LocalY2 > 1000000000)

throw new Exception("Одно из значений не входит в промежуток (0 - 1000000000)");

if (LocalCp > 100 || LocalCp < 0)

throw new Exception("Значение цвета не входит в промежуток (1 - 100)");

bool ErrorZeroPoint = Check();

if (!(LocalX1 > MaxElement && LocalX2 > MaxElement && LocalY1 > MaxElement && LocalY2 > MaxElement))

{

for (int i = 0; i < MaxElement && WasFound; i++)

{

for (int j = 0; j < MaxElement && WasFound; j++)

{

if (LocalX1 <= j && j < LocalX2 && LocalY1 <= i && i < LocalY2)

{

if (Convert.ToInt32(dataGridView2.Rows[i].Cells[j].Value) == 0)

WasFound = true;

else WasFound = false;

}

}

}

}

if (WasFound)

{

DataTable dt = (DataTable)(dataGridView.DataSource);

DataRow newRow = dt.NewRow();

num++;

newRow[0] = Convert.ToString(LocalX1);

newRow[1] = Convert.ToString(LocalY1);

newRow[2] = Convert.ToString(LocalX2);

newRow[3] = Convert.ToString(LocalY2);

newRow[4] = Cp;

dt.Rows.InsertAt(newRow, dt.Rows.Count);

if (ErrorZeroPoint)

{

dataGridView.Rows.RemoveAt(0);

num--;

}

}

else throw new Exception("Прямоугольники пересекаются!");

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка: {ex.Message}\nЗначения не были добавлены.");

}

}

//Удаление участка=======

private void DelRow(string x1, string y1, string x2, string y2, DataGridView dataGridView)

{

int MaxElement = MaxElem(dataGridView);

try

{

if (x1 == "" && y1 == "" && x2 == "" && y2 == "")

throw new Exception("Вы ничего не ввели!");

if (x1 == "" || y1 == "" || x2 == "" || y2 == "")

throw new Exception("Одно из значений не заполнено");

if (x1 == x2 || y1 == y2)

throw new Exception("Значения x1, x2 или y1=y2 равны между собой!");

int LocalX1 = Convert.ToInt32(x1);

int LocalY1 = Convert.ToInt32(y1);

int LocalX2 = Convert.ToInt32(x2);

int LocalY2 = Convert.ToInt32(y2);

bool WasFound = true;

if (LocalX1 > LocalX2 || LocalY1 > LocalY2)

{

if (LocalX1 > LocalX2)

{

int temp;

temp = LocalX1;

LocalX1 = LocalX2;

LocalX2 = temp;

}

if (LocalY1 > LocalY2)

{

int temp;

temp = LocalY1;

LocalY1 = LocalY2;

LocalY2 = temp;

}

MessageBox.Show("Координаты были введены неверно.\nПрограмма всё исправила!");

}

if (LocalX1 < 0 || LocalX1 > 1000000000 ||

LocalY1 < 0 || LocalY1 > 1000000000 ||

LocalX2 < 0 || LocalX2 > 1000000000 ||

LocalY2 < 0 || LocalY2 > 1000000000)

throw new Exception("Одно из значений не входит в промежуток (0 - 1000000000)");

for (int i = 0; i < MaxElement && WasFound; i++)

{

for (int j = 0; j < MaxElement && WasFound; j++)

{

if (Convert.ToInt32(dataGridView.Rows[i].Cells[0].Value) == LocalX1

&& Convert.ToInt32(dataGridView.Rows[i].Cells[1].Value) == LocalY1

&& Convert.ToInt32(dataGridView.Rows[i].Cells[2].Value) == LocalX2

&& Convert.ToInt32(dataGridView.Rows[i].Cells[3].Value) == LocalY2)

{

dataGridView.Rows.RemoveAt(i);

WasFound = false;

num--;

}

}

}

if (WasFound) throw new Exception("Прямоугольник по вашим данным не найден");

else MessageBox.Show("Прямоугольник был удалён!");

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка: {ex.Message}\nЗначения не были добавлены.");

}

}

//Строит прямоугольник минимальной площади=

static Tuple<int, int, int, int> MinimalPream(DataGridView dataGridView, int numOf)

{

int x1 = 1000000000, y1 = 1000000000, x2 = 0, y2 = 0;

for (int i = 0; i < numOf; i++)

{

if (Convert.ToInt32(dataGridView.Rows[i].Cells[0].Value) < x1) x1 = Convert.ToInt32(dataGridView.Rows[i].Cells[0].Value);

if (Convert.ToInt32(dataGridView.Rows[i].Cells[1].Value) < y1) y1 = Convert.ToInt32(dataGridView.Rows[i].Cells[1].Value);

if (Convert.ToInt32(dataGridView.Rows[i].Cells[2].Value) > x2) x2 = Convert.ToInt32(dataGridView.Rows[i].Cells[2].Value);

if (Convert.ToInt32(dataGridView.Rows[i].Cells[3].Value) > y2) y2 = Convert.ToInt32(dataGridView.Rows[i].Cells[3].Value);

}

return Tuple.Create<int, int, int, int>(x1, y1, x2, y2);

}

//Находит общую площадь всех прямоугольников=

private int TotalArea(DataGridView dataGridView)

{

int Area = 0;

int MaxElement = MaxElem(dataGridView1);

for (int i = 0; i < MaxElement; i++)

{

for (int j = 0; j < MaxElement; j++)

{

if (Convert.ToInt32(dataGridView.Rows[i].Cells[j].Value) != 0)

Area++;

}

}

return Area;

}

//Цвет участка с самой большой площадью====

private int ColorArea(DataGridView dataGridView)

{

int[] Area = new int[100];

int max = 0;

int maxValue = 0;

int MaxElement = MaxElem(dataGridView1);

for (int i = 0; i < 100; i++)

Area[i] = 0;

for (int i = 0; i < MaxElement; i++)

for (int j = 0; j < MaxElement; j++)

{

if (Convert.ToInt32(dataGridView.Rows[i].Cells[j].Value) != 0)

Area[Convert.ToInt32(dataGridView.Rows[i].Cells[j].Value)]++;

}

for (int i = 0; i < 100; i++)

if (Area[i] > maxValue)

{

max = i;

maxValue = Area[i];

}

return max;

}

//Сортировка цветов в порядке возрастания==

private void Sort(DataGridView dataGridView)

{

DataGridViewColumn newColumn = dataGridView.Columns[4];

dataGridView.Sort(newColumn, ListSortDirection.Ascending);

}

//===============

//Основное задание

//===============

//Поиск максимальной территории горизонтально

static Tuple<int, int, int, int, int> FindMaxGorizontal(DataGridView dataGridView, int maxElement)

{

int Value;

int firstX, secondX;

int firstY, secondY = 0;

int k = -1;

int[,] Matrix = new int[1000, 4];

int[] Arr = new int[1000];

bool Continue;

bool Midle;

for (int i = 0; i < maxElement; i++)

for (int j = 0; j < maxElement; j++)

{

Continue = true;

Midle = true;

if (Convert.ToInt32(dataGridView.Rows[i].Cells[j].Value) != 0)

{

if (j == 0)

Midle = true;

else

if (Convert.ToInt32(dataGridView.Rows[i].Cells[j].Value) == Convert.ToInt32(dataGridView.Rows[i].Cells[j - 1].Value))

Midle = false;

if (Midle)

{

k++;

Value = Convert.ToInt32(dataGridView.Rows[i].Cells[j].Value);

int TempIndex = j + 1;

firstX = j;

firstY = i;

while (Convert.ToInt32(dataGridView.Rows[i].Cells[TempIndex].Value) == Value)

TempIndex++;

secondX = TempIndex;

while (Continue)

for (int p = firstY; Continue; p++)

{

for (int q = firstX; q < secondX; q++)

if (Convert.ToInt32(dataGridView.Rows[p].Cells[q].Value) != Value)

Continue = false;

secondY = p;

}

Matrix[k, 0] = firstX;

Matrix[k, 1] = firstY;

Matrix[k, 2] = secondX;

Matrix[k, 3] = secondY;

Arr[k] = (secondX - firstX) \* (secondY - firstY);

}

}

}

int max = 0;

int index = 0;

for (int i = 0; i < Arr.Length; i++)

if (Arr[i] > max)

{

max = Arr[i];

index = i;

}

return Tuple.Create<int, int, int, int, int>(Matrix[index, 0], Matrix[index, 1], Matrix[index, 2], Matrix[index, 3], max);

}

//Поиск максимальной территории вертикально

static Tuple<int, int, int, int, int> FindMaxVertical(DataGridView dataGridView, int maxElement)

{

int Value;

int firstX, secondX = 0;

int firstY, secondY;

int k = -1;

int[,] Matrix = new int[1000, 4];

int[] Arr = new int[1000];

bool Continue;

bool Midle;

for (int i = 0; i < maxElement; i++)

for (int j = 0; j < maxElement; j++)

{

Continue = true;

Midle = true;

if (Convert.ToInt32(dataGridView.Rows[i].Cells[j].Value) != 0)

{

if (i == 0)

Midle = true;

else

if (Convert.ToInt32(dataGridView.Rows[i - 1].Cells[j].Value) == Convert.ToInt32(dataGridView.Rows[i].Cells[j].Value))

Midle = false;

if (Midle)

{

k++;

Value = Convert.ToInt32(dataGridView.Rows[i].Cells[j].Value);

int TempIndex = i + 1;

firstX = j;

firstY = i;

while (Convert.ToInt32(dataGridView.Rows[TempIndex].Cells[j].Value) == Value)

TempIndex++;

secondY = TempIndex;

while (Continue)

for (int p = firstX; Continue; p++)

{

for (int q = firstY; q < secondY; q++)

if (Convert.ToInt32(dataGridView.Rows[q].Cells[p].Value) != Value)

Continue = false;

secondX = p;

}

Matrix[k, 0] = firstX;

Matrix[k, 1] = firstY;

Matrix[k, 2] = secondX;

Matrix[k, 3] = secondY;

Arr[k] = (secondX - firstX) \* (secondY - firstY);

}

}

}

int max = 0;

int index = 0;

for (int i = 0; i < Arr.Length; i++)

if (Arr[i] > max)

{

max = Arr[i];

index = i;

}

return Tuple.Create<int, int, int, int, int>(Matrix[index, 0], Matrix[index, 1], Matrix[index, 2], Matrix[index, 3], max);

}

}

}

## **Форма справки**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Practic

{

public partial class FAQ : Form

{

public FAQ()

{

InitializeComponent();

radioButton2.Checked = true;

label1.Text = "Справка программы \"Обрабатываемые земли\"";

radioButton1.Checked = false;

}

int Btn = 0;

//Button1\_Click=============

//о программе===============

private void Button1\_Click(object sender, EventArgs e)

{

SlidePanel.Height = button1.Height;

SlidePanel.Top = button1.Top;

if (radioButton2.Checked)

userControl1.BringToFront();

else if (radioButton1.Checked)

userControl2.BringToFront();

else if (radioButton3.Checked)

userControl91.BringToFront();

Btn = 1;

}

//Button2\_Click=============

//о программе===============

private void Button2\_Click(object sender, EventArgs e)

{

SlidePanel.Height = button2.Height;

SlidePanel.Top = button2.Top;

if (radioButton2.Checked)

userControl31.BringToFront();

else if (radioButton1.Checked)

userControl41.BringToFront();

else if (radioButton3.Checked)

userControl101.BringToFront();

Btn = 2;

}

//Button3\_Click=============

//о программе===============

private void Button3\_Click(object sender, EventArgs e)

{

SlidePanel.Height = button3.Height;

SlidePanel.Top = button3.Top;

if (radioButton2.Checked)

userControl51.BringToFront();

else if (radioButton1.Checked)

userControl61.BringToFront();

else if (radioButton3.Checked)

userControl111.BringToFront();

Btn = 3;

}

//Button5\_Click=============

//об авторе===============

private void Button5\_Click(object sender, EventArgs e)

{

SlidePanel.Height = button5.Height;

SlidePanel.Top = button5.Top;

if (radioButton2.Checked)

userControl71.BringToFront();

else if (radioButton1.Checked)

userControl81.BringToFront();

else if (radioButton3.Checked)

userControl121.BringToFront();

Btn = 4;

}

//Button4\_Click=============

//Закрытие справки==========

private void Button4\_Click(object sender, EventArgs e)

{

this.Hide();

}

//PictureBox3\_Click=========

//Румынский язык==============

private void PictureBox2\_Click(object sender, EventArgs e)

{

radioButton1.Checked = true;

radioButton2.Checked = false;

radioButton3.Checked = false;

}

//PictureBox3\_Click=========

//Русский язык==============

private void PictureBox3\_Click(object sender, EventArgs e)

{

radioButton1.Checked = false;

radioButton2.Checked = true;

radioButton3.Checked = false;

}

//PictureBox4\_Click=========

//Английский язык==============

private void PictureBox4\_Click(object sender, EventArgs e)

{

radioButton1.Checked = false;

radioButton2.Checked = false;

radioButton3.Checked = true;

}

public void RadioButton1\_CheckedChanged(object sender, EventArgs e)

{

if (Btn == 1)

Button1\_Click(sender, e);

else if (Btn == 2)

Button2\_Click(sender, e);

else if (Btn == 3)

Button3\_Click(sender, e);

else if (Btn == 4)

Button5\_Click(sender, e);

label1.Text = "Ajutor \"Terenuri cultivate\"";

button1.Text = "Despre program";

button2.Text = "Termeni de sarcini";

button3.Text = "Principala";

button5.Text = "Despre autor";

}

private void RadioButton2\_CheckedChanged(object sender, EventArgs e)

{

if (Btn == 1)

Button1\_Click(sender, e);

else if (Btn == 2)

Button2\_Click(sender, e);

else if (Btn == 3)

Button3\_Click(sender, e);

else if (Btn == 4)

Button5\_Click(sender, e);

label1.Text = "Справка программы \"Обрабатываемые земли\"";

button1.Text = "О программе";

button2.Text = "Условия задач";

button3.Text = "Главная";

button5.Text = "Об авторе";

}

private void RadioButton3\_CheckedChanged(object sender, EventArgs e)

{

if (Btn == 1)

Button1\_Click(sender, e);

else if (Btn == 2)

Button2\_Click(sender, e);

else if (Btn == 3)

Button3\_Click(sender, e);

else if (Btn == 4)

Button5\_Click(sender, e);

label1.Text = "FAQ \"Cultivated land.\"";

button1.Text = "About the program";

button2.Text = "Task conditions";

button3.Text = "The main";

button5.Text = "About the author";

}

}

}

## **Форма загрузки**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Practic

{

public partial class Loading : Form

{

public Loading()

{

InitializeComponent();

timer1.Start();

}

private void Timer1\_Tick(object sender, EventArgs e)

{

this.Opacity += .03;

if(this.Opacity==1)

{

timer1.Stop();

}

}

}

}

# **Вывод**

Практическая работа – это самая главная контрольная/тест или скорее всего самостоятельная работа по всем знания, которые мы получили в течении целого года обучения в колледже.

Каждый получил своё индивидуальное задание, которое необходимо было реализовать не в обычной консоли, а в полноценном приложении C#. Для нас это был новый опыт, но очень ценный опыт, который показал, разницу между С++ и С#, который показал гибкость языков, правильное использование ресурсов компьютера, а самое главное сделать внешний вид программы приятной для восприятия пользователя.

Для меня данная практическая работа представляла лишь одну трудность – знания языка C# - желали лучшего, но надо было как-то решать эту проблему. Основным моим помощником стал – Google. Далее шаг за шагом я разбирал каждую функцию по шагам.

На последних заданиях я уже всё делал без каких-либо подсказок. Многие тонкости языка схожи с С++, а новые знания (к примеру: DataGridView и работа с ним) уже довольно хорошо вырисовывались в моей голове.

Моя практическая работа не была сложной, но она требовала много внимания к деталям и различным ошибкам (как логическим, так и тем, которые могли возникнуть при разборе самого задания).

Ещё одним важным пунктом работы с C# Forms – умение сделать дизайн своей программы лёгким в понимание и «свежим» на взгляд для пользователя. Программа должна быть одновременно и простой и в то же время завлекать чем-то (к примеру, так называемые «свежие» цвета, которые привлекают внимание человека). Эта практическая работа меня с этим прекрасно познакомила.

Если подводить общие итоги, я бы сказал, что данная практика ещё один, но очень большой шаг вперёд. Данная программа может стать отличным примером в CV, её не стыдно показать и представить.

Я очень рад, что колледж CEITI может предоставлять такие услуги начинающим программистам. Это очень приятно, а главное – интересно!

# **Библиография**

Джеффри Рихтер, & Мартвен ван де Боспурт. (б.д.). *WinRT программирование на С# для профессионалов.*

П.В., Е. (б.д.). *С# на примерах.*

Робинсон, У. (б.д.). *C# без лишних слов.*

Фленов, М. (б.д.). *Библия C# 3-е издание.*

Эндрю Стиллмен, & Дженнифер Грин. (б.д.). *Изучаем C#.*