



МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Колледж программирования и кибербезопасности

**УЧЕБНАЯ ДИСЦИПЛИНА ОП.03 Основы алгоритмизации и
программирования**

ПЦК Информационной безопасности

ВИД УЧЕБНОГО МАТЕРИАЛА Практическое занятие

ПРЕПОДАВАТЕЛЬ Беседин А.В.

СЕМЕСТР 4 семестр

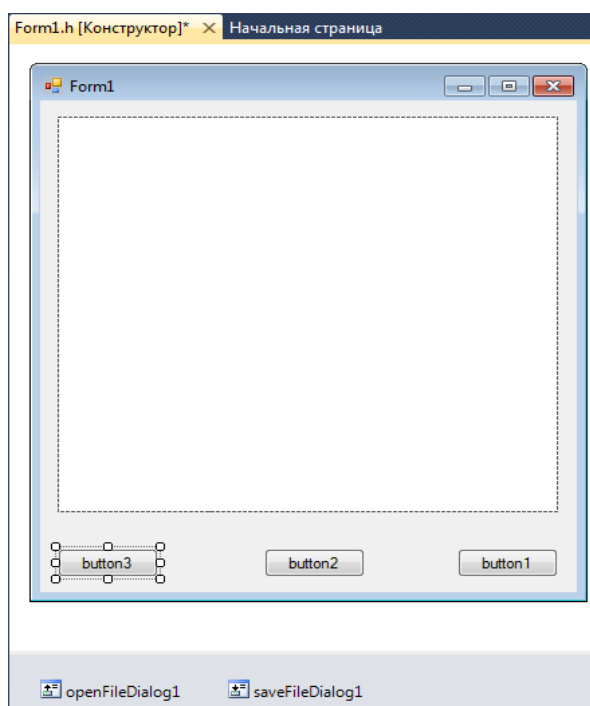
Практическое занятие №36

Разработка многомодульных приложений. Открытие и запись текстового файла.

1. Отработайте навыки работы с элементами `openFileDialog` и `saveFileDialog`. Выполните представленную инструкцию написания программы.

Рассмотрим два очень важных элемента – это `"openFileDialog"`, который будет загружать файл из определённой папки с помощью проводника и `"saveFileDialog"`, который будет сохранять проделанную работу в виде файла в определённую папку, так же с помощью проводника. Все не раз пользовались такой кнопкой, как “Обзор”, загружая фотографии, текстовые документы и прочие файлы. Именно это и будет рассмотрено в этом задании. В коде этих элементов пишется “фильтр”, в котором указываются форматы файлов, которые нужно загрузить, прочие проводник просто не будет отображать, и так же фильтр ставится на форматы, в которых файл может быть сохранён. Рассмотрим работу с изображением - пользователь загружает в `"PictureBox"` картинку и рисует что-нибудь на ней кистью. После того, как картинка видоизменена, пользователь сохраняет её в определённую папку, задав ей определённое имя.

Создайте проект в приложении `"Windows Forms"` и перенесите на форму шесть элементов: 1 `"PictureBox"`, 3 `"button"`, 1 `"openFileDialog"` и 1 `"saveFileDialog"`, оформив программу следующим образом:



Button1 будет выполнять функцию загрузки файла.

Button2 будет сохранять изменённое изображение.

Button3 будет выполнять функцию очистки, если пользователю не понравилось, то как он редактировал изображение.

В этом проекте понадобится вызвать три события у "PictureBox" – "MouseMove", "MouseDown", "MouseUp". Как всем известно изображения бывают разных форматов, поэтому в фильтре будут написаны все возможные. По мимо этого в коде будет описано свойство, с помощью которого размер "PictureBox" будет равен размеру загружаемого изображения, возможно и обратное.

Код программы :

```
namespace OpenSave_1_
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            pen_for_draw.StartCap = pen_for_draw.EndCap =
            System.Drawing.Drawing2D.LineCap.Round;
        }

        private Bitmap bmp_for_draw;
        private Point start_point;
        private bool Draw;
        private Pen pen_for_draw = new Pen(Color.Black, 4);
        private string full_name_of_image;

        private void button1_Click(object sender, EventArgs e)
        {
            OpenFileDialog open_dialog = new OpenFileDialog();
            open_dialog.Filter = "Image
            Files (*.BMP;*.JPG;*.GIF;*.PNG)|*.BMP;*.JPG;*.GIF;*.PNG|All files (*.*)|*.*";
            if (open_dialog.ShowDialog() == DialogResult.OK)
            {
                try
                {
                    full_name_of_image = open_dialog.FileName;
                    bmp_for_draw = new Bitmap(open_dialog.FileName);
                    this.pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
                    pictureBox1.Image = bmp_for_draw;
                    pictureBox1.Invalidate();
                }
                catch
                {
                    DialogResult rezult = MessageBox.Show("Impossible to open selected file",
                    "Warning", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            if (pictureBox1.Image != null)
            {
                string format = full_name_of_image.Substring(full_name_of_image.Length - 4, 4);

                SaveFileDialog savedialog = new SaveFileDialog();
```

```

savedialog.Title = "Сохранить картинку как ...";
savedialog.OverwritePrompt = true;
savedialog.CheckPathExists = true;
savedialog.Filter = "Image Files (*.BMP)|*.BMP|Image Files (*.JPG)|*.JPG|Image Files (*.GIF)|*.GIF|Image Files (*.PNG)|*.PNG|All files (*.*)|*.*";
savedialog.ShowHelp = true;
// If selected, save
if (savedialog.ShowDialog() == DialogResult.OK)
{
    try
    {
        bmp_for_draw.Save(savedialog.FileName, System.Drawing.Imaging.ImageFormat.Jpeg);
    }
    catch
    {
        MessageBox.Show("Impossible to save image", "FATAL ERROR",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void button3_Click(object sender, EventArgs e)
{
    Graphics Графика = pictureBox1.CreateGraphics();
    Графика.Clear(SystemColors.Window);
}

private void pictureBox1_MouseMove(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        using (Graphics graf = Graphics.FromImage(pictureBox1.Image))
        {
            graf.DrawLine(pen_for_draw, start_point, e.Location);
            start_point = e.Location;
            pictureBox1.Invalidate();
        }
    }
}

private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        Draw = true;
        start_point = e.Location;
    }
}

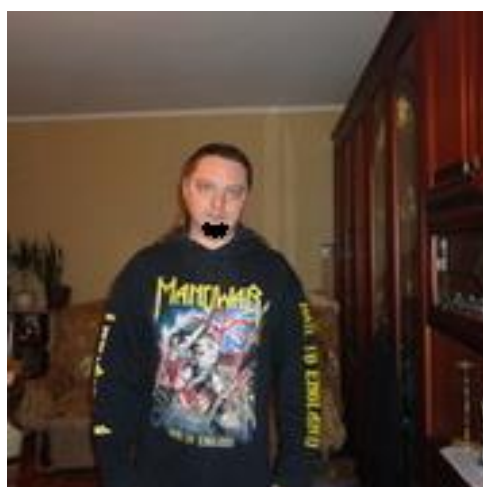
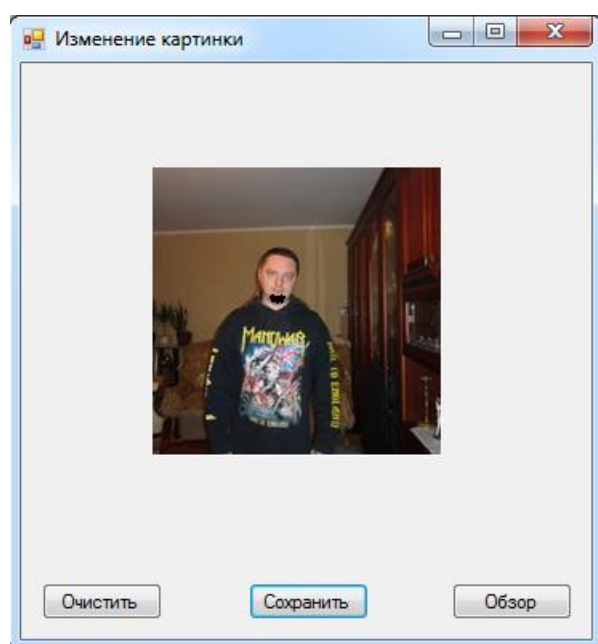
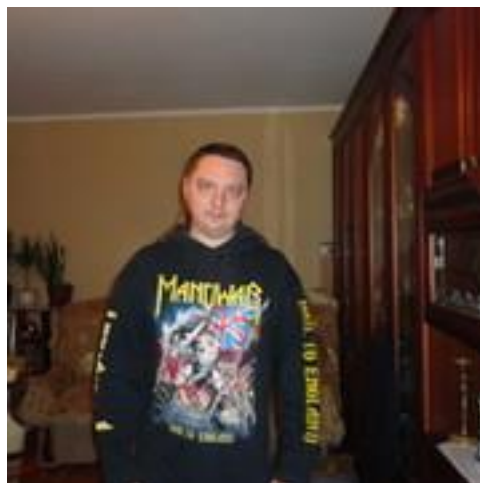
private void pictureBox1_MouseUp(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        Draw = false;
    }
}

private void Form1_Load(object sender, EventArgs e)
{
    this.Text = "Изменение картинки";
    button1.Text = "Обзор";
    button2.Text = "Сохранить";
    button3.Text = "Очистить";
}

}

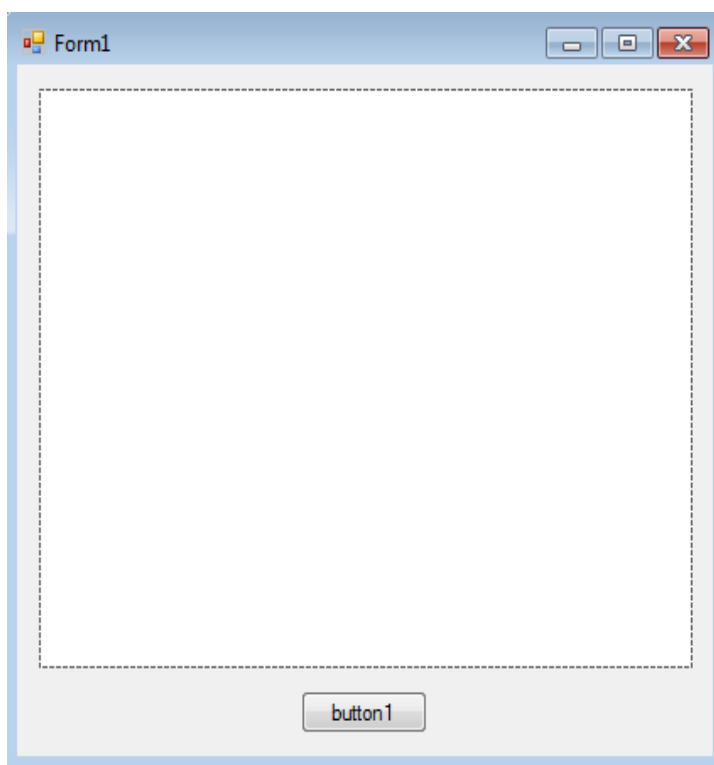
```

Результат:



2. Отработайте навыки работы с элементом Paint. Выполните представленную инструкцию написания программы.

Все знают такую программу, как "Paint". Одно из самых ярких его свойств – это рисование произвольных линий кистью - именно это мы сейчас и сделаем. Для этого нужно перетащить на форму "PictureBox" и выбрать у него свойства "MouseDown", "MouseUp" и "MouseMove". Алгоритм работы программы следующий: когда пользователь нажимает на левую кнопки мышки, то за курсором начинает прорисовываться огромное количество маленьких квадратиков. Размер этих квадратиков указывается в коде. Так же на форму нужно перенести кнопку "button", которая будет очищать поле рисования. Вид программы:



Теперь перейдём к коду, обязательно нужно создать переменную типа "bool":

```
#pragma endregion
```

```
bool Draw;
```

```
private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e) {  
    this->Text = "Рисование кистью"; button1->Text = "Очистить"; }
```

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {  
    Graphics ^ Графика = pictureBox1->CreateGraphics();  
    Графика->Clear(SystemColors::Window);  
}
```

```

private: System::Void pictureBox1_MouseMove(System::Object^ sender,
System::Windows::Forms::MouseEventArgs^ e) {
Graphics^ graf = pictureBox1->CreateGraphics();
if (Drow == true)
{
graf->FillEllipse(Brushes::Violet, e->X, e->Y, 3,3); // толщина кисти
}
}

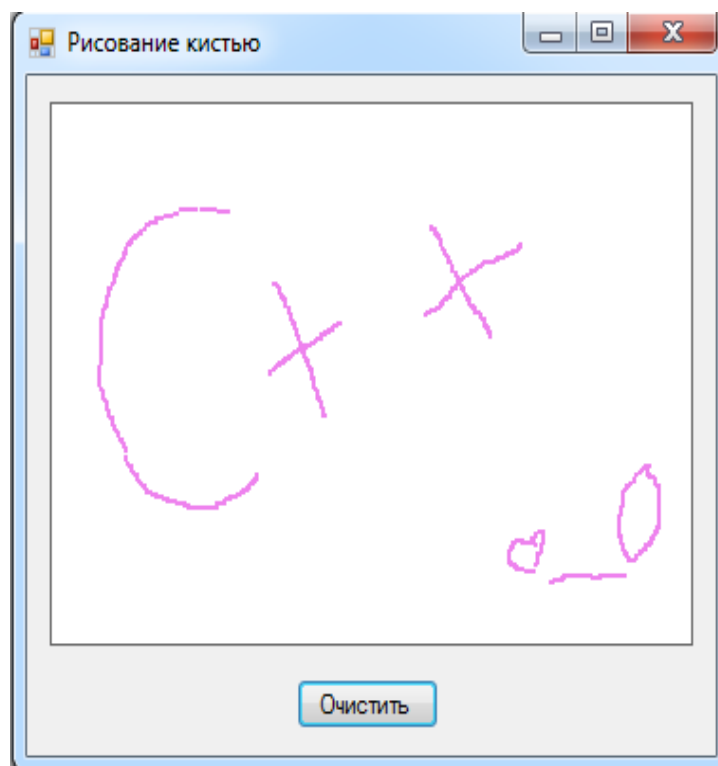
private: System::Void pictureBox1_MouseDown(System::Object^ sender,
System::Windows::Forms::MouseEventArgs^ e) {
Drow = true;
}

private: System::Void pictureBox1_MouseUp(System::Object^ sender,
System::Windows::Forms::MouseEventArgs^ e) {
Drow = false;
}

};
}

```

Результат:



Сформировать отчет по решенным задачам (пример отчета страница 8 – 23).
Решенные задачи должны сопровождаться, минимум двумя тестовыми
испытаниями.

Критерий оценки:

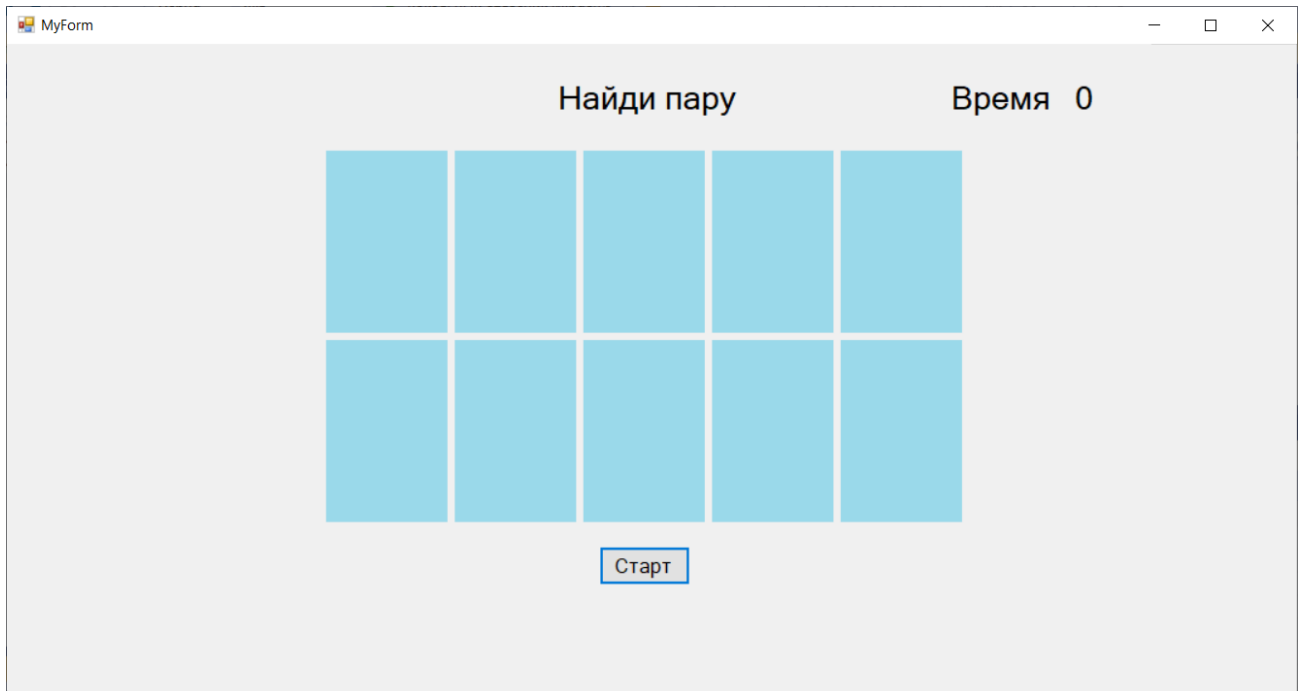
5 (отлично) – отработаны две задачи. Соблюдено время на выполнение задания. Сформирован отчет по двум задачам.

4 (хорошо) – отработаны две задачи. Время на выполнение задания не соблюдено. Сформирован отчет по двум задачам.

3 (удовлетворительно) – отработана одна из двух задач. Соблюдено время на выполнение задания. Сформирован отчет по одной задаче.

2 (неудовлетворительно) – не решено ни одной задачи.

1. Написать игру «Найди пару». Игрок должен открывать картинки по очереди и находить две одинаковые картинки. Если игрок открыл две не одинаковые картинки, они закрываются. Найденные одинаковые картинки исчезают с игрового поля. Также должно учитываться время прохождения игры.



Код программы (MyForm.h)

```
#pragma once
#include <iostream>
#include <time.h>
#include <windows.h>
#include <dos.h>

namespace igrapara {

    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;

    double Time;
    int flag = 0;
    int mas1[10] = { 0 };
    int mas[10] = { 0 }, z, f, f1 = 0, k = 0, a1 = 0, a2 = 0, a3 = 0, a4 = 0, a5 = 0;
    int q1, q2, m;
    int s = 0, k1, k2;
    int x, t=0;
```

```

/// <summary>
/// Сводка для MyForm
/// </summary>
public ref class MyForm : public System::Windows::Forms::Form
{
public:
    MyForm(void)
    {
        InitializeComponent();
        //
        //TODO: добавьте код конструктора
        //
    }

protected:
    /// <summary>
    /// Освободить все используемые ресурсы.
    /// </summary>
    ~MyForm()
    {
        if (components)
        {
            delete components;
        }
    }

private: System::Windows::Forms::PictureBox^ pictureBox1;
protected:
private: System::Windows::Forms::PictureBox^ pictureBox2;
private: System::Windows::Forms::PictureBox^ pictureBox3;
private: System::Windows::Forms::PictureBox^ pictureBox4;
private: System::Windows::Forms::PictureBox^ pictureBox5;
private: System::Windows::Forms::PictureBox^ pictureBox6;
private: System::Windows::Forms::PictureBox^ pictureBox7;
private: System::Windows::Forms::PictureBox^ pictureBox8;
private: System::Windows::Forms::PictureBox^ pictureBox9;
private: System::Windows::Forms::PictureBox^ pictureBox10;
private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::Button^ button1;
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::Timer^ timer1;
private: System::Windows::Forms::Label^ label3;
private: System::Windows::Forms::Label^ label4;
private: System::Windows::Forms::Timer^ timer2;
private: System::Windows::Forms::Label^ label5;
private: System::Windows::Forms::Label^ label6;

```

```
private: System::Windows::Forms::Label^ label7;
private: System::Windows::Forms::Label^ label8;
private: System::Windows::Forms::Timer^ timer3;
private: System::Windows::Forms::Timer^ timer4;
private: System::ComponentModel::IContainer^ components;
```

```
private:
    /// <summary>
    /// Обязательная переменная конструктора.
    /// </summary>
```

```
#pragma region Windows Form Designer generated code
```

```
    /// <summary>
    /// Требуемый метод для поддержки конструктора – не изменяйте
    /// содержимое этого метода с помощью редактора кода.
    /// </summary>
    void InitializeComponent(void)
    {
        this->components = (gcnew System::ComponentModel::Container());
        this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());
        this->pictureBox2 = (gcnew System::Windows::Forms::PictureBox());
        this->pictureBox3 = (gcnew System::Windows::Forms::PictureBox());
        this->pictureBox4 = (gcnew System::Windows::Forms::PictureBox());
        this->pictureBox5 = (gcnew System::Windows::Forms::PictureBox());
        this->pictureBox6 = (gcnew System::Windows::Forms::PictureBox());
        this->pictureBox7 = (gcnew System::Windows::Forms::PictureBox());
        this->pictureBox8 = (gcnew System::Windows::Forms::PictureBox());
        this->pictureBox9 = (gcnew System::Windows::Forms::PictureBox());
        this->pictureBox10 = (gcnew System::Windows::Forms::PictureBox());
        this->label1 = (gcnew System::Windows::Forms::Label());
        this->button1 = (gcnew System::Windows::Forms::Button());
        this->label2 = (gcnew System::Windows::Forms::Label());
        this->timer1 = (gcnew System::Windows::Forms::Timer(this-
>components));
        this->label3 = (gcnew System::Windows::Forms::Label());
        this->label4 = (gcnew System::Windows::Forms::Label());
        this->timer2 = (gcnew System::Windows::Forms::Timer(this-
>components));
        this->label5 = (gcnew System::Windows::Forms::Label());
        this->label6 = (gcnew System::Windows::Forms::Label());
        this->label7 = (gcnew System::Windows::Forms::Label());
        this->label8 = (gcnew System::Windows::Forms::Label());
        this->timer3 = (gcnew System::Windows::Forms::Timer(this-
>components));
```

```

        this->timer4 = (gcnew System::Windows::Forms::Timer(this-
>components));

        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox1))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox2))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox3))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox4))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox5))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox6))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox7))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox8))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox9))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox10))->BeginInit();
        this->SuspendLayout();
        //
        // pictureBox1
        //
        this->pictureBox1->Location = System::Drawing::Point(263, 88);
        this->pictureBox1->Name = L"pictureBox1";
        this->pictureBox1->Size = System::Drawing::Size(100, 150);
        this->pictureBox1->TabIndex = 0;
        this->pictureBox1->TabStop = false;
        this->pictureBox1->Click += gcnew System::EventHandler(this,
&MyForm::pictureBox1_Click);
        //
        // pictureBox2
        //
        this->pictureBox2->Location = System::Drawing::Point(369, 88);
        this->pictureBox2->Name = L"pictureBox2";
        this->pictureBox2->Size = System::Drawing::Size(100, 150);
        this->pictureBox2->TabIndex = 0;
        this->pictureBox2->TabStop = false;
        this->pictureBox2->Click += gcnew System::EventHandler(this,
&MyForm::pictureBox2_Click);
        //
        // pictureBox3
        //

```

```

        this->pictureBox3->Location = System::Drawing::Point(475, 88);
        this->pictureBox3->Name = L"pictureBox3";
        this->pictureBox3->Size = System::Drawing::Size(100, 150);
        this->pictureBox3->TabIndex = 0;
        this->pictureBox3->TabStop = false;
        this->pictureBox3->Click += gcnew System::EventHandler(this,
&MyForm::pictureBox3_Click);
        //
        // pictureBox4
        //
        this->pictureBox4->Location = System::Drawing::Point(581, 88);
        this->pictureBox4->Name = L"pictureBox4";
        this->pictureBox4->Size = System::Drawing::Size(100, 150);
        this->pictureBox4->TabIndex = 0;
        this->pictureBox4->TabStop = false;
        this->pictureBox4->Click += gcnew System::EventHandler(this,
&MyForm::pictureBox4_Click);
        //
        // pictureBox5
        //
        this->pictureBox5->Location = System::Drawing::Point(687, 88);
        this->pictureBox5->Name = L"pictureBox5";
        this->pictureBox5->Size = System::Drawing::Size(100, 150);
        this->pictureBox5->TabIndex = 0;
        this->pictureBox5->TabStop = false;
        this->pictureBox5->Click += gcnew System::EventHandler(this,
&MyForm::pictureBox5_Click);
        //
        // pictureBox6
        //
        this->pictureBox6->Location = System::Drawing::Point(263, 244);
        this->pictureBox6->Name = L"pictureBox6";
        this->pictureBox6->Size = System::Drawing::Size(100, 150);
        this->pictureBox6->TabIndex = 0;
        this->pictureBox6->TabStop = false;
        this->pictureBox6->Click += gcnew System::EventHandler(this,
&MyForm::pictureBox6_Click);
        //
        // pictureBox7
        //
        this->pictureBox7->Location = System::Drawing::Point(369, 244);
        this->pictureBox7->Name = L"pictureBox7";
        this->pictureBox7->Size = System::Drawing::Size(100, 150);
        this->pictureBox7->TabIndex = 0;
        this->pictureBox7->TabStop = false;

```

```

        this->pictureBox7->Click += gcnew System::EventHandler(this,
&MyForm::pictureBox7_Click);
        //
        // pictureBox8
        //
        this->pictureBox8->Location = System::Drawing::Point(475, 244);
        this->pictureBox8->Name = L"pictureBox8";
        this->pictureBox8->Size = System::Drawing::Size(100, 150);
        this->pictureBox8->TabIndex = 0;
        this->pictureBox8->TabStop = false;
        this->pictureBox8->Click += gcnew System::EventHandler(this,
&MyForm::pictureBox8_Click);
        //
        // pictureBox9
        //
        this->pictureBox9->Location = System::Drawing::Point(581, 244);
        this->pictureBox9->Name = L"pictureBox9";
        this->pictureBox9->Size = System::Drawing::Size(100, 150);
        this->pictureBox9->TabIndex = 0;
        this->pictureBox9->TabStop = false;
        this->pictureBox9->Click += gcnew System::EventHandler(this,
&MyForm::pictureBox9_Click);
        //
        // pictureBox10
        //
        this->pictureBox10->Location = System::Drawing::Point(687, 244);
        this->pictureBox10->Name = L"pictureBox10";
        this->pictureBox10->Size = System::Drawing::Size(100, 150);
        this->pictureBox10->TabIndex = 0;
        this->pictureBox10->TabStop = false;
        this->pictureBox10->Click += gcnew System::EventHandler(this,
&MyForm::pictureBox10_Click);
        //
        // label1
        //
        this->label1->AutoSize = true;
        this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 20, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->label1->Location = System::Drawing::Point(448, 28);
        this->label1->Name = L"label1";
        this->label1->Size = System::Drawing::Size(161, 31);
        this->label1->TabIndex = 1;
        this->label1->Text = L"Найди пару";
        //
        // button1

```

```

        //
        this->button1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 12, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
            static_cast<System::Byte>(204)));
        this->button1->Location = System::Drawing::Point(488, 414);
        this->button1->Name = L"button1";
        this->button1->Size = System::Drawing::Size(75, 32);
        this->button1->TabIndex = 2;
        this->button1->Text = L"Цапт";
        this->button1->UseVisualStyleBackColor = true;
        this->button1->Click += gcnew System::EventHandler(this,
&MyForm::Button1_Click);
        //
        // label2
        //
        this->label2->AutoSize = true;
        this->label2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 20, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
            static_cast<System::Byte>(204)));
        this->label2->Location = System::Drawing::Point(27, 487);
        this->label2->Name = L"label2";
        this->label2->Size = System::Drawing::Size(0, 31);
        this->label2->TabIndex = 3;
        //
        // timer1
        //
        this->timer1->Interval = 1000;
        this->timer1->Tick += gcnew System::EventHandler(this,
&MyForm::timer1_Tick);
        //
        // label3
        //
        this->label3->AutoSize = true;
        this->label3->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 20, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
            static_cast<System::Byte>(204)));
        this->label3->Location = System::Drawing::Point(874, 28);
        this->label3->Name = L"label3";
        this->label3->Size = System::Drawing::Size(29, 31);
        this->label3->TabIndex = 4;
        this->label3->Text = L"0";
        //
        // label4
        //
        this->label4->AutoSize = true;

```

```

        this->label4->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 20, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->label4->Location = System::Drawing::Point(772, 28);
        this->label4->Name = L"label4";
        this->label4->Size = System::Drawing::Size(96, 31);
        this->label4->TabIndex = 5;
        this->label4->Text = L"Время";
        //
        // timer2
        //
        this->timer2->Interval = 10;
        this->timer2->Tick += gcnew System::EventHandler(this,
&MyForm::timer2_Tick);
        //
        // label5
        //
        this->label5->AutoSize = true;
        this->label5->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 20, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->label5->Location = System::Drawing::Point(24, 543);
        this->label5->Name = L"label5";
        this->label5->Size = System::Drawing::Size(0, 31);
        this->label5->TabIndex = 6;
        //
        // label6
        //
        this->label6->AutoSize = true;
        this->label6->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 20, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->label6->Location = System::Drawing::Point(469, 500);
        this->label6->Name = L"label6";
        this->label6->Size = System::Drawing::Size(0, 31);
        this->label6->TabIndex = 7;
        //
        // label7
        //
        this->label7->AutoSize = true;
        this->label7->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 20, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->label7->Location = System::Drawing::Point(469, 556);
        this->label7->Name = L"label7";
        this->label7->Size = System::Drawing::Size(0, 31);

```

```

this->label7->TabIndex = 8;
//
// label8
//
this->label8->AutoSize = true;
this->label8->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 20, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
this->label8->Location = System::Drawing::Point(579, 500);
this->label8->Name = L"label8";
this->label8->Size = System::Drawing::Size(0, 31);
this->label8->TabIndex = 9;
//
// timer3
//
this->timer3->Interval = 1000;
this->timer3->Tick += gcnew System::EventHandler(this,
&MyForm::Timer3_Tick);
//
// timer4
//
this->timer4->Tick += gcnew System::EventHandler(this,
&MyForm::Timer4_Tick);
//
// MyForm
//
this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(1044, 638);
this->Controls->Add(this->label8);
this->Controls->Add(this->label7);
this->Controls->Add(this->label6);
this->Controls->Add(this->label5);
this->Controls->Add(this->label4);
this->Controls->Add(this->label3);
this->Controls->Add(this->label2);
this->Controls->Add(this->button1);
this->Controls->Add(this->label1);
this->Controls->Add(this->pictureBox10);
this->Controls->Add(this->pictureBox5);
this->Controls->Add(this->pictureBox9);
this->Controls->Add(this->pictureBox4);
this->Controls->Add(this->pictureBox8);
this->Controls->Add(this->pictureBox3);
this->Controls->Add(this->pictureBox7);
this->Controls->Add(this->pictureBox2);

```

```

        this->Controls->Add(this->pictureBox6);
        this->Controls->Add(this->pictureBox1);
        this->Name = L"MyForm";
        this->Text = L"MyForm";
        this->Load += gcnew System::EventHandler(this, &MyForm::MyForm_Load);
        (cli::safe_cast<System::ComponentModel::ISupportInitialize>(this->pictureBox1))->EndInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize>(this->pictureBox2))->EndInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize>(this->pictureBox3))->EndInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize>(this->pictureBox4))->EndInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize>(this->pictureBox5))->EndInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize>(this->pictureBox6))->EndInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize>(this->pictureBox7))->EndInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize>(this->pictureBox8))->EndInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize>(this->pictureBox9))->EndInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize>(this->pictureBox10))->EndInit();
        this->ResumeLayout(false);
        this->PerformLayout();

    }
#pragma endregion

    void proverka()
    {
        s = 0;
        for (int i = 0; i < 10; i++)
        {
            if (mas1[i] == 1) s++;
        }
        if (s == 2 && m == 1)
        {
            for (int i = 0; i < 10; i++)
            {
                if (mas1[i] == 1) { k1 = i; break; }
            }
            for (int i = 9; i > -1; i--)
            {
                if (mas1[i] == 1) { k2 = i; break; }
            }
        }
    }

```

```

        }
        if (mas[k1] == mas[k2])
        {
            Controls["pictureBox" + Convert::ToString(k1 + 1)]->Visible = false;
            Controls["pictureBox" + Convert::ToString(k2 + 1)]->Visible = false;
            t++;
        }
        else
        {
            for (int i = 1; i < 11; i++)
                Controls["pictureBox" + Convert::ToString(i)]->BackgroundImage = Image::FromFile("fon.jpg");
            for (int i = 0; i < 10; i++)
                mas1[i] = 0;
        }
        for (int i = 0; i < 10; i++)
        {
            mas1[i] = 0;
        }
    }
    if (t == 5)
    {
        timer2->Stop();
    }
}

```

```

private: System::Void MyForm_Load(System::Object^ sender, System::EventArgs^ e)
{
    for (int i = 1; i < 11; i++)
        Controls["pictureBox" + Convert::ToString(i)]->BackgroundImage = Image::FromFile("fon.jpg");
}

```

```

private: System::Void Button1_Click(System::Object^ sender, System::EventArgs^ e)
{
    f = 0, k = 0, Time = 0, a1 = 0, a2 = 0, a3 = 0, a4 = 0, a5 = 0, m = 1;
    s = 0; x = 0, t = 0;

    for (int i = 0; i < 10; i++)
    {
        mas[i] = 0;
        mas1[i] = 0;
        Controls["pictureBox" + Convert::ToString(i + 1)]->Visible = true;
    }
}

```

```

timer1->Enabled = false;
srand(time(0));

for (int q = 0; q < 10; q++)
{
    z = rand() % 5 + 1;
    if (a1 != 3 && z == 1) a1++;
    if (a2 != 3 && z == 2) a2++;
    if (a3 != 3 && z == 3) a3++;
    if (a4 != 3 && z == 4) a4++;
    if (a5 != 3 && z == 5) a5++;
    f = 0;
    for (int i = 0; i < 10; i++)
    {
        if (z == 1 && a1 == 3) { q--; break; }
        if (z == 2 && a2 == 3) { q--; break; }
        if (z == 3 && a3 == 3) { q--; break; }
        if (z == 4 && a4 == 3) { q--; break; }
        if (z == 5 && a5 == 3) { q--; break; }
        if (mas[i] == 0 && f == 0)
        {
            k++;
            mas[i] = z;
            Controls["pictureBox" + Convert::ToString(k)]->BackgroundImage
= Image::FromFile(z + ".jpg");
            f = 1; break;
        }
    }
}
timer1->Enabled = true;
}

private: System::Void timer1_Tick(System::Object^ sender, System::EventArgs^ e)
{
    timer1->Enabled = false;
    for (int i = 1; i < 11; i++)
        Controls["pictureBox" + Convert::ToString(i)]->BackgroundImage =
Image::FromFile("fon.jpg");
    timer2->Enabled = true;
}

private: System::Void timer2_Tick(System::Object^ sender, System::EventArgs^ e)
{

```

```

        Time += 0.00001 * timer1->Interval;
        label3->Text = String::Format("{0:F2}", Time);
    }

private: System::Void pictureBox1_Click(System::Object^ sender, System::EventArgs^ e)
{
    if (m == 1)
    {
        mas1[0] = 1;
        Controls["pictureBox" + Convert::ToString(1)]->BackgroundImage =
Image::FromFile(Convert::ToString(mas[0]) + ".jpg");
    }
    timer3->Stop();
    timer3->Enabled = true;
    timer3->Start();
}

private: System::Void pictureBox2_Click(System::Object^ sender, System::EventArgs^ e)
{
    if (m == 1)
    {
        mas1[1] = 1;
        Controls["pictureBox" + Convert::ToString(2)]->BackgroundImage =
Image::FromFile(Convert::ToString(mas[1]) + ".jpg");
    }
    timer3->Stop();
    timer3->Enabled = true;
    timer3->Start();
}

private: System::Void pictureBox3_Click(System::Object^ sender, System::EventArgs^ e)
{
    if (m == 1)
    {
        mas1[2] = 1;
        Controls["pictureBox" + Convert::ToString(3)]->BackgroundImage =
Image::FromFile(Convert::ToString(mas[2]) + ".jpg");
    }
    timer3->Stop();
    timer3->Enabled = true;
    timer3->Start();
}

private: System::Void pictureBox4_Click(System::Object^ sender, System::EventArgs^ e)
{
    if (m == 1)

```

```

        {
            mas1[3] = 1;
            Controls["pictureBox" + Convert::ToString(4)]->BackgroundImage =
Image::FromFile(Convert::ToString(mas[3]) + ".jpg");
        }
        timer3->Stop();
        timer3->Enabled = true;
        timer3->Start();
    }
private: System::Void pictureBox5_Click(System::Object^ sender, System::EventArgs^ e)
{
    if (m == 1)
    {
        mas1[4] = 1;
        Controls["pictureBox" + Convert::ToString(5)]->BackgroundImage =
Image::FromFile(Convert::ToString(mas[4]) + ".jpg");
    }
    timer3->Stop();
    timer3->Enabled = true;
    timer3->Start();
}
private: System::Void pictureBox6_Click(System::Object^ sender, System::EventArgs^ e)
{
    if (m == 1)
    {
        mas1[5] = 1;
        Controls["pictureBox" + Convert::ToString(6)]->BackgroundImage =
Image::FromFile(Convert::ToString(mas[5]) + ".jpg");
    }
    timer3->Stop();
    timer3->Enabled = true;
    timer3->Start();
}
private: System::Void pictureBox7_Click(System::Object^ sender, System::EventArgs^ e)
{
    if (m == 1)
    {
        mas1[6] = 1;
        Controls["pictureBox" + Convert::ToString(7)]->BackgroundImage =
Image::FromFile(Convert::ToString(mas[6]) + ".jpg");
    }
    timer3->Stop();
    timer3->Enabled = true;
    timer3->Start();
}
private: System::Void pictureBox8_Click(System::Object^ sender, System::EventArgs^ e)

```

```

{
    if (m == 1)
    {
        mas1[7] = 1;
        Controls["pictureBox" + Convert::ToString(8)]->BackgroundImage =
Image::FromFile(Convert::ToString(mas[7]) + ".jpg");
    }
    timer3->Stop();
    timer3->Enabled = true;
    timer3->Start();
}
private: System::Void pictureBox9_Click(System::Object^ sender, System::EventArgs^ e)
{
    if (m == 1)
    {
        mas1[8] = 1;
        Controls["pictureBox" + Convert::ToString(9)]->BackgroundImage =
Image::FromFile(Convert::ToString(mas[8]) + ".jpg");
    }
    timer3->Stop();
    timer3->Enabled = true;
    timer3->Start();
}
private: System::Void pictureBox10_Click(System::Object^ sender, System::EventArgs^ e)
{
    if (m == 1)
    {
        mas1[9] = 1;
        Controls["pictureBox" + Convert::ToString(10)]->BackgroundImage =
Image::FromFile(Convert::ToString(mas[9]) + ".jpg");
    }
    timer3->Stop();
    timer3->Enabled = true;
    timer3->Start();
}

private: System::Void Timer3_Tick(System::Object^ sender, System::EventArgs^ e)
{
    proverka();
    timer3->Stop();
    timer3->Enabled = false;
}

private: System::Void Timer4_Tick(System::Object^ sender, System::EventArgs^ e)
{
}

```

```
};  
}
```

```
//label12->Text = "";      label6->Text = "";  label7->Text = "";  label8->Text = "";  
//label12->Text += Convert::ToString(mas1[i]) + " ";  
//label18->Text = Convert::ToString(s);  
//label6->Text = Convert::ToString(k1);  
//label7->Text = Convert::ToString(k2);
```

Тестовые испытания

