**Тема 8: «Разработка игрового приложения»**

**Цель работы:** получить практические навыки работы с файловой системой и разработки игрового приложения с использованиемтаймера.

**Вариант 5.**

Съедобные и несъедобные грибы. Викторина выводит в случайном порядке изображения съедобных и несъедобных грибов, требуется выбрать название гриба и съедобный он или же нет. Варианты ответов в виде падающего списка.

**Задание 1**:

Разработать проект «Слайд-шоу». Программа должна листать картинки вперёд и назад от начала до конца списка и отображать картинку по выбору пользователя, а также поочередно отображать картинки в режиме слайд-шоу при помощи таймера. Картинки разместить во вложенной папке Images. Список имён графических файлов должен формироваться автоматически в программе при помощи команд поиска директории и поиска файлов с заданным расширением.

**Задание 2:**

Добавьте в проект викторину. Требуется угадывать картинки, которые программа предлагает в случайном порядке. Ограничить время викторины с помощью таймера. Варианты ответов выводить в виде зависимых кнопок выбора и сформировать следующим образом: один верный и еще несколько вариантов из списка (без повторений). По окончании викторины вычислить процент правильных ответов и вывести результат по 100-балльной системе.

**Задание 3:**

Добавьте в проект настройку параметров: времени сеанса викторины, цвета формы. Настройки должны сохраняться в параметрах проекта и применяться при новом запуске приложения. Также сохраняйте в параметрах проекта лучший рекорд пользователя

1. **Разработка подпрограмм**
   1. **Код модуля Form1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Windows.Forms;

namespace \_7lab

{

public partial class Form1 : Form

{

private List<string> img;

private int ind = 0;

private Timer tim;

private Timer quizTimer;

private Random random = new Random();

private int correctAnswers = 0;

private int totalQuestions = 0;

private DateTime quizStartTime;

private int score = 0;

private int timeLeft = 60; // in seconds

private List<Mushroom> mushrooms = new List<Mushroom>();

private Mushroom currentMushroom;

int quizDuration = 60;

int bestScore = 0;

public Form1()

{

InitializeComponent();

LoadImageFiles();

init\_timer();

InitializeQuizTimer();

button3.Visible = false;

label3.Visible = false;

LoadSettings();

}

private void LoadImageFiles()

{

string imagesPath = Path.Combine(Application.StartupPath, "images");

img = Directory.GetFiles(imagesPath, "\*.jpeg").Concat(Directory.GetFiles(imagesPath, "\*.jpg")).ToList();

if (img.Any())

{

DisplayImage(0);

}

else

{

MessageBox.Show("Изображение не найдено");

}

}

private void DisplayImage(int index)

{

if (img.Any())

{

pictureBox1.Image = Image.FromFile(img[index]);

}

}

private void init\_timer()

{

tim = new Timer();

tim.Interval = 2000;

tim.Tick += tim\_tick;

}

private void tim\_tick(object sender, EventArgs e)

{

ind = (ind + 1) % img.Count;

DisplayImage(ind);

}

private void button\_prev\_Click(object sender, EventArgs e)

{

ind = (ind - 1 + img.Count) % img.Count;

DisplayImage(ind);

}

private void button\_next\_Click(object sender, EventArgs e)

{

ind = (ind + 1) % img.Count;

DisplayImage(ind);

}

private void button\_strslsh\_Click(object sender, EventArgs e)

{

tim.Start();

}

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

{

timeLeft--;

label3.Text = $"Осталось времени: {timeLeft} сек";

if (timeLeft <= 0)

{

EndQuiz();

}

}

private void EndQuiz()

{

quizTimer.Stop();

int result = (score \* 100) / 10;

if (score > bestScore)

{

bestScore = score;

SaveSettings();

}

MessageBox.Show($"Ваша оценка: {result} \nЛучший результат: {bestScore \*10}");

groupBox1.Visible = false;

button\_next.Visible = true;

button\_prev.Visible = true;

button\_strslsh.Visible = true;

button1.Visible = true;

button4.Visible = true;

button2.Visible = true;

label3 .Visible = false;

button3.Visible = false;

button5.Visible = true;

}

private void InitializeQuizTimer()

{

quizTimer = new Timer();

quizTimer.Interval = 1000; // 1 second

quizTimer.Tick += QuizTimer\_Tick;

}

private void button1\_Click\_1(object sender, EventArgs e)

{

using (OpenFileDialog openFileDialog = new OpenFileDialog())

{

openFileDialog.InitialDirectory = Path.Combine(Application.StartupPath, "images");

openFileDialog.Filter = "Image Files (\*.jpg; \*.jpeg)|\*.jpg; \*.jpeg";

if (openFileDialog.ShowDialog() == DialogResult.OK)

{

pictureBox1.Image = Image.FromFile(openFileDialog.FileName);

}

}

}

private void NextQuestion()

{

if (totalQuestions == 0)

{

EndQuiz();

return;

}

totalQuestions--;\

int index = random.Next(mushrooms.Count);

currentMushroom = mushrooms[index];

pictureBox1.Image = Image.FromFile(currentMushroom.ImagePath);

var answers = mushrooms.Select(m => m.Name).OrderBy(x => random.Next()).Take(4).ToList();

if (!answers.Contains(currentMushroom.Name))

{

answers[random.Next(4)] = currentMushroom.Name;

}

comboBox1.Items.Clear();

comboBox1.Items.AddRange(answers.ToArray());

comboBox1.SelectedIndex = 0;

radioButton1.Checked = false;

radioButton2.Checked = false;

}

private void button2\_Click(object sender, EventArgs e)

{

groupBox1.Visible = true;

button\_next.Visible = false;

button\_prev.Visible = false;

button\_strslsh.Visible = false;

button1.Visible = false;

button3.Visible = true;

button4.Visible = false;

button5.Visible = false;

label3.Visible = true;

LoadMushrooms();

totalQuestions = 10;

score = 0;

timeLeft = quizDuration;

quizTimer.Start();

NextQuestion();

tim.Stop();

button2.Visible= false;

}

private void LoadMushrooms()

{

mushrooms.Add(new Mushroom { Name = "Белый гриб", IsEdible = true, ImagePath = "images/1.jpg" });

mushrooms.Add(new Mushroom { Name = "Шампиньон", IsEdible = true, ImagePath = "images/2.jpeg" });

mushrooms.Add(new Mushroom { Name = "Лисичка", IsEdible = true, ImagePath = "images/3.jpg" });

mushrooms.Add(new Mushroom { Name = "Сатанинский гриб", IsEdible = false, ImagePath = "images/10.jpeg" });

mushrooms.Add(new Mushroom { Name = "Подберезовик", IsEdible = true, ImagePath = "images/9.jpeg" });

mushrooms.Add(new Mushroom { Name = "Ложные опята", IsEdible = false, ImagePath = "images/8.jpeg" });

mushrooms.Add(new Mushroom { Name = "Мухомор", IsEdible = false, ImagePath = "images/7.jpg" });

mushrooms.Add(new Mushroom { Name = "Бледная поганка", IsEdible = false, ImagePath = "images/6.jpeg" });

mushrooms.Add(new Mushroom { Name = "Рыжик", IsEdible = true, ImagePath = "images/5.jpg" });

mushrooms.Add(new Mushroom { Name = "Ленин", IsEdible = false, ImagePath = "images/4.jpg" });

}

private void button3\_Click(object sender, EventArgs e)

{

bool isCorrect = (comboBox1.SelectedItem.ToString() == currentMushroom.Name) && ((radioButton1.Checked && currentMushroom.IsEdible) || (radioButton2.Checked && !currentMushroom.IsEdible));

if (isCorrect)

{

score++;

}

NextQuestion();

}

private void button4\_Click(object sender, EventArgs e)

{

tim.Stop();

}

private void LoadSettings()

{

quizDuration = Properties.Settings.Default.QuizTime;

this.BackColor = Properties.Settings.Default.FormColor;

bestScore = Properties.Settings.Default.BestScore;

}

private void SaveSettings()

{

Properties.Settings.Default.QuizTime = quizDuration;

Properties.Settings.Default.FormColor = this.BackColor;

Properties.Settings.Default.BestScore = bestScore;

Properties.Settings.Default.Save();

}

private void button5\_Click(object sender, EventArgs e)

{

using (var settingsForm = new Form2(quizDuration, this.BackColor))

{

if (settingsForm.ShowDialog() == DialogResult.OK)

{

quizDuration = settingsForm.QuizTime;

this.BackColor = settingsForm.FormColor;

SaveSettings();

}

}

}

}

public class Mushroom

{

public string Name { get; set; }

public bool IsEdible { get; set; }

public string ImagePath { get; set; }

}

}

**1.2. Код модуля Form2**

using System;

using System.Drawing;

using System.Windows.Forms;

namespace \_7lab

{

public partial class Form2 : Form

{

public int QuizTime { get; private set; }

public Color FormColor { get; private set; }

public Form2(int quizTime, Color formColor)

{

InitializeComponent();

numericUpDown1.Value = quizTime;

colorDialog1.Color = formColor;

}

private void button1\_Click(object sender, EventArgs e)

{

QuizTime = (int)numericUpDown1.Value;

FormColor = colorDialog1.Color;

DialogResult = DialogResult.OK;

}

private void button2\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() == DialogResult.OK)

{

panel1.BackColor = colorDialog1.Color;

}

}

}

}

**1.2.1** Форма сохраняет настройки параметров. Добавление параметров в приложение рисунок 1.1.

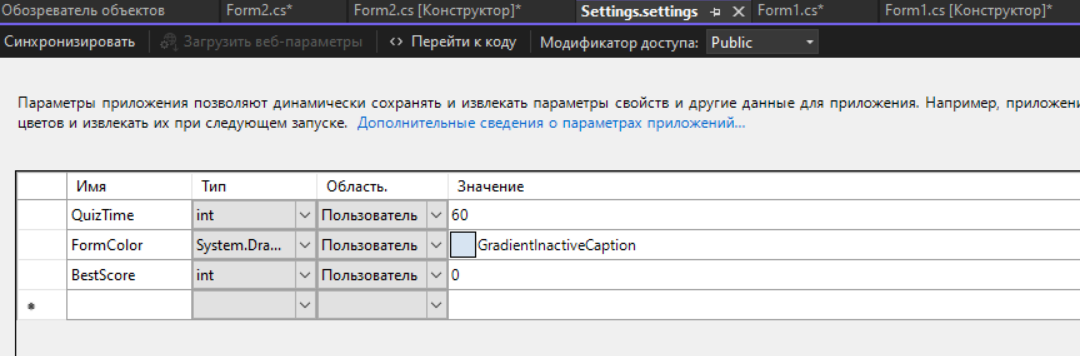
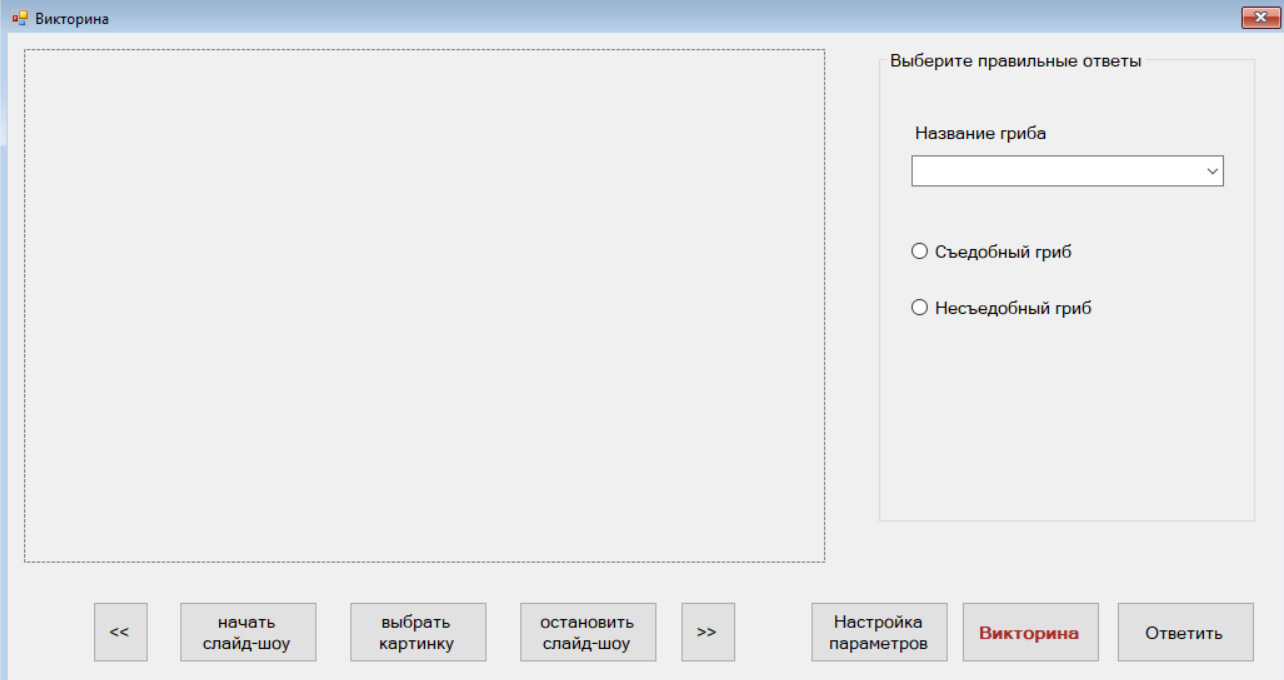


Рис. 1.1. – добавление параметров к приложению

**1.3. Создание пользовательского интерфейса**

Основная форма представлена на рис. 1.2.



button\_next

radiobutton2

radiobutton1

groupBox1

comboBox1

label1

button3

button2

button5

button4

button1

button\_str

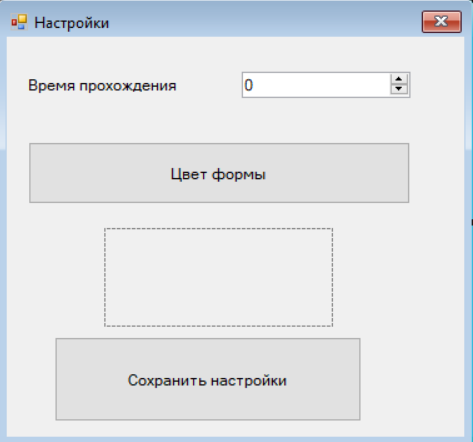
button\_pr

pictureBox1

Form1

Рис.1.2 – Form1

Интерфейс фомы настроек представлен на рис. 1.3



button2

panel1

button1

numericUpDowm1

label1

Form2

Рис. 1.3 – Form2

* 1. **Свойства компонентов формы**

*Таблица 1.1.* Значения свойств, установленные на этапе конструирования интерфейса Form1

|  |  |  |
| --- | --- | --- |
| **Название компонента** | **Свойства** | **Значения** |
| form1 | Text | Викторина |
| Font | 9 |
| Начертание: | обычный |
| button1 | Text | «Выбрать картинку» |
| button2 | Text | «Викторина» |
| ForeColor | Brown |
| button3 | Text | «Ответить» |
| button4 | Text | «Остановить слайд-шоу» |
| button5 | Text | «Настройка параметров» |
| button\_pr | Text | «<<» |
| button\_str | Text | «Начать слайд-шоу» |
| button\_next | Text | «>>» |
| groupBox1 | Text | «Выберите правильные ответы» |
| Radiobutton1 | Text | «Съедобный гриб» |
| RadioButton2 | Text | «Несъедобный гриб» |

*Таблица 1.2.* Значения свойств, установленные на этапе конструирования интерфейса Form2

|  |  |  |
| --- | --- | --- |
| **Название компонента** | **Свойства** | **Значения** |
| form1 | Text | Настройки |
| Font | 9 |
| Начертание: | обычный |
| button1 | Text | «Цвет формы» |
| button2 | Text | «Сохранить настройки» |

* 1. **Сценарий использования программы**
     1. Запуск программы.
     2. Выбор кнопки «Начать слайд-шоу»
        1. Начинается слайд-шоу в pictureBox1
     3. Выбор кнопки «остановить слайд-шоу».
        1. Остановка слайд-шоу.
     4. Выбор кнопки «выбрать картинку».
        1. Открытие диалогового окна, пользователь выбирает файл с картинкой.
        2. Отображение картинки в pictureBox1.
     5. Кнопки «<<» и «>>» перелистывают на изображение назад и вперед соответственно.
     6. Выбор кнопки «Викторина».
        1. Появление поля с вопросами и вариантами ответов.
        2. Пользователь отвечает на вопрос и нажимает на кнопку «Ответить», регистрируя ответ.
        3. В конце игры появляется сообщение с результатом.
     7. Выбор кнопки «Настройка параметров».
        1. Открытие формы с настройками.
        2. Пользователь устанавливает свои настройки, которые будут применены к приложению в дальнейшем.
     8. Возможность вернуться к предыдущим пунктам.
     9. Завершение работы с программой.
  2. **Протокол испытаний**

*Таблица 2.1. протокол испытаний*

|  |  |  |  |
| --- | --- | --- | --- |
| **Проверяемые требования** | **Сообщения программы и вводимые значения** | **Ожидаемые результаты** | **Фактические результаты** |
| Способность программы обеспечить ввод данных и сохранение введённых данных во время работы программы | Открытие диалогового окна и поиск файлов с заданным расширением (.jpg, .jpeg) | Открытие папки «images» и отображения для выбора файлов с указанным расширением. | Соответствует ожидаемому результату (см. Рисунки 2.1.1) |
| Способность программы обеспечить просмотр слайд-шоу | Нажать на кнопку «начать слайд-шоу» | С интервалом 2 секунды в pictureBox1 меняются изображения | Соответствует ожидаемому результату (см. Рисунок 2.1.2) |
| Способность программы обеспечить остановку слайд-шоу | Нажать на кнопку «остановить слайд-шоу» | Слайд-шоу останавливается | Соответствует ожидаемому результату |
| Способность программы обеспечить проведение викторины на тему «Съедобные / несъедобные грибы» | Нажать на кнопку «Викторина». | Появление groupBox1 с вопросами и ответами. Отображение оставшегося времени до конца викторины | Соответствует ожидаемому результату (см. Рисунки 2.1.3) |
| Способность программы обеспечивать регистрацию результатов ответов, для подсчета баллов. | Нажать на кнопку «Ответить». После заполнения полей с ответами на вопросы | После окончания раунда выводится MessageBox с результатом по 100-балльной шкале | Соответствует ожидаемому результату (см. Рисунки 2.1.4) |
| Способность программы обеспечивать регистрацию лучших результатов прохождения викторины. | Прохождение викторины несколько раз. | После окончания раунда выводится MessageBox с результатом прошедшей игры и лучший результат из всех игр по 100-балльной шкале | Соответствует ожидаемому результату (см. Рисунки 2.1.5) |

* 1. **Результаты тестировния проекта**

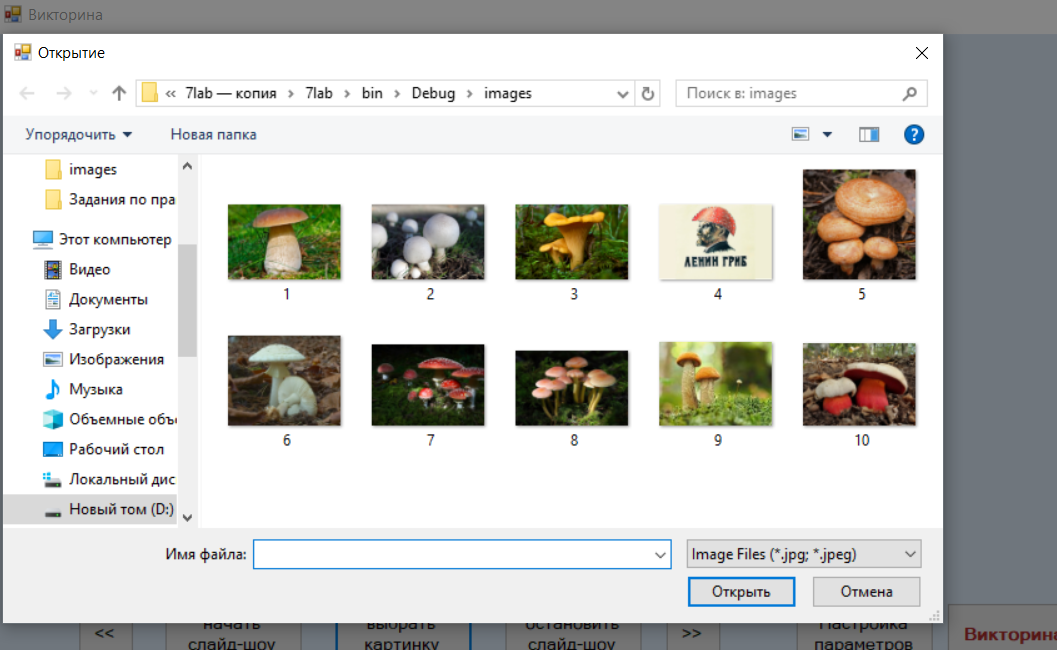
****

Рис. 2.1.1

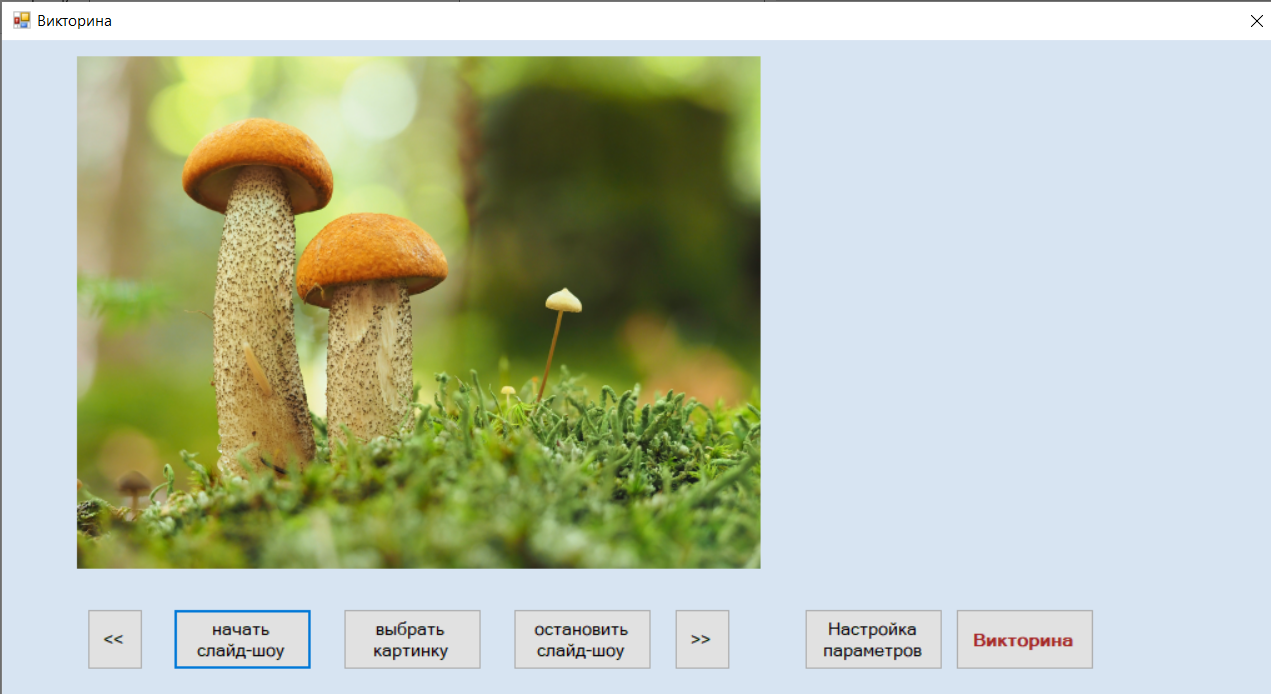
****

Рис.2.1.2

****

Рис.2.1.3

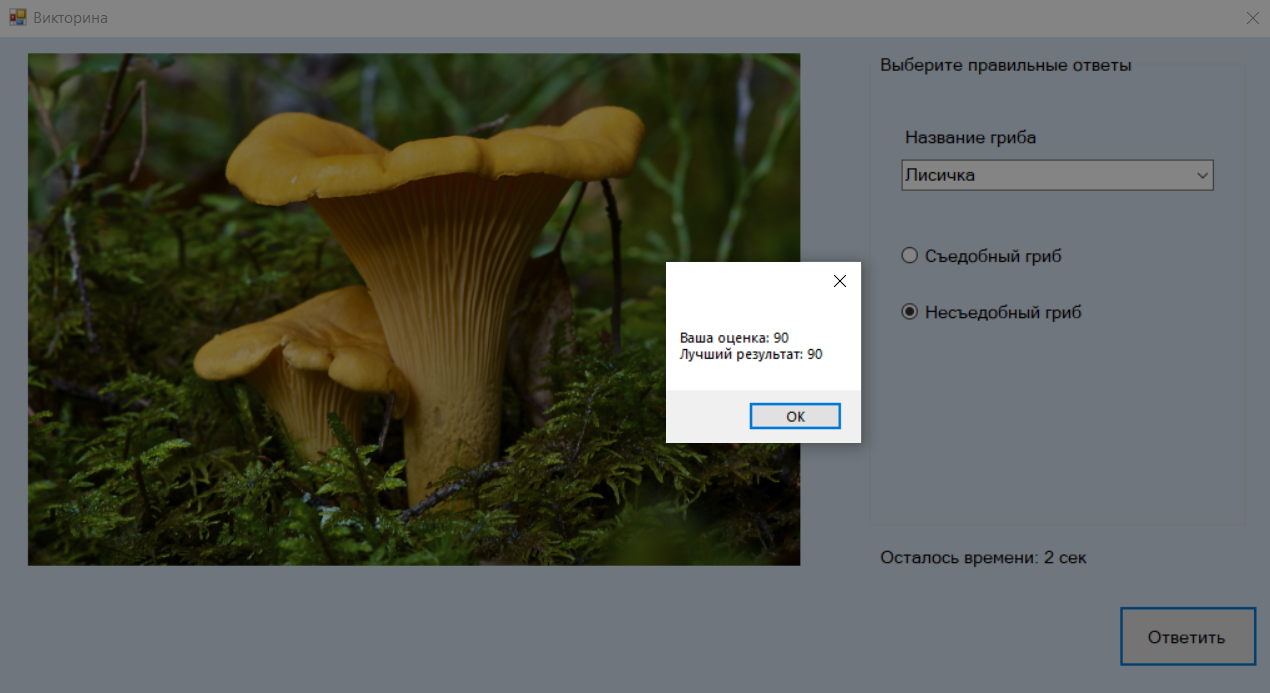
****

Рис.2.1.4

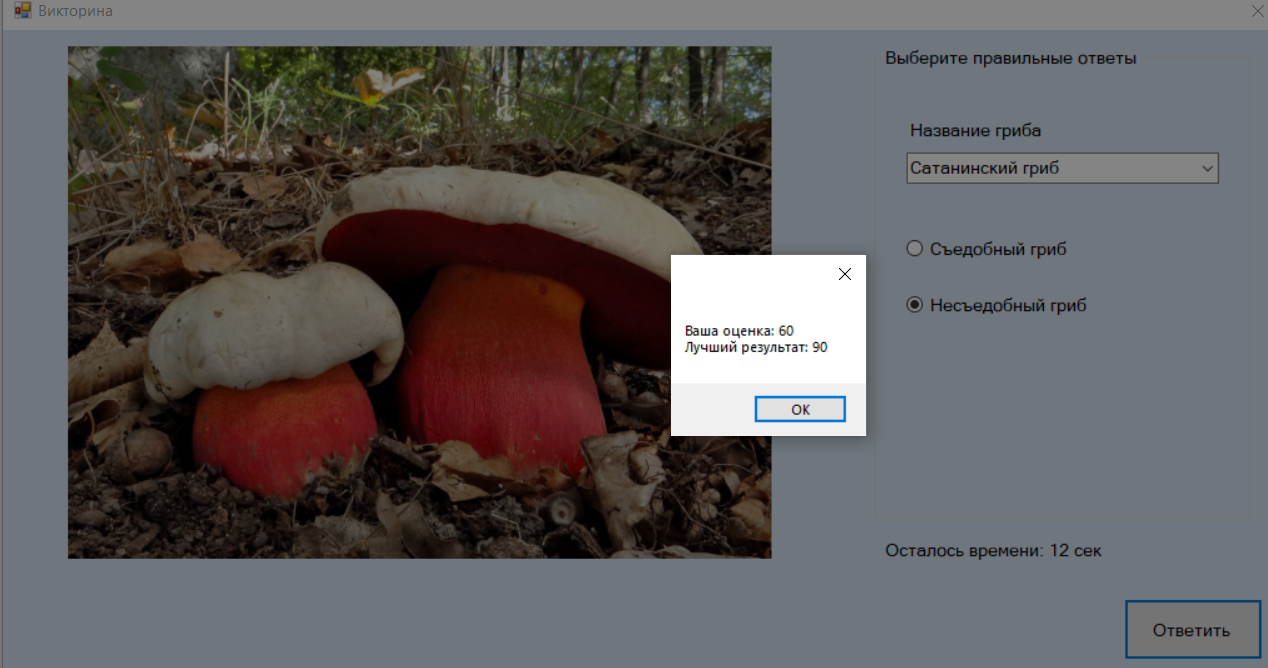
****

Рис.2.1.5