**Задание:**

Реализовать настольное приложение по мотивам телевизионного шоу «Кто хочет стать миллионером?»

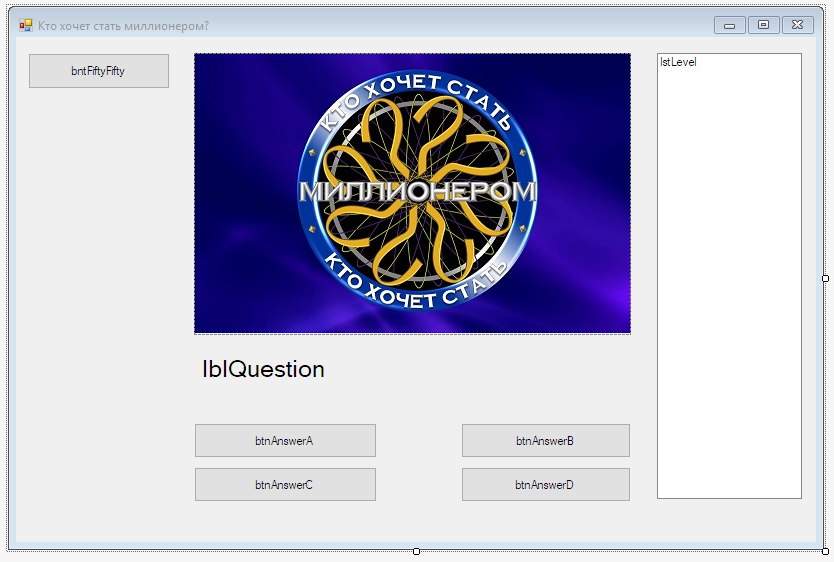
**Правила игры:** Для победы в игре игроку необходимо верно ответить на 15 вопросов из различных областей знаний. Каждый вопрос имеет 4 варианта ответа, из которых только один является верным. Сложность вопросов постоянно возрастает. Время на раздумье над каждым вопросом у игрока не ограничено.

Каждый из пятнадцати вопросов имеет конкретную денежную стоимость: 3 000 000, 1 500 000, 800 000, 400 000, 200 000, 100 000, 50 000, 25 000, 15 000, 10 000, 5 000, 3 000, 2 000, 1 000, 500.

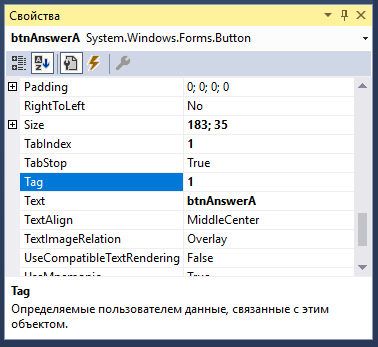
Все суммы являются заменяемыми, то есть, ответив на следующий вопрос не суммируются с суммой за ответ на предыдущий. В игре существует одна несгораемая сумма - её выбирают сами участники перед началом игры. Эта сумма остаётся у игроков даже при неправильном ответе на один из последующих вопросов.

Игроку предлагаются 5 подсказок. Участники могут использовать только четыре из пяти подсказок по ходу игры.

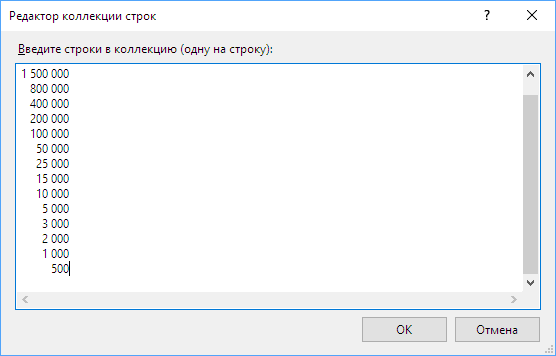
1. «Помощь зала» – каждый зритель в студии голосует за правильный, на его взгляд, ответ, а игроку предоставляется статистика голосования.
2. «50 на 50» – компьютер убирает два неправильных ответа.
3. «Звонок другу» – в течение 30 секунд игрок может посоветоваться с одним из пяти друзей, заявленных заранее.
4. «Право на ошибку» – Игрок может дать два ответа на заданный вопрос. Если первый ответ игрока оказался правильным, подсказка всё равно считается использованной.
5. «Замена вопроса» – игрок может заменить вопрос на другой того же уровня и той же стоимости.
6. Создайте новый проект Windows Forms с именем «WhoWantsToBeAMillionaire».
7. Разместите на форме элементы управления как показано на рисунке ниже.



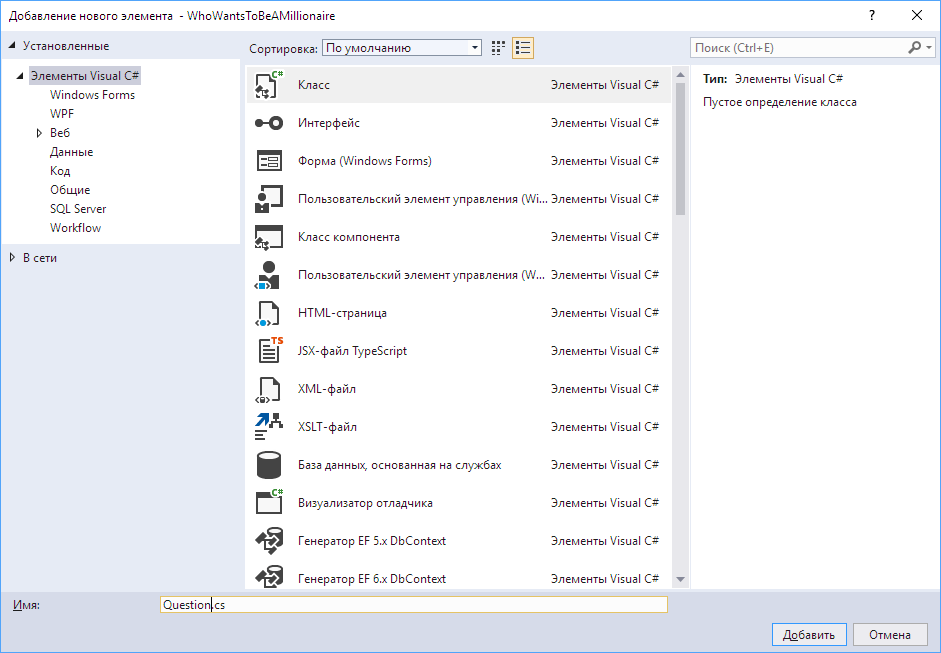
1. Добавьте на форму элемент управления PictureBox и установите его свойство Image изображением из файла «picture.jpg».
2. У кнопок ответов установите свойства Tag значениями «1», «2», «3», «4» соответственно.



1. Добавьте на форму элемент Label для отображения текста вопроса. Задайте его свойство Name в «lblQuestion». Установите свойство AutoSize в False. Настройте шрифт, установив свойство Font.
2. У списка lstLevel установите свойство Items значениями «3 000 000, 1 500 000, 800 000, 400 000, 200 000, 100 000, 50 000, 25 000, 15 000, 10 000, 5 000, 3 000, 2 000, 1 000, 500». Настройте шрифт списка.



1. Добавьте к проекту описание класса Question, выполнив команду «Проект\Добавить класс…».



1. Код класса Question.

namespace WhoWantsToBeAMillionaire

{

class Question

{

public string Text { get; private set; }

public string[] Answers { get; private set; }

public int RightAnswer { get; private set; }

public int Level { get; private set; }

public Question(string[] s)

{

Text = s[0];

Answers = new string[4];

for (int i = 0; i < 4; i++)

Answers[i] = s[i + 1];

RightAnswer = int.Parse(s[5]);

Level = int.Parse(s[6]);

}

}

}

1. Добавьте в класс формы поля.

public partial class Form1 : Form

{

List<Question> questions = new List<Question>();

private Random rnd = new Random();

int level = 0;

Question currentQuestion;

public Form1()

{

InitializeComponent();

}

}

1. Добавьте к коду формы метод чтения вопросов из файла.

private void ReadFile()

{

string path = @"Вопросы.txt";

using (StreamReader sr = new StreamReader(path))

{

string line;

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

{

questions.Add(new Question(line.Split('\t')));

}

}

}

1. Добавьте к коду формы метод отображения вопроса и ответов.

private void ShowQuestion(Question q)

{

lblQuestion.Text = q.Text;

btnAnswerA.Text = q.Answers[0];

btnAnswerB.Text = q.Answers[1];

btnAnswerC.Text = q.Answers[2];

btnAnswerD.Text = q.Answers[3];

}

1. Добавьте к коду формы метод получения вопроса с заданным уровнем сложности.

private Question GetQuestion(int level)

{

var questionsWithLevel = questions.Where(q => q.Level == level).ToList();

return questionsWithLevel[rnd.Next(questionsWithLevel.Count)];

}

1. Добавьте к коду формы метод NextStep

private void NextStep()

{

Button[] btns = new Button[] { btnAnswerA, btnAnswerB,

btnAnswerC, btnAnswerC };

foreach (Button btn in btns)

btn.Enabled = true;

level++;

currentQuestion = GetQuestion(level);

ShowQuestion(currentQuestion);

lstLevel.SelectedIndex = lstLevel.Items.Count - level;

}

1. Добавьте к коду формы метод startGame.

private void startGame()

{

level = 0;

NextStep();

}

1. В конструктор формы добавьте вызов методов.

ReadFile();

startGame();

1. Каждой из четырех кнопок ответов добавьте обработчик события Click.

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

{

Button button = (Button)sender;

if (currentQuestion.RightAnswer == int.Parse(button.Tag.ToString()))

NextStep();

else

{

MessageBox.Show("Неверный ответ!");

startGame();

}

}

1. Добавьте обработчик кнопки подсказки «50/50»

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

{

Button[] btns = new Button[] { btnAnswerA, btnAnswerB,

btnAnswerC, btnAnswerC };

int count = 0;

while (count < 2)

{

int n = rnd.Next(4);

int answer = int.Parse(btns[n].Tag.ToString());

if (answer!=currentQuestion.RightAnswer && btns[n].Enabled)

{

btns[n].Enabled=false;

count++;

}

}

}

1. Итоговй код формы имеет следующий вид.

using System;

using System.Collections.Generic;

using System.Data;

using System.IO;

using System.Linq;

using System.Windows.Forms;

namespace WhoWantsToBeAMillionaire

{

public partial class Form1 : Form

{

List<Question> questions = new List<Question>();

private Random rnd = new Random();

int level = 0;

Question currentQuestion;

public Form1()

{

InitializeComponent();

ReadFile();

startGame();

}

private void ReadFile()

{

string path = @"Вопросы.txt";

using (StreamReader sr = new StreamReader(path))

{

string line;

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

{

questions.Add(new Question(line.Split('\t')));

}

}

}

private void ShowQuestion(Question q)

{

lblQuestion.Text = q.Text;

btnAnswerA.Text = q.Answers[0];

btnAnswerB.Text = q.Answers[1];

btnAnswerC.Text = q.Answers[2];

btnAnswerD.Text = q.Answers[3];

}

private Question GetQuestion(int level)

{

var questionsWithLevel = questions.Where(q => q.Level == level).ToList();

return questionsWithLevel[rnd.Next(questionsWithLevel.Count)];

}

private void NextStep()

{

Button[] btns = new Button[] { btnAnswerA, btnAnswerB, btnAnswerC, btnAnswerC };

foreach (Button btn in btns)

btn.Enabled = true;

level++;

currentQuestion = GetQuestion(level);

ShowQuestion(currentQuestion);

lstLevel.SelectedIndex = lstLevel.Items.Count - level;

}

private void startGame()

{

level = 0;

NextStep();

}

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

{

Button button = (Button)sender;

if (currentQuestion.RightAnswer == int.Parse(button.Tag.ToString()))

NextStep();

else

{

MessageBox.Show("Неверный ответ!");

startGame();

}

}

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

{

Button[] btns = new Button[] { btnAnswerA, btnAnswerB,

btnAnswerC, btnAnswerC };

int count = 0;

while (count < 2)

{

int n = rnd.Next(4);

int answer = int.Parse(btns[n].Tag.ToString());

if (answer!=currentQuestion.RightAnswer && btns[n].Enabled)

{

btns[n].Enabled=false;

count++;

}

}

}

}

}