**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc11760983)

[1 ОБЩАЯ ЧАСТЬ 5](#_Toc11760984)

[1.1 Характеристика места прохождения практики 5](#_Toc11760985)

[1.2 Техническое задание 5](#_Toc11760986)

[1.3 Выбор и обоснование средств реализации БД 9](#_Toc11760987)

[1.4 Выбор и обоснование средств разработки программного продукта 11](#_Toc11760988)

[2 СПЕЦИАЛЬНАЯ ЧАСТЬ 13](#_Toc11760989)

[2.1 Реализация структуры базы данных 13](#_Toc11760990)

[2.2 Разработка программного продукта 13](#_Toc11760991)

[2.3 Тестирование и отладка 16](#_Toc11760992)

[2.4 Руководство пользователя 19](#_Toc11760993)

[ЗАКЛЮЧЕНИЕ 21](#_Toc11760994)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 24](#_Toc11760995)

[ПРИЛОЖЕНИЯ 38](#_Toc11760995)

# **ВВЕДЕНИЕ**

Производственная практика — это часть учебного процесса, которая позволяет продемонстрировать практические навыки выполнения работы квалифицированных рабочих и специалистов.

Во время прохождения производственной практики идёт закрепление и конкретизация результатов теоретического и практического обучения в рамках программы, а также приобретения у студентов умения и навыков практической работы по присваиваемой специальности.

Производственная практика по модулю ПМ 02 «Участие в разработке информационных систем» проходила на базе «ООО Сорус». Перед прохождение производственной практики был пройден инструктаж по технике безопасности. Данная практика проходила в период с 03.09.2019 по 19.12.2019. Продолжительность составляла 252 учебных часа.

Актуальность данной практики заключается в необходимости создания программного продукта, помогающего мобильно и коммуникабельно осуществлять определённую деятельность на предприятии или в учреждении.

Цель в процессе работы заключалась в том, чтобы разработать программу и базу данных, с которой программа будет работать, а также изучение новых технологий, которые помогут работать с информационными системами в будущем.

В данном случае для создания были выбраны приложение-конструктор для тестирования, который можно использовать в учебном процессе и внеклассных мероприятиях.

В рамках поставленной цели были определены следующие задачи для решения:

* Составить технические задания;
* Выбрать и обосновать средства реализации БД;
* Выбрать и обосновать средства реализации интерфейса;
* Разработать базу данных для приложений;
* Разработать интерфейс для спроектированной базы данных;
* Отладить программные продукты и устранить ошибки при их работе;
* Разработать руководства пользователя.

Объектом автоматизации является база данных, содержащая в себе информацию о пользователе.

Предметом автоматизации является метод тестирования для учебного процесса и внеклассных мероприятий.

# **1 ОБЩАЯ ЧАСТЬ**

## **1.1 Характеристика места прохождения практики**

Компания «Сорус» имеет статус 1С: Франчайзи и предоставляет полный спектр IT-услуг по автоматизации предприятий.

Компания предоставляет следующие услуги:

* внедрение, сопровождение 1с (ИТС);
* услуги ит-аутсорсинга, внедрение информационных систем;
* сопровождение торгового оборудования;
* поставка компьютерного оборудования, оргтехники;
* лицензионное программное обеспечение;
* разработка и внедрение систем контроля доступа (СКУД).

Компания «Сорус» молодая, быстроразвивающаяся компания. В настоящее время обслуживает более 200 объектов в Новокузнецке и Кемеровской области.

Клиенты компании:

* ООО «ЕвразХолдинг»;
* СПК «Чистогорский» и сеть магазинов «Чистогорские деликатесы» ;
* Оптовая компания и сеть магазинов «Сибирский цирюльник»;
* Медицинский Центр «Абсолютное здоровье»;
* Оптовая компания «Ива»;
* АО «Завод Универсал»;
* Новокузнецкий завод резервуарных металлоконструкций имени Н.Е. Крюкова;
* ООО «Новокузнецкий завод металлообработки»;
* Сеть магазинов спортивных товаров «Твоя стихия»;
* Сеть салонов «Оптика 100%»;
* Сеть магазинов верхней одежды «Bogner»;
* МБУ «Комбинат питания»;
* Инженерная компания «Аква»;
* Авто-мото-центр «Кузнецкие моторы»;
* Оптовая компания профессиональной косметики «Эверест»;
* Торговая компания «Снабсервис»;
* Оптовая компания «СибИнвест;

и многие другие.

Компания реализует продукцию и является партнером таких компаний как:

* Фирма «1С»;
* «Microsoft»;
* «ESET»;
* «Kaspersky Lab»;
* «Атол»;
* «Штрих-М».

В офисе компании установлено 6 стационарных компьютеров DEXP Aquilon O176, нетбук HP Stream 14-ax018ur, ноутбук Acer Aspire 3 A315-21G-66F2 и МФУ для печати HP LaserJet Pro M132nw

Доступ к сети Ethernet предоставляется провайдером «РОСТЕЛЕКОМ». Используемое оборудование:

* маршрутизатор ADSL2+TP-LINK TD-W8151N
* коммутатор NETIS ST3310GF

Топология сети, используемая в компании- «Общая шина» в соответствии с рисунком 1.

Общая шина, представляет собой общий кабель (называемый шина или магистраль), к которому подсоединены все рабочие станции. На концах кабеля находятся терминаторы, для предотвращения отражения сигнала.

Достоинства:

* + - небольшое время установки сети;
    - дешевизна (требуется меньше кабеля и сетевых устройств);
    - простота настройки;
* выход из строя рабочей станции не отражается на работе сети.

Недостатки:

* неполадки в сети, такие как обрыв кабеля и выход из строя терминатора, полностью блокируют работу всей сети;
* сложная локализация неисправностей;
* с добавлением новых рабочих станций падает производительность сети.

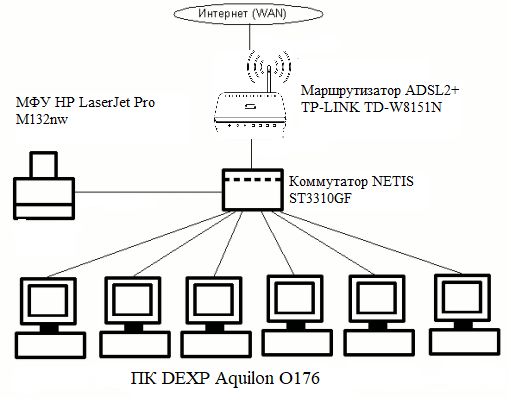


Рисунок 1 - Топология сети в офисе компании «СОРУС»

## **1.2 Техническое задание**

Разработать конструктор-тест, который осуществляет опрос игроков, ведет подсчет очков и заносит результаты в базу данных.

## **1.3 Выбор и обоснование средств реализации БД**

Основным требованием работы является наличие интерфейса приложения, разрабатываемого в СУБД Microsoft Access 2016.

На сегодняшний день СУБД Microsoft Access является одной из самых известных, удобных, простых и быстрых из всего семейства существующих СУБД.

Рассмотрим преимущества данной СУБД.

Простой и понятный интерфейс. Приступить к работе с данной СУБД можно, не имея никаких особых навыков. Кроме того, на помощь новичку приходят многочисленные справочные пособия, онлайн- и офлайн-курсы. СУБД MS Access общеизвестна и очень популярна в настольных ПК. Это значит, что вы всегда можете обратиться за помощью к знакомому, уже давно использующему данную СУБД, за советом, а также вместе работать с одной базой данных по сети. Практически безграничные возможности экспорта данных: данные из таблиц щелчком одной клавиши мыши можно перенести в Excel, Word, экспортировать в XML, опубликовать в PDF, не говоря уже о том, чтобы без проблем перенести выбранные объекты в другую базу данных. Невысокая цена. Если покупать MS Access в составе полного пакета Microsoft Office, то, по сравнению с другими платными СУБД, цена окажется очень заманчивой. Широкий выбор конструкторов для построения форм, отчетов и запросов, с помощью которых можно отфильтровать данные и вывести их в удобном виде. Широкие возможности импорта данных: если у вас есть табличные данные, созданные при помощи текстового процессора MS Word или табличного процессора MS Excel, вы с помощью мастера без труда перенесете их в свою базу. Импорт, кроме того, можно выполнить из простого текстового документа, из документа XML, а также из файлов баз данных, созданных в других СУБД (таких как dBASE, PARADOX). Возможность создания пароля на своей базе данных. Встроенный язык VBA высокого уровня. Возможность записи макросов. Редактор SQL. Для программистов можно выделить еще один значительный плюс: ядро Access JET 4 встроено в операционные системы Microsoft, начиная с Windows-98. Это значит, что разработанное приложение, использующее в работе базу данных, созданную в Access, можно без проблем переносить на любой компьютер с ОС Windows. Оно будет работать, не требуя установки самой СУБД. В крайнем случае потребуется лишь установка RUNTIME - версии, которую можно абсолютно бесплатно скачать с сайта Microsoft.

При написании программ в данной работе выбран язык C#. Данный язык программирования наиболее оптимален для создания и работы с приложениями при взаимодействии с как с Microsoft Access.

В С# разрабатывается интерфейс программы, а также задаются все необходимые функции для работы информационной системы. Базы данных создается и хранится в Microsoft Access, с её помощью работает приложение-викторина. Соединение C# с Microsoft Access, на котором находится база данных осуществляются через прямое непосредственное подключение OLE DB, с указание пути к базе прописанное в коде программы. После этого можно работать с системой.

Таким образом основными критериями выбора данной СУБД были:

* Многопоточность. Поддержка нескольких одновременных запросов.
* Оптимизация связей с присоединением многих данных за один проход.
* Записи фиксированной и переменной длины.
* Гибкая система привилегий и паролей.
* До 16 ключей в таблице. Каждый ключ может иметь до 15 полей.
* Поддержка ключевых полей и специальных полей в операторе CREATE.
* Поддержка чисел длинной от 1 до 4 байт (ints, float, double, fixed), строк переменной длины и меток времени.
* Интерфейс на языке C#.
* Основанная на потоках, быстрая система памяти.
* Псевдонимы применимы как к таблицам, так и к отдельным колонкам в таблице.
* Все поля имеют значение по умолчанию. INSERT можно использовать на любом подмножестве полей.
* Легкость управления таблицей, включая добавление и удаление ключей и полей.

Итак, популярность Microsoft Access определяется не только тем, что пакет распространяется бесплатно и имеет высокую производительность, но ещё и тем, что имеет множество вышеперечисленных достоинств, многих из которых нет у других СУБД.

Разрабатываемые приложения нуждаются в возможности обработки и хранения больших объемов информации. Microsoft Access предоставляет всем эту возможность. Это действительно одна из самых мощных, быстрых и надежных СУБД.

## **1.4 Выбор и обоснование средств разработки программного продукта**

Для реализации программных продуктов с c# были рассмотрены следующие программные средства:

* MS Access 2016;
* Visual Studio 2017;
* Язык программирования C#.

Главными критериями при выборе сервера выступали базовый набор компонентов, простота и надежность. Среди всех вариантов наиболее приемлемым стал MS Access.

MS Access входит в состав пакета Microsoft Office и является удобным для работы приложений на любом компьютере в соединении с С#.

C# – это язык программирования, на котором написано много приложений и разных информационных систем. Данные программные продукты были написаны на этом языке в приложении Windows Forms в среде Visual Studio 2017. На формах создан интерфейс, программа взаимодействует с базами данных через прямое подключение OLE DB. В Visual Studio можно редактировать код, поэтому в данной работе не использовался отдельный редактор, поскольку эти функции предусмотрены в данной программе. Visual Studio включает в себя следующие полезные компоненты:

* подсветка синтаксиса,
* проверка правильности расстановки скобок,
* контекстная помощь по коду.

Другие редакторы и программные языки были рассмотрены, но не выбраны, поскольку C# самый удобный и доступный вариант в быстром освоении и проектировании систем.

# **2 СПЕЦИАЛЬНАЯ ЧАСТЬ**

## **2.1 Реализация структуры базы данных**

База данных «Vopros-otvet.accdb», была создана в программе Microsoft Access 2016.

Для создания таблиц использовался режим конструктора, позволяющий установить свойства полей таблицы.

Физическая модель базы данных «Vopros-otvet.accdb» создана на основе трех таблиц:

* «Vopros-otvet»;
* «POLSOVATELI»;
* «IGROKI».

Структура базы данных «Vopros-otvet.accdb» представлена на рисунке 2.

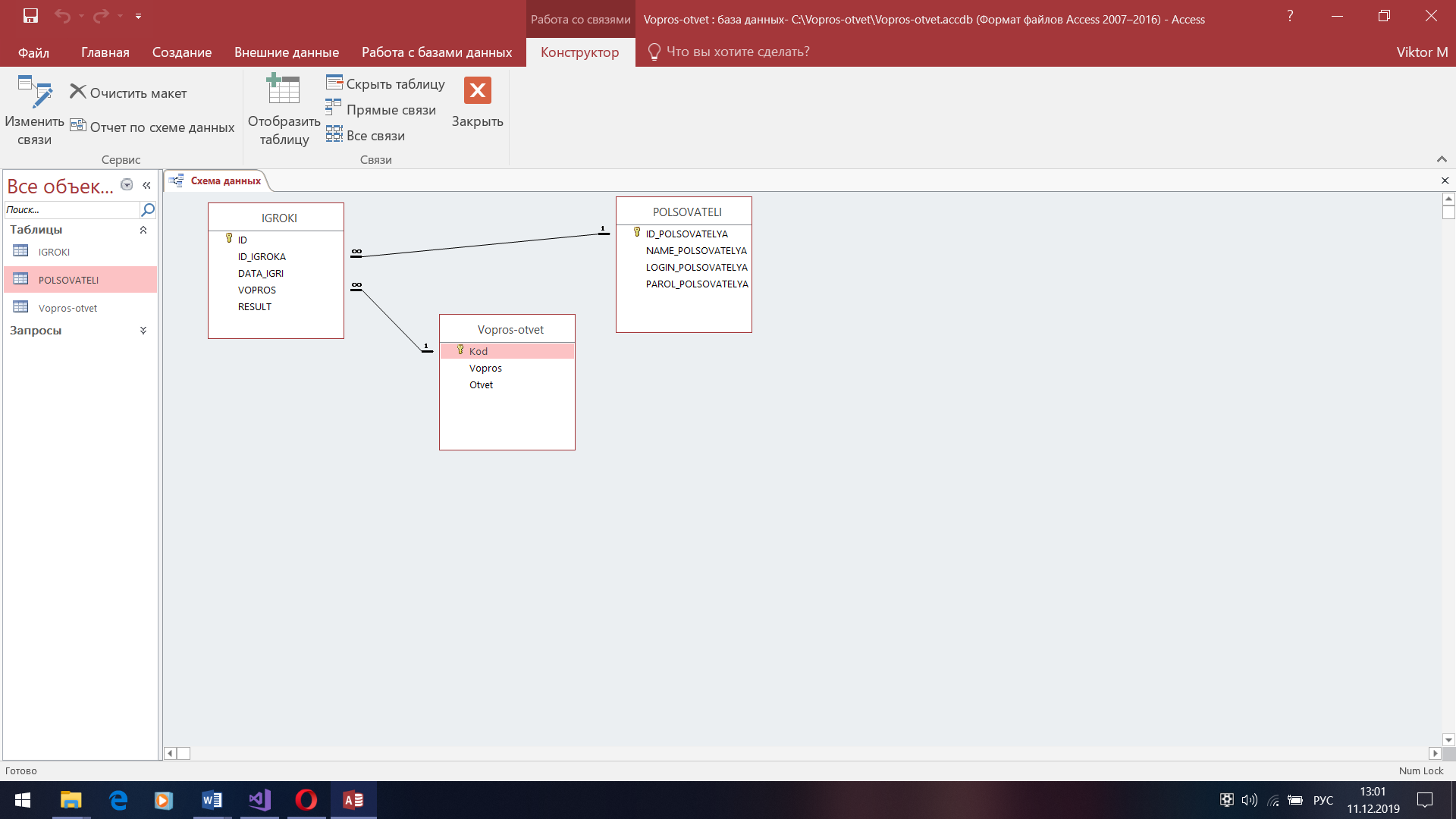


Рисунок 2 - Структура базы данных «Vopros-otvet.accdb»

## **2.2 Разработка программного продукта**

При разработке программных продуктов было создано несколько подсистем в каждом из них.

В программе «Приложение-конструктор для тестирования». были реализованы подсистемы «Авторизация» (код авторизации представлен в листинге 1), в которой хранятся данные о пользователях, в подсистеме «Регистрация» (листинг 3), в которой добавляются новые пользователи.

Листинг 1 Код подсистемы «Авторизация».

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

{

OleDbConnection con = new OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0;Data Source= C:\Vopros-otvet\Vopros-otvet.accdb");

OleDbDataAdapter dataAdapter = new OleDbDataAdapter("Select Count(\*) From Database where Login\_Polsovatelya ='" + textBox1.Text + "' and Parol\_Polsovatelya ='" + textBox2.Text + "'", con);

DataTable dt = new DataTable();

dataAdapter.Fill(dt);

if (dt.Rows[0][0].ToString() == "1")

{

this.Hide();

Form1 ss = new Form1();

ss.Show();

}

else

{

MessageBox.Show("Неправильно введённые имя или пароль");

}

}

Так же в подсистеме авторизация имеется проверка типа пользователя, которая позволяет зайти в систему администратору под логином и паролем. На листинге 2 представлен код авторизации администратора в данных приложениях.

Листинг 2 Авторизация администратора в системе «Приложение-конструктор для тестирования».

OleDbConnection con = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source= +Application.StartupPath + "/ Vopros-otvet.accdb");

OleDbDataAdapter ada = new OleDbDataAdapter("SELECT \* FROM POLSOVATELI WHERE Login\_Polsovatelya='" + textBox1.Text + "' AND Parol\_Polsovatelya='" + textBox2.Text + "'", con);

DataTable dt = new DataTable();

ada.Fill(dt);

if (dt.Rows[0][0].ToString() == "1")

{

Категории p1 = new Категории();

p1.Show();

p1.Hide();

p1.Visible = true;

this.Visible = false;

MessageBox.Show("Вход в режиме администратора");

}

else

{

MessageBox.Show("Неправильный логин или пароль!!!");

}

Листинг 3 Код подсистемы «Регистрация».

connection.Open();

OleDbCommand cmd = new OleDbCommand("INSERT INTO [Account] ([Login\_Polsovatelya], [Parol\_Polsovatelya], [Name\_Polsivatelya]) VALUES (@Login, @Password, @Name)", connection);

cmd.Parameters.AddWithValue("Login", textBox1.Text);

cmd.Parameters.AddWithValue("Password", textBox3.Text);

cmd.Parameters.AddWithValue("Name", textBox4.Text);

cmd.ExecuteNonQueryAsync();

MessageBox.Show("Данные добавлены!");

При старте программы появляется главное окно, игра начинается по нажатии кнопки «Начать игру». На форме присутствуют скрытые элементы (dataGridView1 и 2 и textBox1) необходимые для работы программы, но не представляющие интереса для пользователей программы, в них поступает информация из базы данных (число вопросов, верный ответ, и рандомный порядок вопросов)

На рисунке 3 показан старт программы, на рисунке 4 можно увидеть скрытые элементы (они выделены)



Рисунок 3 – Старт программы

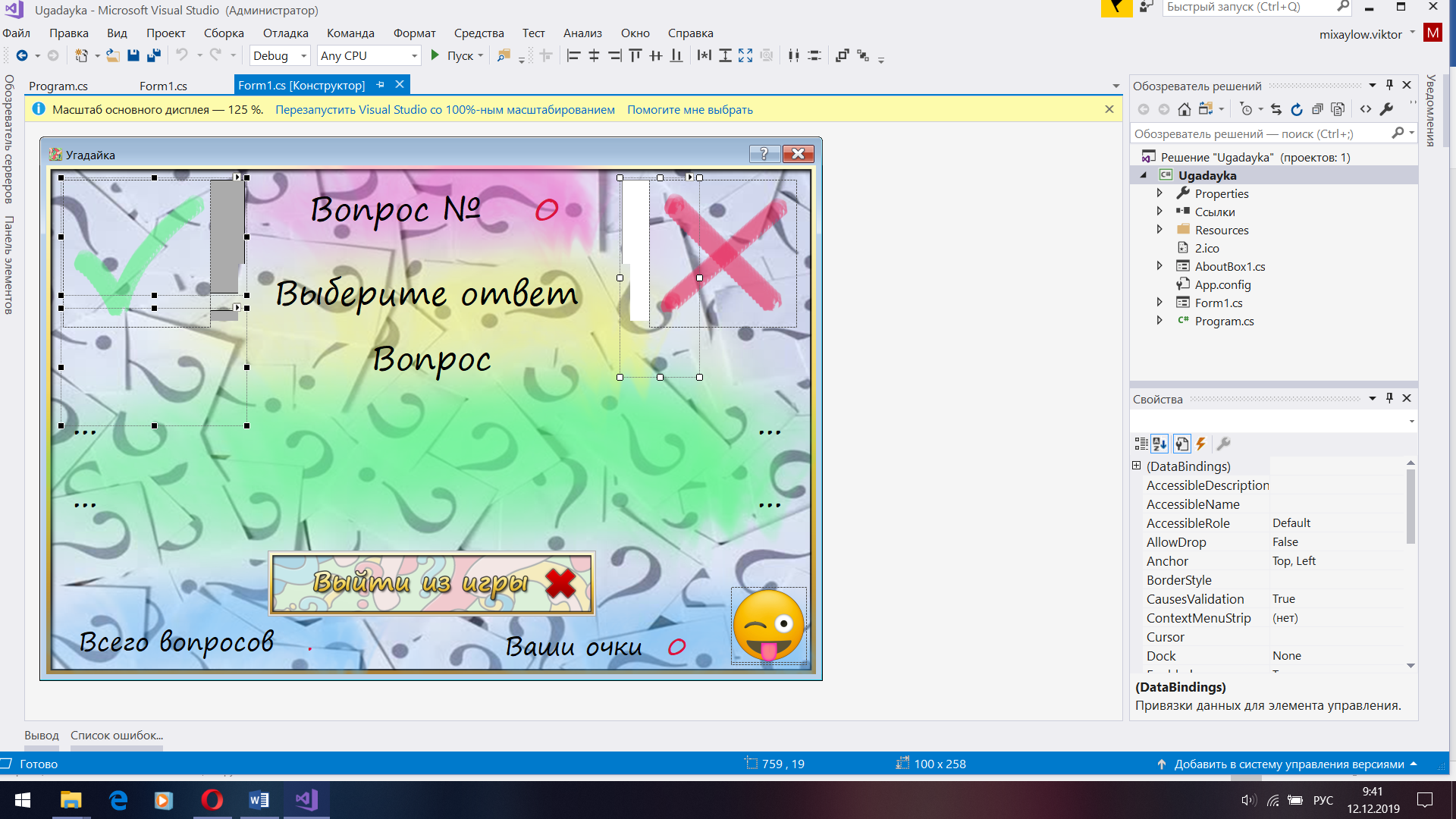


Рисунок 4 – Окно построения программы в Visual Studio 2107

Код всей программы указан в приложении В

## **2.3 Тестирование и отладка**

Тестирование – это процесс выполнения программы, целью которой является выявление ошибок в системе.

Отладка – процесс поиска, локализации и исправления ошибок в программе.

Тестирование – это процесс выполнения ПО системы или компонента в условиях анализа или записи получаемых результатов с целью проверки (оценки) некоторых свойств тестируемого объекта.

Тестирование – это процесс анализа пункта требований к ПО с целью фиксации различий между существующим состоянием ПО и требуемым (что свидетельствует о проявлении ошибки) при экспериментальной проверке соответствующего пункта требований.

Тестирование – это контролируемое выполнение программы на конечном множестве тестовых данных и анализ результатов этого выполнения для поиска ошибок.

Шаги процесса задаются тестами.

Каждый тест определяет:

* Свой набор исходных данных и условий для запуска программы.
* Набор ожидаемых результатов работы программы.

Целью проектирования тестовых вариантов является систематическое обнаружение различных классов ошибок при минимальных затратах времени и стоимости.

Тестирование обеспечивает:

* Обнаружение ошибок.
* Демонстрацию соответствия функций программы ее назначению.
* Демонстрацию реализации требований к характеристикам программы.
* Отображение надежности как индикатора качества программы.

Цель тестирования противоположна целям других этапов разработки. Его целью является нахождение ошибок. Успешным считается тест, нарушающий работу ПО. Все остальные этапы разработки направлены на предотвращение ошибок и недопущение нарушения работы программы.

Тестирование никогда не доказывает отсутствия ошибок. Отсутствие ошибок может указывать как на безупречность программы, так и на неэффективность или неполноту тестов.

Тестирование не повышает качества ПО - оно указывает на качество программы, но не влияет на него.

Виды тестирования.

Тестирование - самая популярная методика повышения качества, подкрепленная многими исследованиями и богатым опытом разработки коммерческих приложений. Существует множество видов тестирования: одни обычно выполняют сами разработчики, а другие - специализированные группы. Виды тестирования перечислены ниже:

* Блочным тестированием называют тестирование полного класса, метода или небольшого приложения, написанного одним программистом или группой, выполняемое отдельно от прочих частей системы.
* Тестирование компонента - это тестирование класса, пакета, небольшого приложения или другого элемента системы, разработанного несколькими программистами или группами, выполняемое в изоляции от остальных частей системы.
* Интеграционное тестирование - это совместное выполнение двух или более классов, пакетов, компонентов или подсистем, созданных несколькими программистами или группами.
* Регрессивным тестированием называют повторное выполнение тестов, направленное на обнаружение дефектов в программе, уже прошедшей этот набор тестов.
* Тестирование системы - это выполнение ПО в его окончательной конфигурации, интегрированного с другими программными и аппаратными системами.

Фазы тестирования.

Реализация тестирования делится на три этапа:

* Создание тестового набора (test suite) путем ручной разработки или автоматической генерации для конкретной среды тестирования (testing environment).
* Прогон программы на тестах, управляемый тестовым монитором (test monitor, test driver) с получением протокола тестирования (test log).
* Оценка результатов выполнения программы на наборе тестов с целью принятия решения о продолжении или остановке тестирования.

Тестирование проводилось в рамках колледжа студентами и преподавателями. В ходе его проведения были выявлены и в дальнейшем в процессе отладки устранены проблемы отображения представлений, вывод нужных данных на формы, а также вывод дипломов и сертификатов на печать.

## **2.4 Руководство пользователя**

Для начала необходимо рассмотреть само «Приложение-конструктор для тестирования».

На рисунке 5 показано начало игры. После нажатия кнопки «Начать игру» в окне программы появляется случайно выбранный из базы вопрос и 4 варианта ответа, один из которых верный. При выборе ответа программа выдает информацию об ответе, записывает или не записывает очки игроку (в зависимости от верности ответа) и кнопка перехода на следующий вопрос становится активной.

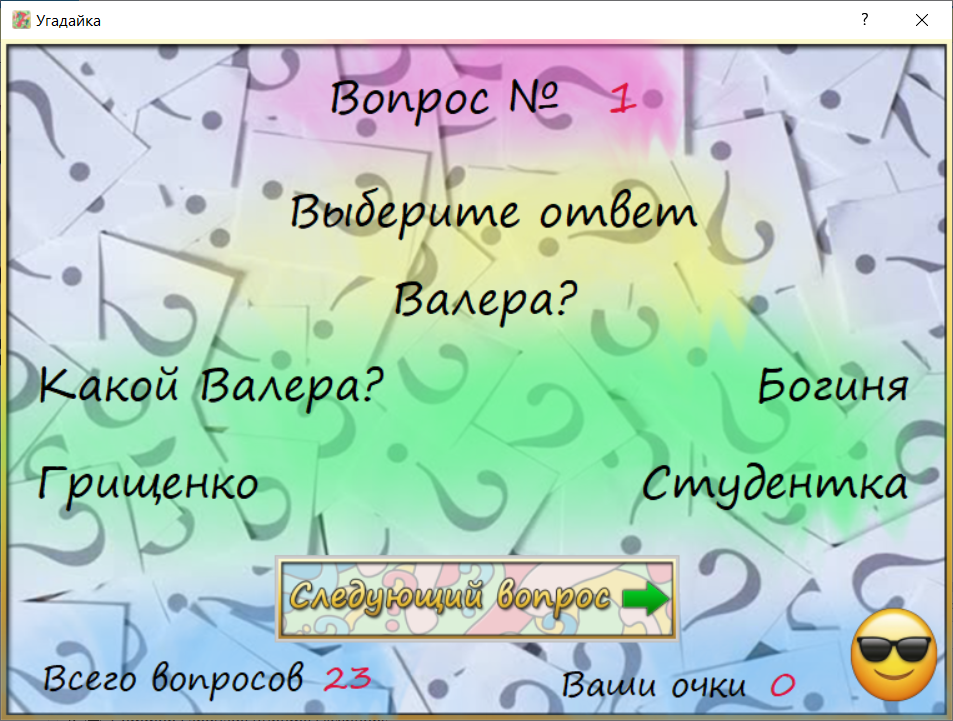


Рисунок 5 – Начало викторины

При нажатии кнопки «Следующий вопрос» программа действует идентичным образом.

В игре так же присутствуют голосовые сообщения. Они срабатывают: при старте программы, выборе ответов и завершении программы. На рисунке 6 видно, какие звуковые файлы добавлены в проект.

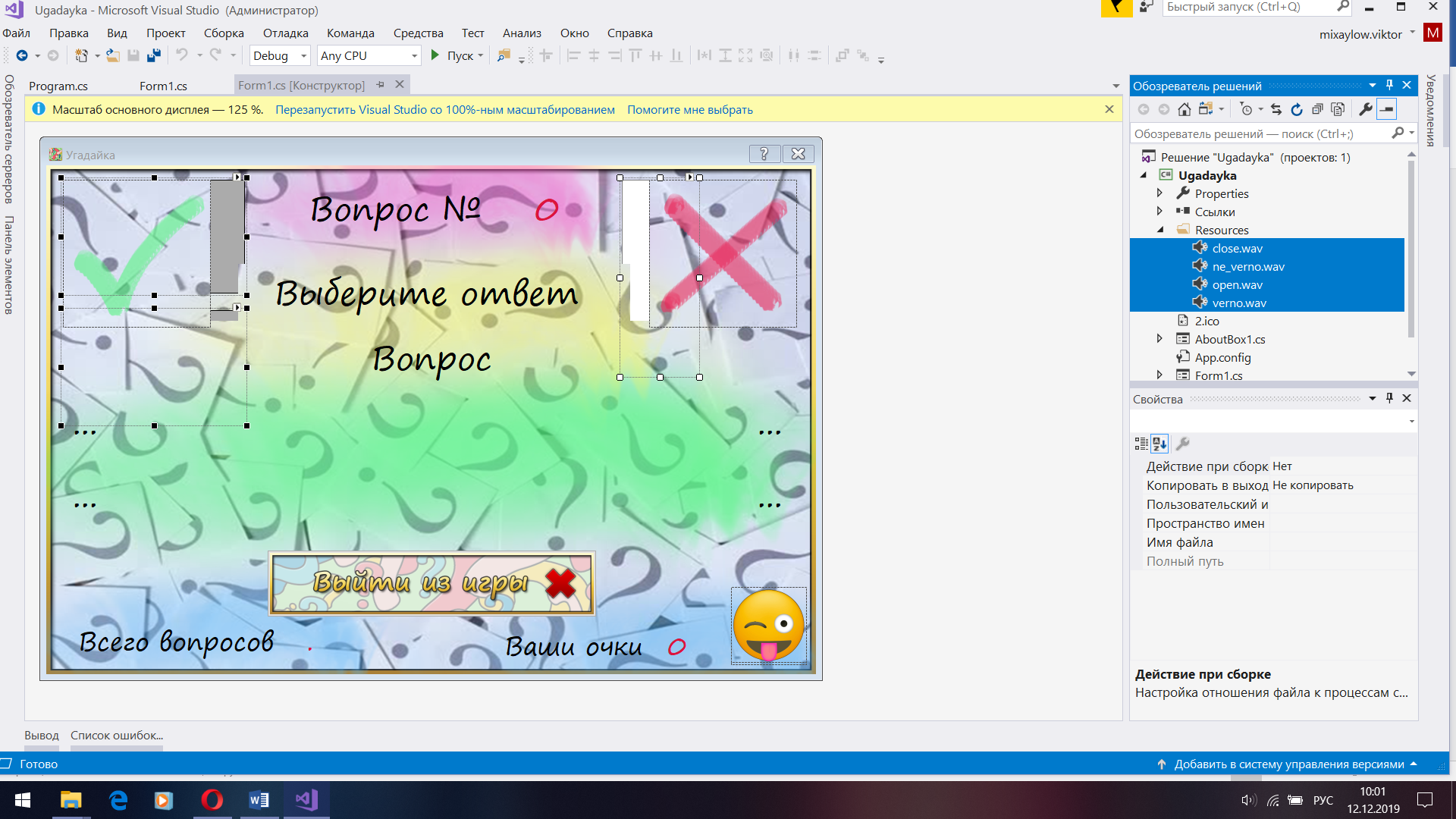


Рисунок 6 – Звуки в игре

Звук Open.wav срабатывает при старте программы. Verno.wav и Ne\_verno.wav проигрываются при выборе ответа. Close.wav включается при закрытии программы.

При выборе ответа в окне программы показываются изображения, символизирующие верность или неверность ответа. Они предоставлены на рисунке 7.

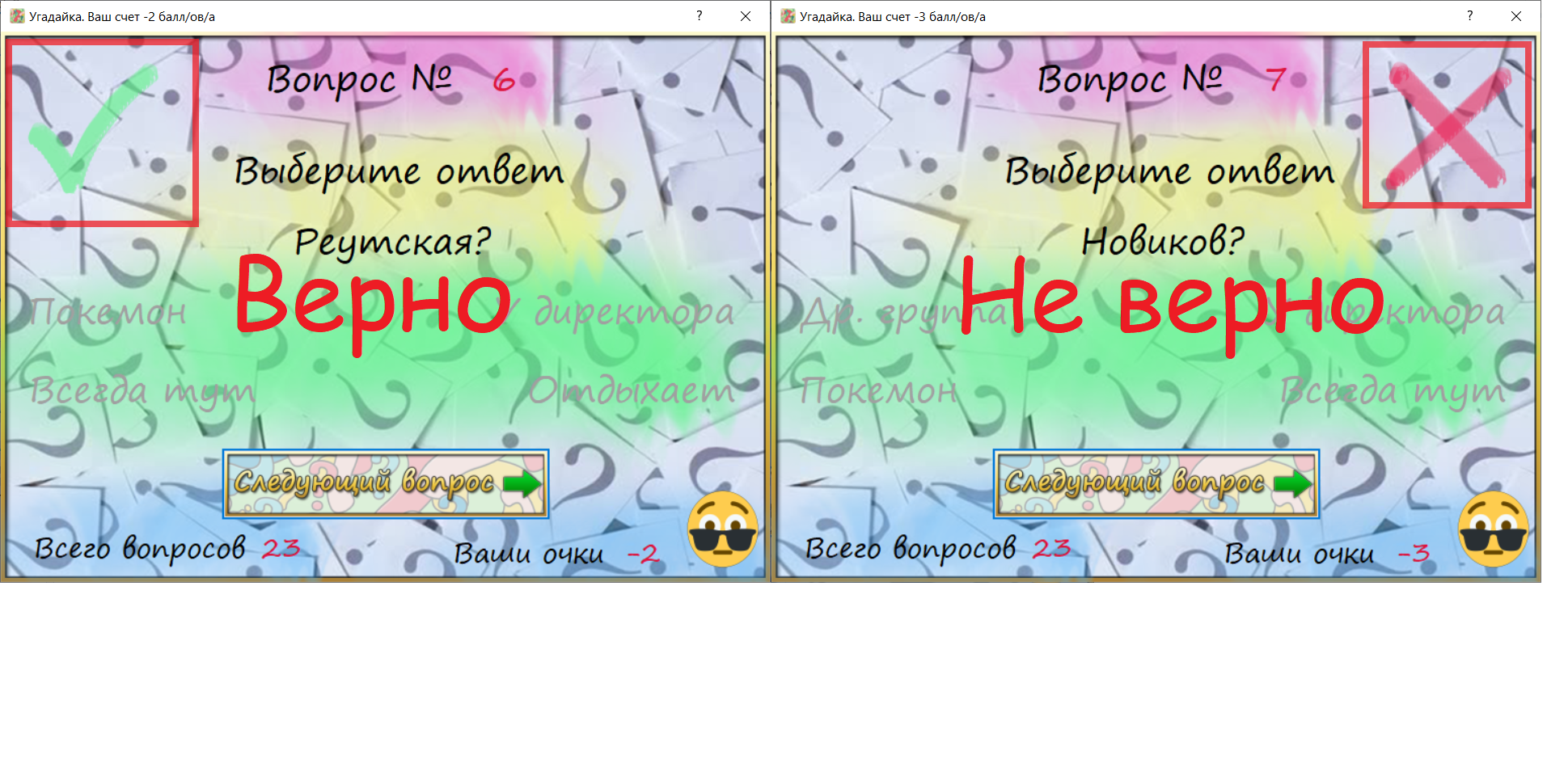


Рисунок 7 – наглядное обозначение верности ответа

Так же общий уровень набранных очков демонстрируется смайликами-подсказками, возникающими в правом нижнем углу окна программы. На рисунке 8 видны все смайлики. В игре они появляются по одному в связи с изменением счета игрока. Если счет меньше нуля появляется злой смайлик, счет меньше 5 – озадаченный, счет 0 – смайлик в очках, счет больше – смайлик подмигивающий, и если счет больше 6 – то довольный смайлик. Код изменения смайликов представлен в листинге 4

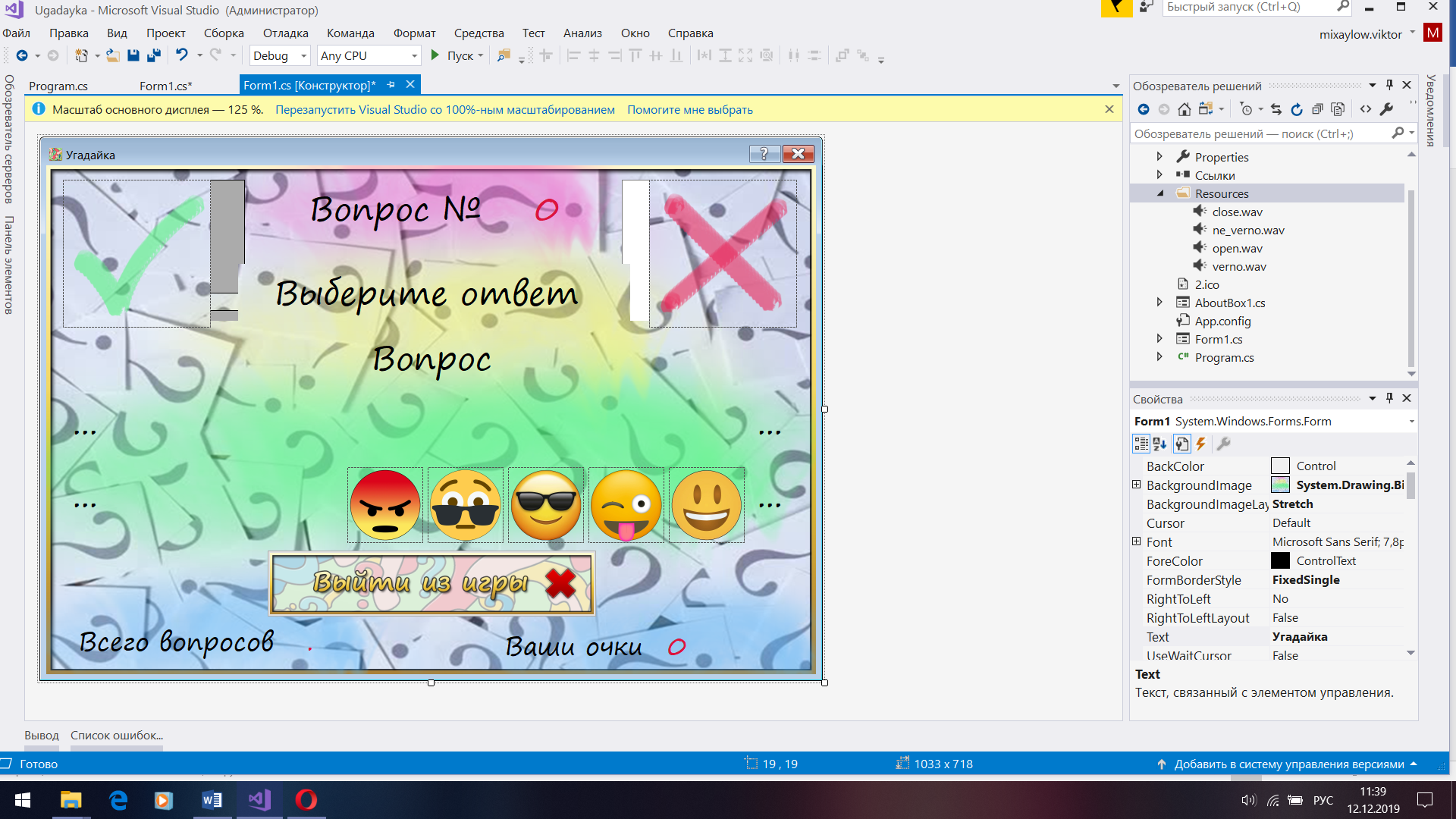


Рисунок 8 – Смайлики

Листинг 4.

if (Convert.ToInt32(label8.Text) < 0)

{

pictureBox4.Visible = false;

pictureBox5.Visible = true;

pictureBox6.Visible = false;

pictureBox7.Visible = false;

pictureBox8.Visible = false;

}

if (Convert.ToInt32(label8.Text) < (-5))

{

pictureBox4.Visible = true;

pictureBox5.Visible = false;

pictureBox6.Visible = false;

pictureBox7.Visible = false;

pictureBox8.Visible = false;

}

if (Convert.ToInt32(label8.Text) == 0)

{

pictureBox4.Visible = false;

pictureBox5.Visible = false;

pictureBox6.Visible = true;

pictureBox7.Visible = false;

pictureBox8.Visible = false;

}

if (Convert.ToInt32(label8.Text) > 0)

{

pictureBox4.Visible = false;

pictureBox5.Visible = false;

pictureBox6.Visible = false;

pictureBox7.Visible = true;

pictureBox8.Visible = false;

}

if (Convert.ToInt32(label8.Text) > 6)

{

pictureBox4.Visible = false;

pictureBox5.Visible = false;

pictureBox6.Visible = false;

pictureBox7.Visible = false;

pictureBox8.Visible = true;

}

По окончании всех вопросов появляется сообщение о том, что вопросы кончились. Его видно на рисунке 9. После этого появляется кнопка «Выйти из игры, по нажатии которой игра зарывается.

Вопросы можно редактировать в базе, открыв ее в программе Microsoft Access. Рисунок 10 показывает все имеющиеся на данный момент вопросы в базе.

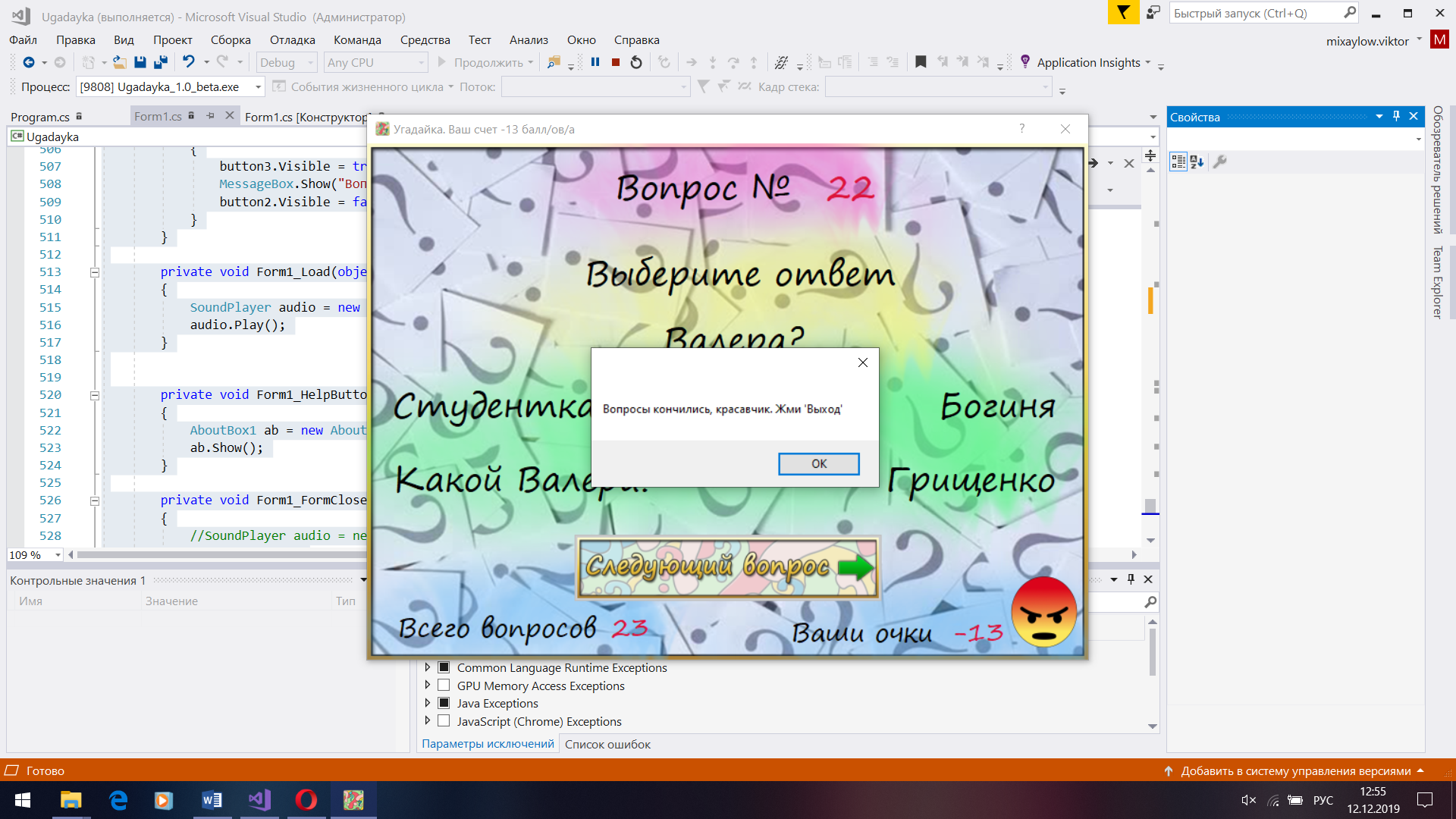


Рисунок 9 – окончание игры

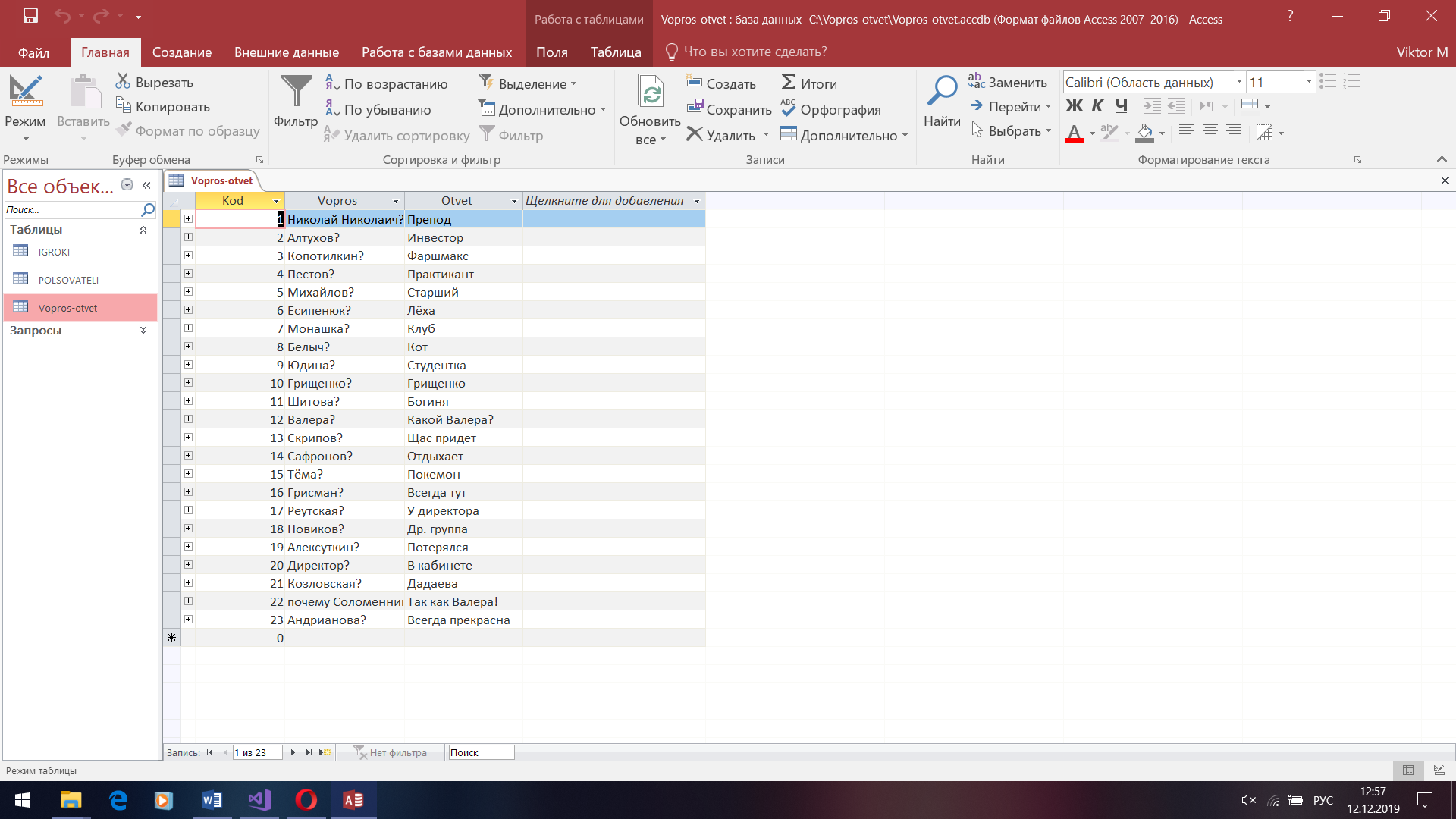


Рисунок 10 – список вопросов и ответов

# **ЗАКЛЮЧЕНИЕ**

В процессе прохождения производственной практики в компании «СОУРС» были поставлены и решены следующие задачи:

* анализ информационно-вычислительной системы предприятия;
* проведение ремонта и обслуживания оргтехники и ПК;
* установка и настройка программного обеспечения;
* реализация учебного проекта информационная система «Электронный дневник».

По окончании данной практики поставленная цель была достигнута, задачи выполнены.

Для создания информационной системы «Электронный дневник» использовалась программа «1С Предприятие», имеющая широчайший функционал, что позволило быстро и эффективно выполнить все поставленные задачи.

# **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Советов,Б.Я.. Базы данных: Учебник для СПО / Б.Я. Советов, В.В.Цехановский, В.Д. Чертовской. – 2-е изд. –М.: Издательство Юрайт, 2017 – 463 с.
2. Рыбальченко,М.В. Архитектура информационных систем: учеб. пособие для СПО / М.В. Рыбальченко. – М.: Издательство Юрайт, 2017. -91 с.
3. Стружкин, Н. П. Базы данных: проектирование. Практикум: учебное пособие для среднего профессионального образования / Н. П. Стружкин, В. В. Годин. — Москва: Издательство Юрайт, 2019. — 291 с.
4. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем: учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2019. — 432 с.
5. Астапчук, В. А. Корпоративные информационные системы: требования при проектировании: учебное пособие для вузов / В. А. Астапчук, П. В. Терещенко. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2019. — 113 с.
6. Проектирование информационных систем: учебник и практикум для академического бакалавриата / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук ; под общей редакцией Д. В. Чистова. — Москва: Издательство Юрайт, 2019. — 258 с.
7. Нестеров, С. А. Базы данных: учебник и практикум для академического бакалавриата / С. А. Нестеров. — Москва: Издательство Юрайт, 2019. — 230 с.
8. Богатырев, В. А. Информационные системы и технологии. Теория надежности: учебное пособие для бакалавриата и магистратуры / В. А. Богатырев. — Москва: Издательство Юрайт, 2019. — 318 с.
9. Казарин, О. В. Основы информационной безопасности: надежность и безопасность программного обеспечения: учебное пособие для среднего профессионального образования / О. В. Казарин, И. Б. Шубинский. — Москва: Издательство Юрайт, 2019. — 342 с.
10. James Bender Professional Test Driven Development with C#; [текст] - Москва, 2019. - 384 c.
11. Jesse Liberty Programming C#; [текст] - Москва, 2018. - 688 c.
12. Рихтер, Джефри CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C#; Питер - Москва, 2018. - 656 c.

**ПРИЛОЖЕНИЯ**

**Приложение B**

Листинг 1 весь код программы

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.Data.OleDb;

using System.Media;

namespace VOPROTV

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

button2.Enabled = false;

label3.Width = this.Width;

label3.TextAlign = ContentAlignment.MiddleCenter;

label3.AutoSize = false;

label4.Enabled = false;

label5.Enabled = false;

label6.Enabled = false;

label7.Enabled = false;

string q = "Select Count([Vopros-otvet].Kod) AS [Count-Kod] FROM[Vopros-otvet]; ";

var R = GetTble(q);

dataGridView1.DataSource = R;

foreach (DataGridViewColumn item in dataGridView1.Columns)

{

item.SortMode = DataGridViewColumnSortMode.NotSortable;

}

label1.Text = Convert.ToString(dataGridView1[0, 0].Value);

string qw = "Select \* FROM[Vopros-otvet]; ";

var Z = GetTable(qw);

dataGridView2.DataSource = Z;

}

public DataTable GetTble(string q)

{

OleDbConnection Этап1sqlConnection1 = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;

Data Source=C:\Vopros-otvet\Vopros-otvet.accdb;

Persist Security Info=False;");

OleDbDataAdapter queryAdapter = new OleDbDataAdapter(q, Этап1sqlConnection1);

DataTable R = new DataTable();

queryAdapter.Fill(R);

Этап1sqlConnection1.Close();

return R;

}

public DataTable GetTable(string qw)

{

OleDbConnection Этап1sqlConnection1 = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;

Data Source=C:\Vopros-otvet\Vopros-otvet.accdb; Persist Security Info=False;");

OleDbDataAdapter queryAdapter = new OleDbDataAdapter(qw, Этап1sqlConnection1);

DataTable R = new DataTable();

queryAdapter.Fill(R);

Этап1sqlConnection1.Close();

return R;

}

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

{

label4.Enabled = true;

label5.Enabled = true;

label6.Enabled = true;

label7.Enabled = true;

pictureBox6.Visible = true;

int st = Convert.ToInt32(label1.Text);

Random ran = new Random();

int[] mas = new int[st];

int a, n, b, m;

n = 0;

m = 0;

for (int i = 0; i <= st - 1; i++)

{

mas[i] = 0;

}

do

{

a = ran.Next(st);

if (mas[a] == 0)

{

n++;

mas[a] = n;

textBox1.AppendText(Convert.ToString(a + "\r\n"));

}

} while (n != st);

// рандомим ответы в 3й текстбокс

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

{

mas[i] = 0;

}

do

{

b = ran.Next(st);

if (mas[b] == 0)

{

m++;

mas[b] = m;

textBox3.AppendText(Convert.ToString(b + "\r\n"));

}

} while (m != 4);

int eq = Convert.ToInt32(label2.Text);

int er = Convert.ToInt32(textBox1.Lines[eq]);

label3.Text = Convert.ToString(dataGridView2[1, er].Value);

eq = eq + 1;

label2.Text = Convert.ToString(eq);

button2.Visible = true;

button1.Visible = false;

textBox2.AppendText(Convert.ToString(dataGridView2[2, er].Value + "\r\n"));//берем верный ответ

if (er > 4)

{

for (int i = 0; i < 3; i++)

{

er = er - 1;

textBox2.AppendText(Convert.ToString(dataGridView2[2, er].Value + "\r\n"));

}

}

else

{

for (int i = 0; i < 3; i++)

{

er = er + 1;

textBox2.AppendText(Convert.ToString(dataGridView2[2, er].Value + "\r\n"));

}

}

int ot1, ot2, ot3, ot4;

ot1 = Convert.ToInt32(textBox3.Lines[0]);

label4.Text = textBox2.Lines[ot1];

ot2 = Convert.ToInt32(textBox3.Lines[1]);

label5.Text = textBox2.Lines[ot2];

ot3 = Convert.ToInt32(textBox3.Lines[2]);

label6.Text = textBox2.Lines[ot3];

ot4 = Convert.ToInt32(textBox3.Lines[3]);

label7.Text = textBox2.Lines[ot4];

}

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

{

label4.Enabled = true;

label5.Enabled = true;

label6.Enabled = true;

label7.Enabled = true;

if (Convert.ToInt32(label8.Text) < 0)

{

pictureBox4.Visible = false;

pictureBox5.Visible = true;

pictureBox6.Visible = false;

pictureBox7.Visible = false;

pictureBox8.Visible = false;

}

if (Convert.ToInt32(label8.Text) < (-5))

{

pictureBox4.Visible = true;

pictureBox5.Visible = false;

pictureBox6.Visible = false;

pictureBox7.Visible = false;

pictureBox8.Visible = false;

}

if (Convert.ToInt32(label8.Text) == 0)

{

pictureBox4.Visible = false;

pictureBox5.Visible = false;

pictureBox6.Visible = true;

pictureBox7.Visible = false;

pictureBox8.Visible = false;

}

if (Convert.ToInt32(label8.Text) > 0)

{

pictureBox4.Visible = false;

pictureBox5.Visible = false;

pictureBox6.Visible = false;

pictureBox7.Visible = true;

pictureBox8.Visible = false;

}

if (Convert.ToInt32(label8.Text) > 6)

{

pictureBox4.Visible = false;

pictureBox5.Visible = false;

pictureBox6.Visible = false;

pictureBox7.Visible = false;

pictureBox8.Visible = true;

}

pictureBox2.Visible = false;

pictureBox3.Visible = false;

label4.Enabled = true;

label5.Enabled = true;

label6.Enabled = true;

label7.Enabled = true;

int st = Convert.ToInt32(label1.Text);

Random ran = new Random();

int[] mas = new int[st];

int a, n, b, m;

n = 0;

m = 0;

for (int i = 0; i <= st - 1; i++)

{

mas[i] = 0;

}

do

{

a = ran.Next(st);

if (mas[a] == 0)

{

n++;

mas[a] = n;

textBox1.AppendText(Convert.ToString(a + "\r\n"));

}

} while (n != st);

// рандомим ответы в 3й текстбокс

textBox3.Clear();

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

{

mas[i] = 0;

}

do

{

b = ran.Next(st);

if (mas[b] == 0)

{

m++;

mas[b] = m;

textBox3.AppendText(Convert.ToString(b + "\r\n"));

}

} while (m != 4);

//

int eq = Convert.ToInt32(label2.Text);

int er = Convert.ToInt32(textBox1.Lines[eq]);

label3.Text = Convert.ToString(dataGridView2[1, er].Value);

eq = eq + 1;

label2.Text = Convert.ToString(eq);

textBox2.Clear();

textBox2.AppendText(Convert.ToString(dataGridView2[2, er].Value + "\r\n"));//берем верный ответ

if (er > 4)

{

for (int i = 0; i < 3; i++)

{

er = er - 1;

textBox2.AppendText(Convert.ToString(dataGridView2[2, er].Value + "\r\n"));

}

}

else

{

for (int i = 0; i < 3; i++)

{

er = er + 1;

textBox2.AppendText(Convert.ToString(dataGridView2[2, er].Value + "\r\n"));

}

}

int ot1, ot2, ot3, ot4;

ot1 = Convert.ToInt32(textBox3.Lines[0]);

label4.Text = textBox2.Lines[ot1];

ot2 = Convert.ToInt32(textBox3.Lines[1]);

label5.Text = textBox2.Lines[ot2];

ot3 = Convert.ToInt32(textBox3.Lines[2]);

label6.Text = textBox2.Lines[ot3];

ot4 = Convert.ToInt32(textBox3.Lines[3]);

label7.Text = textBox2.Lines[ot4];

button2.Enabled = false;

if (Convert.ToInt32(label1.Text) == Convert.ToInt32(label2.Text) + 1)

{

button3.Visible = true;

MessageBox.Show("Вопросы кончились, красавчик. Жми 'Выход'");

button2.Visible = false;

}

}

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

{

SoundPlayer audio = new SoundPlayer(Ugadayka.Properties.Resources.close);

audio.Play();

System.Threading.Thread.Sleep(2000);

this.Close();

}

private void label4\_Click(object sender, EventArgs e)

{

button2.Enabled = false;

pictureBox2.Visible = false;

pictureBox3.Visible = false;

int ot5;

ot5 = Convert.ToInt32(label8.Text);

if (label4.Text == textBox2.Lines[0])

{

SoundPlayer audio = new SoundPlayer(Ugadayka.Properties.Resources.verno);

audio.Play();

label8.Text = Convert.ToString(ot5 + 1);

label4.Enabled = false;

label5.Enabled = false;

label6.Enabled = false;

label7.Enabled = false;

this.Text = "Угадайка. Ваш счет " + label8.Text + " балл/ов/а";

button2.Enabled = true;

pictureBox2.Visible = true;

}

else

{

SoundPlayer audio = new SoundPlayer(Ugadayka.Properties.Resources.ne\_verno);

audio.Play();

label8.Text = Convert.ToString(ot5 - 1);

label4.Enabled = false;

label5.Enabled = false;

label6.Enabled = false;

label7.Enabled = false;

this.Text = "Угадайка. Ваш счет " + label8.Text + " балл/ов/а";

button2.Enabled = true;

pictureBox3.Visible = true;

}

if (Convert.ToInt32(label1.Text) == Convert.ToInt32(label2.Text) + 1)

{

button3.Visible = true;

button2.Visible = false;

}

}

private void label5\_Click(object sender, EventArgs e)

{

button2.Enabled = false;

pictureBox2.Visible = false;

pictureBox3.Visible = false;

int ot5;

ot5 = Convert.ToInt32(label8.Text);

if (label5.Text == textBox2.Lines[0])

{

SoundPlayer audio = new SoundPlayer(Ugadayka.Properties.Resources.verno);

audio.Play();

label8.Text = Convert.ToString(ot5 + 1);

label4.Enabled = false;

label5.Enabled = false;

label6.Enabled = false;

label7.Enabled = false;

this.Text = "Угадайка. Ваш счет " + label8.Text + " балл/ов/а";

button2.Enabled = true;

pictureBox2.Visible = true;

}

else

{

SoundPlayer audio = new SoundPlayer(Ugadayka.Properties.Resources.ne\_verno);

audio.Play();

label8.Text = Convert.ToString(ot5 - 1);

label4.Enabled = false;

label5.Enabled = false;

label6.Enabled = false;

label7.Enabled = false;

this.Text = "Угадайка. Ваш счет " + label8.Text + " балл/ов/а";

button2.Enabled = true;

pictureBox3.Visible = true;

}

if (Convert.ToInt32(label1.Text) == Convert.ToInt32(label2.Text) + 1)

{

button3.Visible = true;

MessageBox.Show("Вопросы кончились, красавчик. Жми 'Выход'");

button2.Visible = false;

}

}

private void label6\_Click(object sender, EventArgs e)

{

button2.Enabled = false;

pictureBox2.Visible = false;

pictureBox3.Visible = false;

int ot5;

ot5 = Convert.ToInt32(label8.Text);

if (label6.Text == textBox2.Lines[0])

{

SoundPlayer audio = new SoundPlayer(Ugadayka.Properties.Resources.verno);

audio.Play();

label8.Text = Convert.ToString(ot5 + 1);

label4.Enabled = false;

label5.Enabled = false;

label6.Enabled = false;

label7.Enabled = false;

this.Text = "Угадайка. Ваш счет " + label8.Text + " балл/ов/а";

button2.Enabled = true;

pictureBox2.Visible = true;

}

else

{

SoundPlayer audio = new SoundPlayer(Ugadayka.Properties.Resources.ne\_verno);

audio.Play();

label8.Text = Convert.ToString(ot5 - 1);

label4.Enabled = false;

label5.Enabled = false;

label6.Enabled = false;

label7.Enabled = false;

this.Text = "Угадайка. Ваш счет " + label8.Text + " балл/ов/а";

button2.Enabled = true;

pictureBox3.Visible = true;

}

if (Convert.ToInt32(label1.Text) == Convert.ToInt32(label2.Text) + 1)

{

button3.Visible = true;

MessageBox.Show("Вопросы кончились, красавчик. Жми 'Выход'");

button2.Visible = false;

}

}

private void label7\_Click(object sender, EventArgs e)

{

button2.Enabled = false;

pictureBox2.Visible = false;

pictureBox3.Visible = false;

int ot5;

ot5 = Convert.ToInt32(label8.Text);

if (label7.Text == textBox2.Lines[0])

{

SoundPlayer audio = new SoundPlayer(Ugadayka.Properties.Resources.verno); // here WindowsFormsApplication1 is the namespace and Connect is the audio file name

audio.Play();

label8.Text = Convert.ToString(ot5 + 1);

label4.Enabled = false;

label5.Enabled = false;

label6.Enabled = false;

label7.Enabled = false;

this.Text = "Угадайка. Ваш счет " + label8.Text + " балл/ов/а";

button2.Enabled = true;

pictureBox2.Visible = true;

}

else

{

SoundPlayer audio = new SoundPlayer(Ugadayka.Properties.Resources.ne\_verno);

audio.Play();

label8.Text = Convert.ToString(ot5 - 1);

label4.Enabled = false;

label5.Enabled = false;

label6.Enabled = false;

label7.Enabled = false;

this.Text = "Угадайка. Ваш счет " + label8.Text + " балл/ов/а";

button2.Enabled = true;

pictureBox3.Visible = true;

}

if (Convert.ToInt32(label1.Text) == Convert.ToInt32(label2.Text) + 1)

{

button3.Visible = true;

MessageBox.Show("Вопросы кончились, красавчик. Жми 'Выход'");

button2.Visible = false;

}

}

private void Form1\_Load(object sender, EventArgs e)

{

SoundPlayer audio = new SoundPlayer(Ugadayka.Properties.Resources.open);

audio.Play();

}

private void Form1\_HelpButtonClicked(object sender, CancelEventArgs e)

{

AboutBox1 ab = new AboutBox1();

ab.Show();

}

private void Form1\_FormClosed(object sender, FormClosedEventArgs e)

{

//SoundPlayer audio = new SoundPlayer(Ugadayka.Properties.Resources.close);

//audio.Play();

}

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

//SoundPlayer audio = new SoundPlayer(Ugadayka.Properties.Resources.close);

//audio.Play();

}

}

}