СОДЕРЖАНИЕ

[Введение 4](#Введение)

[1 Экономическая характеристика объекта 5](#Эконом_харак)

[2 Постановка задачи 7](#Постанов_задач)

[2.1 Экономическое содержание задачи 7](#Содерж_задач)

[2.2 Входная информация 7](#Вход_информац)

[2.3 Выходная информация 7](#Выходн_информац)

[3 Технические и инструментальные средства для решения задачи 8](#Технические_и_инструмен)

[3.1 Обоснование выбора языка программирования 8](#Обоснование_выбора)

[3.2 Технические требования к компьютеру для решения задачи 11](#Технические_требования)

[4 Информационное обеспечение задачи 12](#Информац_обеспечение)

[4.1 Классификаторы, используемые для решения задачи 12](#Классификаторы)

[5 Программное обеспечение задачи 17](#Программное_обеспечение)

[5.1 Описание методов 17](#Описание_методов)

[5.2 Алгоритм решения задачи 19](#Алгоритм_решения)

[5.3 Описание блок – схемы 20](#Описание_блок_схемы)

[Заключение 24](#Заключение)

[Список использованных источников 25](#Список_источников)

[Приложение А 26](#Прилож_А)

[Приложение Б 27](#Прилож_Б)

# Введение

Тесты – стандартизированные измерительные методы оценки образовательных достижений (а также компетенций, способностей, черт личности и т.п.). Тесты отличаются от нестандартизированных методов тем, что все испытуемые (экзаменуемые) оказываются в строго идентичных условиях по формату и содержанию предъявляемых заданий, процедуре проведения и автоматизированной обработки результатов с количественным их отображением в виде тестовых баллов.

В настоящее время в среде педагогической общественности вузов сформировалась критическая оценка ситуации в области контроля качества вузовского образования, сложившейся в связи с тем, что традиционные методы и процедуры (прежде всего устные экзамены как главный и фактически единственный серьезный инструмент контроля) в новых условиях обнаружили свои явные недостатки, несоответствие требованиям времени:

– конъюнктура трудоустройства выпускников повысила значимость процедур контроля, способных зафиксировать уровень сформированности не только и не столько абстрактных теоретических знаний, но и практических умений (компетенций) в решение практических задач;

– неправомерное повышение процента высших оценок (оценок «отлично») создало негативные тенденции снижения мотивации учащихся в условиях уравниловки и циничного отношения к беспринципности существующей практики контроля.

# 1 Экономическая характеристика объекта

Волгодонский техникум – это образовательное учреждение, имеющее славную 20-летнюю историю, которая официально началась с 1990 года как профессиональное училище № 105. Быстро развиваясь, в 1996 году училище было преобразовано в лицей, а в 2005 году ему присвоено имя первого директора Владимира Васильевича Самарского, в 2010 году образовательное учреждение было преобразовано и получило статус техникума.

Сегодня техникум – это современное и престижное образовательное учреждение системы среднего профессионального образования Ростовской области, прочно занявшее одно из ведущих мест среди средних профессиональных образовательных учреждений региона, вошедшее в национальный реестр «Ведущие образовательные учреждения России».

Техникум, сохраняя и совершенствуя лучшие традиции отечественной системы подготовки высококвалифицированных специалистов, в процессе подготовки профессионального, компетентностного работника активно применяет инновационные и информационные технологии в обучении, основываясь на компетентностном подходе и практико-ориентированном обучении. Подготовка в техникуме ведётся по современным, востребованным на рынке труда специальностям и профессиям.

Сегодня техникум обладает развитой образовательной инфраструктурой: здание техникума удобно расположено в центральной части города, красивый и просторный трёхэтажный учебный корпус с учебными кабинетами, лабораториями и цехами, оснащенными современным учебным и производственным оборудованием, объединёнными локальной сетью, сетью Интернет и Интранет, позволяют эффективно решать учебные и воспитательные задачи. В образовательном процессе широко используются программные и инструментальные средства, с помощью которых техникум успешно реализует профессиональную подготовку выпускников и является одним из лучших в Южном Федеральном округе. В образовательном учреждении создана мощная компьютерная база, включающая 12 лабораторий с современными компьютерными и мультимедийными программами, 60 терминалов, имеющих доступ к глобальной сети Internet, в техникуме введена в действие система электронного обучения и система электронных коммуникаций. [[1]](#Один)

2 Постановка задачи

2.1 Содержание задачи

Цель курсового проекта — создание программного модуля тестирования оценки знаний студентов.

Разработанные модуль позволяет производить различные тесты, а также проверять знания студентов.

Программа создана на языке C#, графический интерфейс приложения дружественный и понятный, сама программа проста в использовании и не требует специального образования или прохождение курсов.

## 2.2 Входная информация

Входная информация в данной программе представляет собой документы текстового формата, включающие в себя разнообразные тесты с вопросами, и правильными ответами, хранящиеся в базе данных, или на компьюторе пользователя.

2.3 Выходная информация

Выходной информацией данного приложения является результат с правильными, и не правильными ответами, а так же оценка пройденного теста, над которым были произведены какие-либо из доступных действий.

# 3 Технические и инструментальные средства для решения задачи

## 3.1 Обоснование выбора языка программирования

C# (произносится как "си шарп") — современный объектно-ориентированный и типобезопасный язык программирования. C# позволяет разработчикам создавать разные типы безопасных и надежных приложений, выполняющихся в .NET. C# относится к широко известному семейству языков C, и покажется хорошо знакомым любому, кто работал с C, C++, Java или JavaScript. Здесь представлен обзор основных компонентов языка C# 8 и более ранних версий. Если вы хотите изучить язык с помощью интерактивных примеров, рекомендуем поработать с вводными руководствами по C#.

C# — объектно-ориентированный, ориентированный на компоненты язык программирования. C# предоставляет языковые конструкции для непосредственной поддержки такой концепции работы. Благодаря этому C# подходит для создания и применения программных компонентов. С момента создания язык C# обогатился функциями для поддержки новых рабочих нагрузок и современными рекомендациями по разработке ПО. В основном C# — объектно-ориентированный язык. Вы определяете типы и их поведение.

Вот лишь несколько функций языка C#, которые позволяют создавать надежные и устойчивые приложения. Сборка мусора автоматически освобождает память, занятую недостижимыми неиспользуемыми объектами. Типы, допускающие значение null, обеспечивают защиту от переменных, которые не ссылаются на выделенные объекты. Обработка исключений предоставляет структурированный и расширяемый подход к обнаружению ошибок и восстановлению после них. Лямбда-выражения поддерживают приемы функционального программирования. Синтаксис LINQ создает общий шаблон для работы с данными из любого источника. Поддержка языков для асинхронных операций предоставляет синтаксис для создания распределенных систем. В C# имеется единая система типов. Все типы C#, включая типы-примитивы, такие как int и double, наследуют от одного корневого типа object. Все типы используют общий набор операций, а значения любого типа можно хранить, передавать и обрабатывать схожим образом. Более того, C# поддерживает как определяемые пользователями ссылочные типы, так и типы значений. C# позволяет динамически выделять объекты и хранить упрощенные структуры в стеке. C# поддерживает универсальные методы и типы, обеспечивающие повышенную безопасность типов и производительность. C# предоставляет итераторы, которые позволяют разработчикам классов коллекций определять пользовательские варианты поведения для клиентского кода.

C# поддерживает управление версиями , чтобы обеспечить совместимость программ и библиотек с течением времени. Вопросы управления версиями существенно повлияли на такие аспекты разработки C#, как раздельные модификаторы virtual и override, правила разрешения перегрузки методов и поддержка явного объявления членов интерфейса. [[2]](#Два)

В качестве среды для разработки приложения была выбрана программа Microsoft Visual Studio, линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight. Существует множество сред разработки, но данная была выбрана в ввиду ее удобного графического интерфейса и средств отладки. Microsoft Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Microsoft Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

Главные возможности среды разработки Microsoft Visual Studio:

* включены все «интеллектуальные» возможности по редактированию кода;
* есть возможность визуального просмотра будущего приложения;
* сборка проекта работает быстро;
* удобный конструктор интерфейсов;
* удобное и интуитивно понятное логирование проекта;
* указанные цвета и рисунки, использованные в layout'e отображаются на границе в виде небольших превью, которые легко помогают понять какой конкретно ресурс используется;
* среда разработки является технологиями компании Microsoft;
* при выборе ресурса, его содержимое отображается во всплывающих окнах;
* возможность создания одним кликом новых окон и страниц;
* мониторинг используемой памяти;
* возможность подключать сторонние дополнения(плагины);
* возможность добавления новых инструментов. [[3]](#Три)

## 3.2 Технические требования к компьютеру для решения задачи

Рекомендуемые системные требования к компьютеру на которой будет установлена Microsoft Visual Studio 2022 следующие:

* 64-разрядный процессор 1,8 ГГц или более мощный. Рекомендуется четырехъядерный или с большим количеством ядер. Процессоры ARM не поддерживаются;
* не менее 4 ГБ ОЗУ. На используемые ресурсы влияет множество факторов; для стандартных профессиональных решений рекомендуется 16 ГБ ОЗУ;
* Windows 365: не менее 2 виртуальных ЦП и 8 ГБ ОЗУ. Рекомендуется 4 виртуальных ЦП и 16 ГБ ОЗУ;
* место на жестком диске: от 850 ГБ до 210 ГБ свободного места в зависимости от установленных компонентов, обычно для установки требуется от 20 до 50 ГБ свободного места. Для повышения производительности рекомендуется устанавливать Windows и Visual Studio на твердотельном накопителе (SSD);
* видеоадаптер с минимальным разрешением WXGA (1366 на 768 пикселей); для оптимальной работы Visual Studio рекомендуется разрешение 1920 на 1080 пикселей или выше;
* минимальное разрешение предполагает масштабирование, параметры DPI и масштабирование текста на уровне 100 %. Если не задано значение 100 %, необходимо соответствующим образом масштабировать минимальное разрешение. Например, если задать для дисплея Windows значение параметра "Масштаб и макет" на устройстве Surface Book, которое имеет физический дисплей 3000 x 2000, значение 200 %, то для Visual Studio логическое разрешение экрана будет 1500 x 1000, что соответствует минимальным требованиям в 1366 x 768. [[4]](#Четыре)

# 4 Информационное обеспечение задачи

## 4.1 Классификаторы, используемые для решения задачи

Классификатор представляет собой набор таблиц (справочников).

Объект – некоторая сущность в цифровом пространстве, обладающая определённым состоянием и поведением в данной предметной области.

Атрибут – элемент, описывающий любую характеристику объекта, важную для данной предметной области.

Объектом в данной базе данных будет: информация о преподователях приведенная в таблице 4.1, авторизации, показанных в таблице 4.2, ответов показана в таблице 4.3, информация о теме показана в таблице 4.4, правельные ответы в таблице 4.5, вопросы в таблице 4.6, теста в таблице 4.7 и тестирование в таблице 4.8.

Таблица 4.1 Атрибуты преподователя

|  |  |
| --- | --- |
| Объект | Атрибуты |
| ID\_Teacher | Name  Surname  LastName  Login  Password  ID\_Subject (FK) |

Таблица 4.2 Атрибуты авторизации

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Avtoris | login  password |

Таблица 4.3 Атрибуты ответов

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Answer | ID\_Answer  Text  ID\_Question (FK) |

Таблица 4.4 Атрибуты темы

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Subject | ID\_Subject  Name |

Таблица 4.5 Атрибуты правельных ответов

|  |  |
| --- | --- |
| Объект | Атрибуты |
| CorrectAnswers | ID\_CorrectAnswers  ID\_Answer (FK)  ID\_Question (FK) |

Таблица 4.6 Атрибуты вопросов

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Question | ID\_Question  Text  ID\_Test (FK) |

Таблица 4.7 Атрибуты теста

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Test | ID\_Test  ID\_Subject (FK)  Name |

Таблица 4.8 Атрибуты тестирования

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Testing | ID\_Testing  CountCorrectAnswer  Date  Mark  ID\_Student (FK)  ID\_Test (FK) |

Для выполнения проекта была выбрана СУБД SQL Server. Схема базы данных представлена на рисунке 4.1.

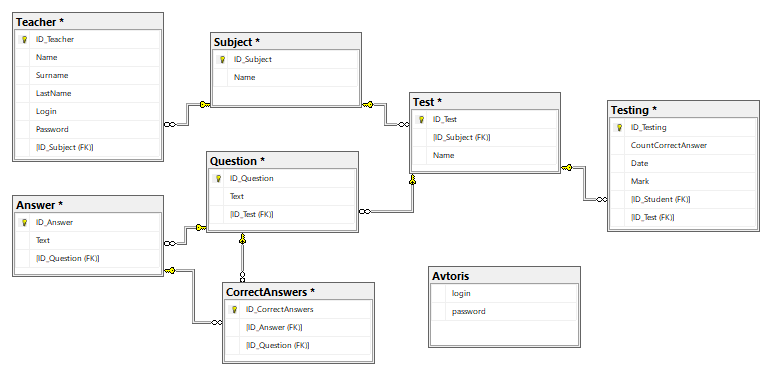


Рисунок 4.1 Схема базы данных

Структура таблиц «преподователя», «авторизации», «ответов», «темы», «правельных ответов», «вопросов», «теста» и «тестирования» базы данных представлена на рисунках 4.2-4.9 соответственно.

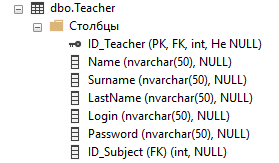


Рисунок 4.2 Таблица «преподователя»

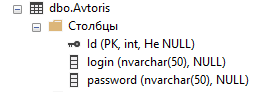


Рисунок 4.3 Таблица «авторизации»

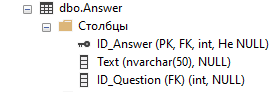


Рисунок 4.4 Таблица «ответов»

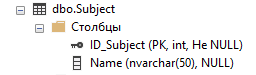


Рисунок 4.5 Таблица «тема»

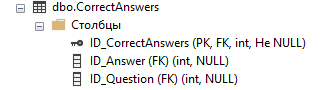


Рисунок 4.6 Таблица «правельных ответов»

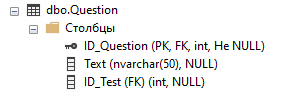


Рисунок 4.7 Таблица «вопросов»

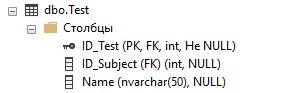


Рисунок 4.8 Таблица «теста»

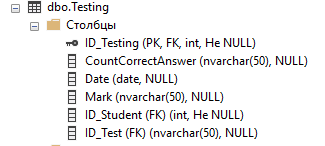


Рисунок 4.9 Таблица «тестирования»

# 5 Программное обеспечение задачи

## 5.1 Описание методов

Для примера взята часть кода для входа в программный модуль под своей учетной записью. Перед нажатием на кнопку вызывается окно авторизации, для дальнейшей работы с приложением.

При нажатии на кнопки, вызываются специализированные команды, реализованные через встроенный язык программирования C#.

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace WpfApp5

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

Open(Auto);

}

public void Open(Border screen)

{

Auto.Visibility = Visibility.Hidden;

Login.Visibility = Visibility.Hidden;

Vopros.Visibility = Visibility.Hidden;

screen.Visibility = Visibility.Visible;

}

public void Click\_reg(object sender, RoutedEventArgs e)

{

if (textBox\_login.Text.Length > 0)

{

if (password.Password.Length > 0)

{

if (password\_Copy.Password.Length > 0)

{

}

else MessageBox.Show("Повторите пароль");

}

else MessageBox.Show("Укажите пароль");

}

else MessageBox.Show("Укажите логин");

if (password.Password == password\_Copy.Password)

{

MessageBox.Show("Пользователь зарегистрирован");

SqlConnection sqlConnection = new SqlConnection(@"server=PC-232-03\SQLEXPRESS; Initial Catalog=OsaKurs; User ID=U-19K; Password=19K$YcYO");

sqlConnection.Open();

SqlCommand command = new SqlCommand("INSERT INTO Avtoris values (@login,@password)", sqlConnection);

command.Parameters.AddWithValue("@password", password.Password);

command.Parameters.AddWithValue("@login", textBox\_login.Text);

command.ExecuteNonQuery();

sqlConnection.Close();

}

else MessageBox.Show("Пароли не совподают");

Open(Login);

}

public void Click\_back(object sender, RoutedEventArgs e)

{

Open(Login);

}

public void Vopros2\_Click(object sender, RoutedEventArgs e)

{

}

public DataTable Select(string selectSQL)

{

DataTable dataTable = new DataTable("dataBase");

SqlConnection sqlConnection = new SqlConnection(@"server=PC-232-03\SQLEXPRESS; Initial Catalog=OsaKurs; User ID=U-19K; Password=19K$YcYO");

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSQL;

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand);

sqlDataAdapter.Fill(dataTable);

sqlConnection.Close();

return dataTable;

}

public void enter\_Click(object sender, RoutedEventArgs e)

{

if (logn.Text.Length > 0)

{

if (password1.Password.Length > 0)

{

DataTable dt\_user = Select(@"SELECT \* FROM [dbo].[Avtoris] WHERE [login] = '" + logn.Text + "' AND [password] = '" + password1.Password + "'");

if (dt\_user.Rows.Count > 0)

{

MessageBox.Show("Пользователь авторизовался");

Open(Vopros);

}

else MessageBox.Show("Пользователя не найден");

}

else MessageBox.Show("Введите пароль");

}

else MessageBox.Show("Введите логин");

}

public void regin\_Click(object sender, RoutedEventArgs e)

{

Open(Auto);

}

}

}

## 5.2 Алгоритм решения задачи

В данном коде имеется таблица взята с SQL Server Авторизация в ней имеются: логин и пароль которые связаны с программным модулем, данные которые мы вписываем в регестрации автоматически переносятся в базу данных и там сохраняются.

## 5.3 Описание блок – схемы

Последовательное описание блок – схемы приложения приведено ниже:

Шаг 1. Запуск программы.

Шаг 2. Вопрос: Указали ли вы логин? Если да, то переходим к следующему шагу, ввод пароля. Если пороли не совподают, то выведится ошибка: Пароли не совподают. Если же они совподают переходим к 3 Шагу.

Шаг 3. Вас перебрасывает на окно логина где нужно вести данные которые вы вписывали ранее. Если же они не будут верны тест не начнется. Если все верно переходим к 4 Шагу. Если вы забыли данние возвращаемся назад через кнопку Регистрация и вводим новые данные.

Шаг 4. Тест начался, вы должны прочитать вопрос и ответить на него из трех представленных ответов ниже. Когда ответили вас перебрасывает к следующему вопросу.

Шаг 5. Отвечаем на остальные вопросы как сказано в 4 Шаге.

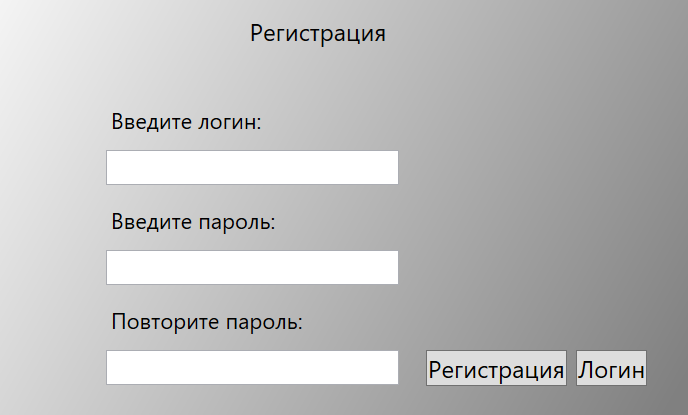
Шаг 6. Когда ответили на оставшиеся вопросы появляется окно Итоги, где показано: имя пользователя, оценка и количество правильных ответов.

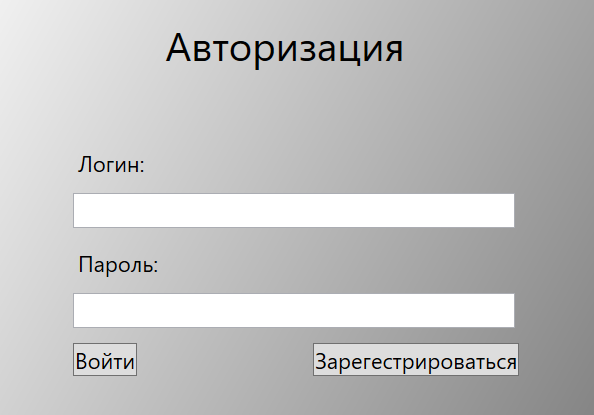
Шаг 7. Закрытие приложения.

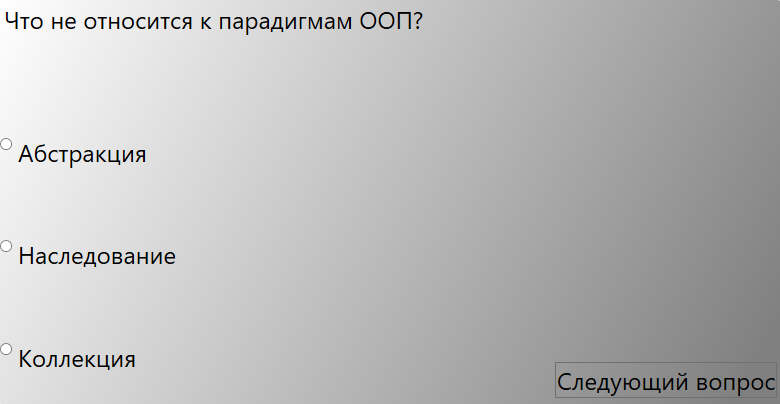
Листинг программы приведен в приложении Б.

5.4 Описание интерфейса программы

В окне регестрации в приложение, представленном на рисунке 5.1, находятся одно текстовое поле и два текстовых пароля, кнопка регестрации и логина. Для дальнейшей работы нужно ввести данные логин и пароль. Если догин не введен, то появится ошибка введите логин, так же с паролем. Когда вы за регестрировались нужно перейти к логину. Там находятся одно текстовое поле и один текстовый пароль, кнопка входа и кнопка зарегестрироваться окно представлено на рисунке 5.2. Если вы не ввели или забыли пароль появится ошибка, перейдите назад и зарегестрируйтесь заново, если все верно, то вы начнете тест. Вопрос представлен на рисунке 5.3.

Рисунок 5.1 Окно ригестрации.

Рисунок 5.2 Окно Авторизации

 Рисунок 5.3 - Окно вопроса

Заключение

Разработанный программный модуль предстовляет собой тестирование оценки студентов на основе программирования.

Программа разработана на языке C# в интегрированной среде разработки Visual Studio с использованием СУБД SQL SERVER. Тест интуитивно понятно даже неопытному пользователю и не затрудняет его работу с тестированием.

В процессе написания кода для данной программы были приобретены практические навыки в работе СУБД SQL SERVER и языке программирования C#.

Цель курсового проекта достигнута. Задачи выполнены.

# Список использованных источников

Интернет ресурсы:

1. ВТИТБиД [Электронный ресурс] Режим доступа:

<http://vtitbid.ru/bespolezn/history>

1. Краткий обзор языка C# [Электронный ресурс] Режим доступа:

<https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/>

3. Документация по Visual Studio [Электронный ресурс] Режим доступа:

<https://docs.microsoft.com/ru-RU/visualstudio/windows/?view=vs-2022&preserve-view=true>

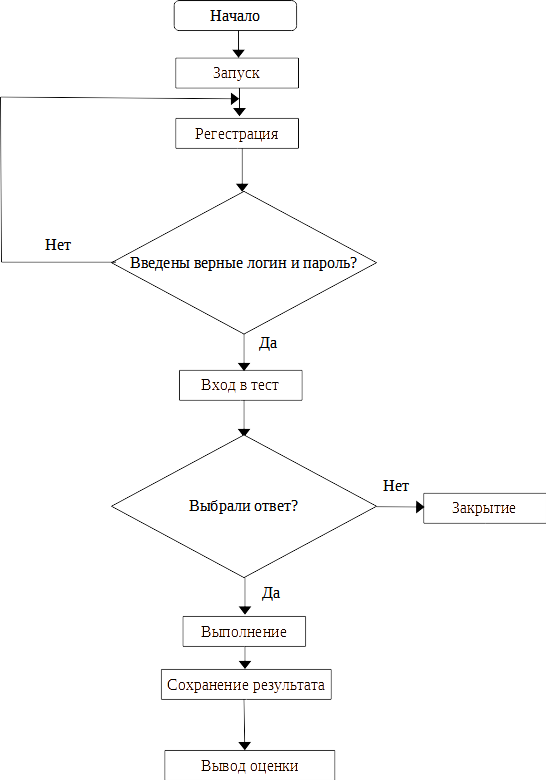
1. Требования к системе для семейства продуктов Visual Studio 2022 [Электронный ресурс] Режим доступа:

<https://docs.microsoft.com/ru-ru/visualstudio/releases/2022/system-requirements>

Приложение А

(обязательное)

Блок - схема программы



Приложение Б

(обязательное)

Листинг программы

<Window x:Class="WpfApp5.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:WpfApp5"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Grid>

<Border Name="Auto">

<Grid>

<Border CornerRadius="10" Grid.RowSpan="4">

<Border.Background>

<LinearGradientBrush>

<GradientStop Color="White" Offset="0.0"/>

<GradientStop Color="Gray" Offset="0.95"/>

</LinearGradientBrush>

</Border.Background>

</Border>

<Label Content="Регистрация" FontSize="24" HorizontalAlignment="Center" Margin="0,30,0,0" VerticalAlignment="Top"/>

<Label Content="Введите логин:" FontSize="22" HorizontalAlignment="Left" Margin="180,120,0,0" VerticalAlignment="Top"/>

<Label Content="Введите пароль:" FontSize="22" HorizontalAlignment="Left" Margin="180,220,0,0" VerticalAlignment="Top"/>

<Label Content="Повторите пароль:" FontSize="22" HorizontalAlignment="Left" Margin="180,320,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="textBox\_login" HorizontalAlignment="Left" FontSize="16" Margin="180,170,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="293" Height="35"/>

<PasswordBox x:Name="password" HorizontalAlignment="Left" FontSize="16" Margin="180,270,0,0" VerticalAlignment="Top" Width="293" Height="35"/>

<PasswordBox x:Name="password\_Copy" HorizontalAlignment="Left" FontSize="16" Margin="180,370,0,0" VerticalAlignment="Top" Width="293" Height="35"/>

<Button x:Name="reg" Click="Click\_reg" Content="Регистрация" FontSize="24" HorizontalAlignment="Left" Margin="500,370,0,0" VerticalAlignment="Top"/>

<Button x:Name="back" Click="Click\_back" Content="Логин" FontSize="24" HorizontalAlignment="Left" Margin="650,370,0,0" VerticalAlignment="Top"/>

</Grid>

</Border>

<Border Name="Login">

<Grid>

<Border CornerRadius="10" Grid.RowSpan="4">

<Border.Background>

<LinearGradientBrush>

<GradientStop Color="White" Offset="0.0"/>

<GradientStop Color="Gray" Offset="0.95"/>

</LinearGradientBrush>

</Border.Background>

</Border>

<Frame x:Name="frame" NavigationUIVisibility="Hidden" FontFamily="Arial Black"/>

<Label Content="Авторизация" FontSize="40" HorizontalAlignment="Center" Margin="0,30,0,0" VerticalAlignment="Top" />

<Label Content="Логин:" FontSize="22" HorizontalAlignment="Left" Margin="180,160,0,0" VerticalAlignment="Top" />

<Label Content="Пароль:" FontSize="22" HorizontalAlignment="Left" Margin="180,260,0,0" VerticalAlignment="Top" />

<TextBox x:Name="logn" HorizontalAlignment="Left" FontSize="26" Margin="180,210,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="442" Height="35" />

<Button x:Name="enter" Content="Войти" FontSize="22" HorizontalAlignment="Left" Margin="180,360,0,0" VerticalAlignment="Top" Click="enter\_Click" />

<Button x:Name="regin" Content="Зарегестрироваться" FontSize="22" HorizontalAlignment="Left" Margin="420,360,0,0" VerticalAlignment="Top" Click="regin\_Click" />

<PasswordBox x:Name="password1" HorizontalAlignment="Left" FontSize="26" Margin="180,310,0,0" VerticalAlignment="Top" Width="442" Height="35" />

</Grid>

</Border>

<Border Name="Vopros">

<Grid>

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Border CornerRadius="10" Grid.RowSpan="4">

<Border.Background>

<LinearGradientBrush>

<GradientStop Color="White" Offset="0.0"/>

<GradientStop Color="Gray" Offset="0.95"/>

</LinearGradientBrush>

</Border.Background>

</Border>

<Label Content="Что не относится к парадигмам ООП?" FontSize="24"/>

<RadioButton Content="Абстракция" Margin="0,0,110,0" Grid.Row="1" FontSize="24" VerticalAlignment="Center" Visibility="Visible" HorizontalAlignment="Left"/>

<RadioButton Content="Наследование" Margin="0,0,78,0" Grid.Row="2" FontSize="24" VerticalAlignment="Center" HorizontalAlignment="Left"/>

<RadioButton Content="Коллекция" Margin="0,0,98,0" Grid.Row="3" FontSize="24" VerticalAlignment="Center" HorizontalAlignment="Left"/>

<Button Click="Vopros2\_Click" Background="Transparent" Content="Следующий вопрос" HorizontalAlignment="Left" Margin="556,56,0,0" FontSize="24" Grid.Row="3" VerticalAlignment="Top" />

</Grid>

</Border>

</Grid>

</Window>

namespace WpfApp5

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

Open(Auto);

}

public void Open(Border screen)

{

Auto.Visibility = Visibility.Hidden;

Login.Visibility = Visibility.Hidden;

Vopros.Visibility = Visibility.Hidden;

screen.Visibility = Visibility.Visible;

}

public void Click\_reg(object sender, RoutedEventArgs e)

{

if (textBox\_login.Text.Length > 0)

{

if (password.Password.Length > 0)

{

if (password\_Copy.Password.Length > 0)

{

}

else MessageBox.Show("Повторите пароль");

}

else MessageBox.Show("Укажите пароль");

}

else MessageBox.Show("Укажите логин");

if (password.Password == password\_Copy.Password)

{

MessageBox.Show("Пользователь зарегистрирован");

SqlConnection sqlConnection = new SqlConnection(@"server=PC-232-03\SQLEXPRESS; Initial Catalog=OsaKurs; User ID=U-19K; Password=19K$YcYO");

sqlConnection.Open();

SqlCommand command = new SqlCommand("INSERT INTO Avtoris values (@login,@password)", sqlConnection);

command.Parameters.AddWithValue("@password", password.Password);

command.Parameters.AddWithValue("@login", textBox\_login.Text);

command.ExecuteNonQuery();

sqlConnection.Close();

}

else MessageBox.Show("Пароли не совподают");

Open(Login);

}

public void Click\_back(object sender, RoutedEventArgs e)

{

Open(Login);

}

public void Vopros2\_Click(object sender, RoutedEventArgs e)

{

}

ublic DataTable Select(string selectSQL)

{

DataTable dataTable = new DataTable("dataBase");

SqlConnection sqlConnection = new SqlConnection(@"server=PC-232-03\SQLEXPRESS; Initial Catalog=OsaKurs; User ID=U-19K; Password=19K$YcYO");

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSQL;

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand);

sqlDataAdapter.Fill(dataTable);

sqlConnection.Close();

return dataTable;

}

public void enter\_Click(object sender, RoutedEventArgs e)

{

if (logn.Text.Length > 0)

{

if (password1.Password.Length > 0)

{

DataTable dt\_user = Select(@"SELECT \* FROM [dbo].[Avtoris] WHERE [login] = '" + logn.Text + "' AND [password] = '" + password1.Password + "'");

if (dt\_user.Rows.Count > 0)

{

MessageBox.Show("Пользователь авторизовался");

Open(Vopros);

}

else MessageBox.Show("Пользователя не найден");

}

else MessageBox.Show("Введите пароль");

}

else MessageBox.Show("Введите логин");

}

public void regin\_Click(object sender, RoutedEventArgs e)

{

Open(Auto);

}

}

}