**Министерство образования И МОЛОДЕЖНОЙ ПОЛИТИКИ**

**Свердловской области**

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

СВЕРДЛОВСКОЙ ОБЛАСТИ

«ВЕРХНЕСАЛДИНСКИЙ АВИАМЕТАЛЛУРГИЧЕСКИЙ КОЛЛЕДЖ им.А.А.ЕВСТИГНЕЕВА»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**КУРСОВОЙ ПРОЕКТ**

***На тему****:*  **Проектирование и разработка программного обеспечения**

**автоматизированной информационной системы "Абитуриент”**

**(ПиДИС)**

***Специальность 09.02.07***

Информационные технологии и программирование

Лунев Леонид Михайлович

**Студент**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись Ф.И.О.

ИСП-301

**Курс \_\_\_\_**\_III\_\_\_**\_\_\_\_\_\_\_ Группа \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Руководитель:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бардина Т.И.

подпись Ф.И.О.

г.Верхняя Салда

2021г.

Содержание

[ВВЕДЕНИЕ 3](#_Toc68793058)

[1 РАЗРАБОТКА ЭСКИЗНОГО И ТЕХНИЧЕСКОГО ПРОЕКТА ПРОГРАММНОГО МОДУЛЯ 4](#_Toc68793059)

[1.1 Назначение и область применения 4](#_Toc68793060)

[1.2 Технические характеристики 4](#_Toc68793061)

[1.2.1 Постановка задачи 4](#_Toc68793062)

[1.2.2 Требования к программе 4](#_Toc68793063)

[1.2.3 Описание алгоритма 4](#_Toc68793064)

[2 РАЗРАБОТКА ИНТЕРФЕЙСА 6](#_Toc68793065)

[3 РАЗРАБОТКА РАБОЧЕГО ПРОЕКТА 6](#_Toc68793066)

[3.1 Разработка баз данных 7](#_Toc68793067)

[3.2 Разработка модулей программы 8](#_Toc68793068)

[3.3 Описание программного модуля 8](#_Toc68793069)

[3.4 Тестирование программного модуля 8](#_Toc68793070)

[4 РАЗРАБОТКА ДОКУМЕНТАЦИИ 9](#_Toc68793071)

[4.1 Руководство пользователя 9](#_Toc68793072)

[4.2 Руководство программиста 9](#_Toc68793073)

[ЗАКЛЮЧЕНИЕ 10](#_Toc68793074)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 11](#_Toc68793075)

[ПРИЛОЖЕНИЕ А 12](#_Toc68793076)

# ВВЕДЕНИЕ

Цифровая трансформация образования (ЦТО) есть процесс изменения содержания, методов и организационных форм учебной работы, который разворачивается в быстро развивающейся цифровой образовательной среде и направлен на решение задач социально-экономического развития страны в условиях четвертой промышленной революции и становления цифровой экономики.

Первая промышленная революция была связана с механизацией производства, вторая — с использованием электричества и конвейеров, третья — с электроникой и автоматизацией.

Четвертая опирается на достижения в области Интернета вещей, средств связи, машинного обучения, промышленных и бытовых роботов, на развитие новых моделей и сценариев взаимодействия, которые поддерживаются цифровыми технологиями. ( По материалам II РОССИЙСКО-КИТАЙСКАЯ КОНФЕРЕНЦИЯ 2019 г.).  
Данная работа выполняет задачи цифровой трансформации образования, поскольку в ней ведётся разработка информационной системы, предназначенной для цифровизации начального этапа обучения в учебном заведении, в его начальной части, которой является процедура зачисления в число студентов, приёма документов от абитуриентов и учёта результатов сдачи вступительных испытаний.

## 1 РАЗРАБОТКА ЭСКИЗНОГО И ТЕХНИЧЕСКОГО ПРОЕКТА ПРОГРАММНОГО МОДУЛЯ

### Назначение и область применения

Разрабатываемое программное обеспечение предполагает учёт абитуриентов, поступающих на специальности нашего учебного заведения, факта предоставления ими необходимых документов, фиксацию результатов вступительных испытаний, предусмотренных для специальности, на которую зачисляется абитуриент, выведение среднего балла и определения факта успешности сдачи.

### Технические характеристики

#### 1.2.1 Постановка задачи

Результатом данного курсового проекта должна быть программа, работающая на персональном компьютере, в которую член приёмной комиссии может вводить персональные данные поступающих учеников, отмечать факт подачи ими необходимых документов, и вводить результаты вступительных экзаменов, предусмотренных для специальности, для поступления на которую подал документы будущий студент. Должна быть предусмотрена функция печати данных, предоставленных данным абитуриентом, и функция подсчёта среднего и проходного балла для данной специальности.

#### 1.2.2 Требования к программе

Программа должна работать под управлением операционной системы Windows 10, использовать все возможности графической оболочки, которую предоставляет данная современная операционная система для удобного, наглядного ввода данных, исключения ошибок при вводе.

#### 1.2.3 Описание алгоритма

Запуск программы должен производится щелчком мышки по иконке, расположенной на рабочем столе пользователя. После запуска программы перед пользователем должно раскрыться главное окно программы, содержащее элементы управления, меню, кнопки, позволяющее перейти к списку абитуриентов, то есть лиц, желающих поступить в наш техникум и предоставивших о себе начальные сведения. Совокупность начальных сведений, требуемых для заведения данных об абитуриенте, представлены в следующей таблице:

|  |  |
| --- | --- |
| Фамилия |  |
| Имя |  |
| Отчество |  |
| Год рождения |  |
| Год окончания школы |  |
| Какую школу закончил |  |
| Избранная специальность |  |
| Код группы |  |
| Средний бал ЕГЭ |  |

После ввода начальных данных в базе данных программы создаётся запись для данного абитуриента, поля которой будут пополняться результатами приёмных испытаний. Рядом с результатами экзаменов должен отображаться средний балл. С помощью одного из элементов управления, расположенного на главном экране программе должна запускаться функция печати со стандартным набором возможностей печати на выбранный принтер или в файл заданного формата.

## 2 РАЗРАБОТКА ИНТЕРФЕЙСА

Для полноценного использования всех графических возможностей взаимодействия с пользователем, которые предоставляет Windows 10 разумно использовать графическую оболочку WPF(Windows Presentation Foundation), которая продвигается Microsoft в качестве основной для создания деск-топных приложений.

Вся работа с программой будет происходить через главное окно, снабжённое самым современным элементом интерфейса Ribbon (Лента), который уже стал стандартом для программ, работающих под Windows, по крайней мере, мы можем видеть его(её) во всех программах Microsoft Office. Магическое сочетание клавиш Ctrl+F1 заставляет появляться/исчезать этот удобный и универсальный элемент интерфейса, предоставляя доступ ко всем функциям программы.

Под лентой мы расположим поля ввода класса TextBox, через которые будем вводить информацию по каждому отдельному абитуриенту, а под ними расположим элемент управления DataGrid, который будет демонстрировать весь ведённый список абитуриентов. Мы предусматриваем также возможность дальнейшего увеличения объёма демонстрируемой информации, для чего можно будет включать дополнительные окна с помощью элемента управления Ribbon, вводя туда дополнительные меню и кнопки.

## 3 РАЗРАБОТКА РАБОЧЕГО ПРОЕКТА

Для разработки проекта первое, с чего мы начнём – это определимся с его названием. Поскольку мы проектируем и разрабатываем программное обеспечение для автоматизированной рабочей системы «Абитуриент», логично назвать проект сочетанием английских слов College Applicants, то есть поступающие в наш колледж студенты, сдавшие документы для поступления. Этим словом мы назовем проект в интегрированной среде разработки Visual Studio. Самой крупной единицей разработки в Visual Studio является решение, поэтому проект Applicant войдёт составной частью в решение, которое назовём Aerometallurgical college. Такая структура разрабатываемого проекта отвечает принципам расширяемости и гибкости, поскольку в будущем мы сможем включить в состав нашего решения новые проекты, обслуживающие дополнительные нужды нашего колледжа.

### 3.1 Разработка баз данных

Все введённые в нашей программе данные должны где-то хранится, причём так, чтобы они оставались в неприкосновенности после выключения компьютера и были доступны для переноса на другой компьютер, если в этом возникнет нужда, были что называется, persistent, не разрушались после окончания работы программы, перезагрузки или выключения питания на компьютере. Для надёжного хранения данных, удовлетворяющих этим требованиям были созданы и широко используются специальные программы управления базами данных, из которых наиболее распространены базы данных реляционного типа. Это базы, в которых информация хранится в виде таблиц, типа тех, которые используются в офисном приложении Excel. К системам управления базами данных реляционного типа относятся такие продукты, как SQL Server, PostgreSql, SQLite, все они могут использоваться на персональном компьютере для хранения вводимой информации.  
Мы сделаем выбор в пользу продукта Microsoft SQL Server, потому что выбрали язык C# этой фирмы для разработки, будем использовать программную среду этой же фирмы для разработки Visual Studio, и можем поэтому надеяться на обилие документации и удобство работы с этой системой управления базами данных (СуБД).

SQL Server назван так потому, что работа с таблицами базы данных ведётся в нём на языке запросов SQL. Это очень широко распространённый язык, в индексе популярности языков <https://www.tiobe.com/tiobe-index/> он занимает 10 строчку, но всё-таки работать на нём не так удобно, как на выбранном нами C#, и он требует дополнительного времени для глубокого освоения. Поэтому фирма Microsoft разработала специальный комплект программ, фреймворк, который называется Entity Framework, и через который работа с базой данных SQL Server происходит через обыкновенные классы языка C#. Мы будем использовать самую последнюю версию этого фреймворка, который называется Entity Framework Core.

При помощи этого набора классов, фреймворка, работа с базой данных происходит через работу над данными в памяти программы, представленными в виде экземпляров классов, то есть в виде наборов данных, собранных вместе для удобной работы.

3.2 Разработка модулей программы  
Модули программы, которые относятся к интерфейсу пользователя, формируются системой программирования Visual Studio автоматически, при начальном выборе типа проекта. Тип проекта при создании решения Aerometallurgical college мы выбираем как Приложение WPF, проект для создания приложения WPF .NET Core. Это самый современный вариант программной среды Microsoft, обеспечивающий максимальную переносимость на другие операционные системы, с высокой скоростью работы, в котором в наиболее полной мере должен быть учтён опыт эксплуатации предыдущих программных сред .NET Microsoft.

3.3 Описание программного модуля  
Итак, основным программным модулем является модуль главного окна программы MainWindow, который существует в виде двух файлов, что вначале формируется в виде почти пустых заготовок при создании проекта. Один из этих файлов образуется с расширением .xaml и является файлом главного окна, сгенерированным системой на специальном языке, описывающем внешний вид экрана, все окошки, кнопки, меню, который называется XAML.

Он очень похож на язык гипертекстовой разметки HTML, который лежит в основе описания WEB-страниц. Кроме этого файла, при работе главного окна используется дополнительно файл с расширением .xaml.cs, написанный на более обычном для программиста, а не дизайнера языке C#.

Каждой записи, сделанной в базу данных, соответствует некоторое действие над экземпляром класса, соответствующим этой записи. Поля класса представляют поля записи, и поэтому для таблицы базы данных, в которой мы будем хранить информацию о студентах, нужно иметь класс, для которого мы создадим отдельный файл с названием Applicant.cs (“Абитуриент”). Вот содержимое этого файла:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Security.Policy;

using System.Text;

using System.Threading.Tasks;

namespace CollegeApplicants

{

class Applicant

{

public int ApplicantId { get; set; }

public string Surname { get; set; }

public string GivenName { get; set; }

public string Patronimic { get; set; }

public string YearOfBirth { get; set; }

public string YearOfGraduation { get; set; }

public string GraduatedSchool { get; set; }

public string ChosenSpeciality { get; set; }

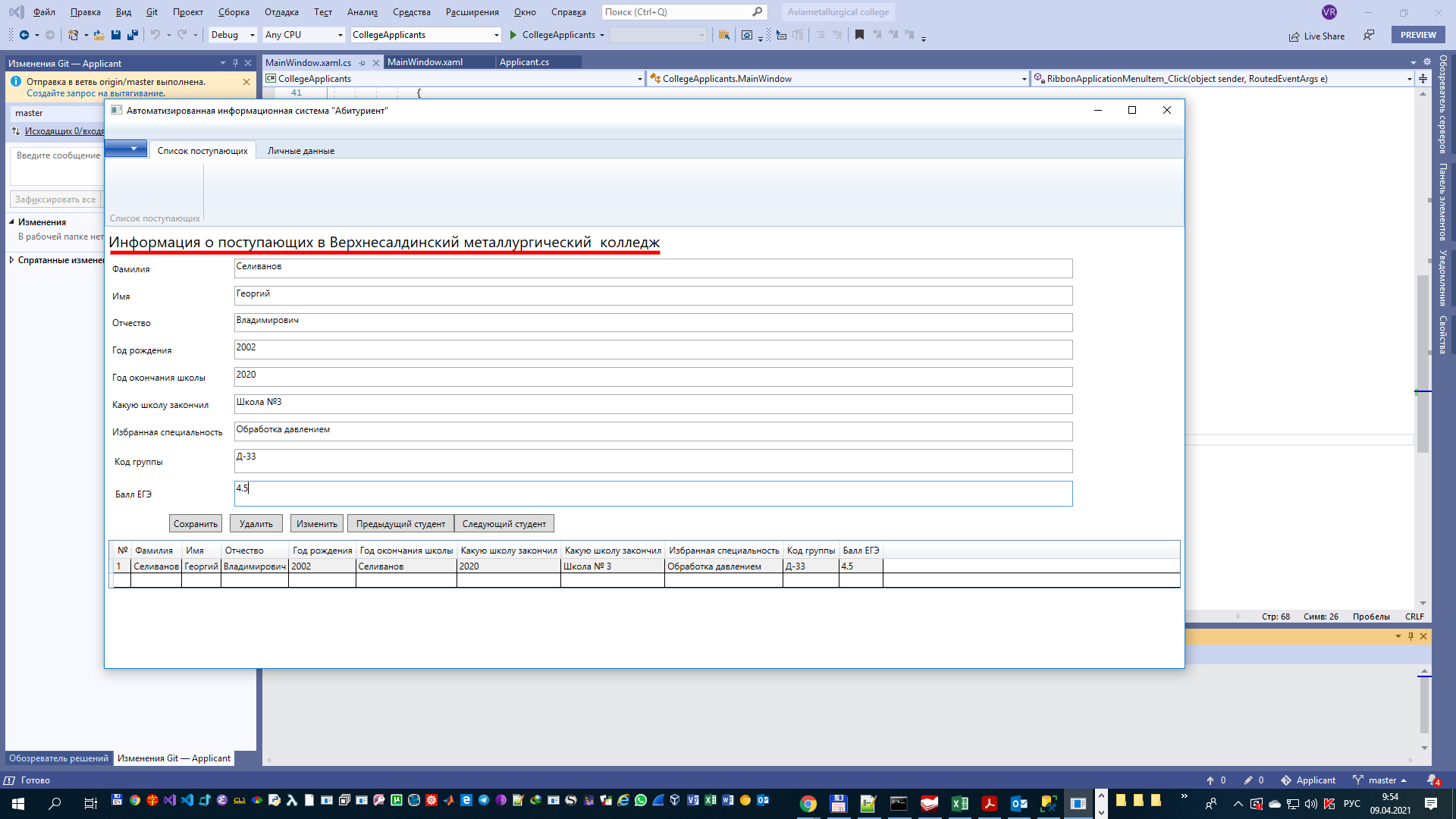
public string CodeOfClass { get; set; }

public string EGEGrade { get; set; }

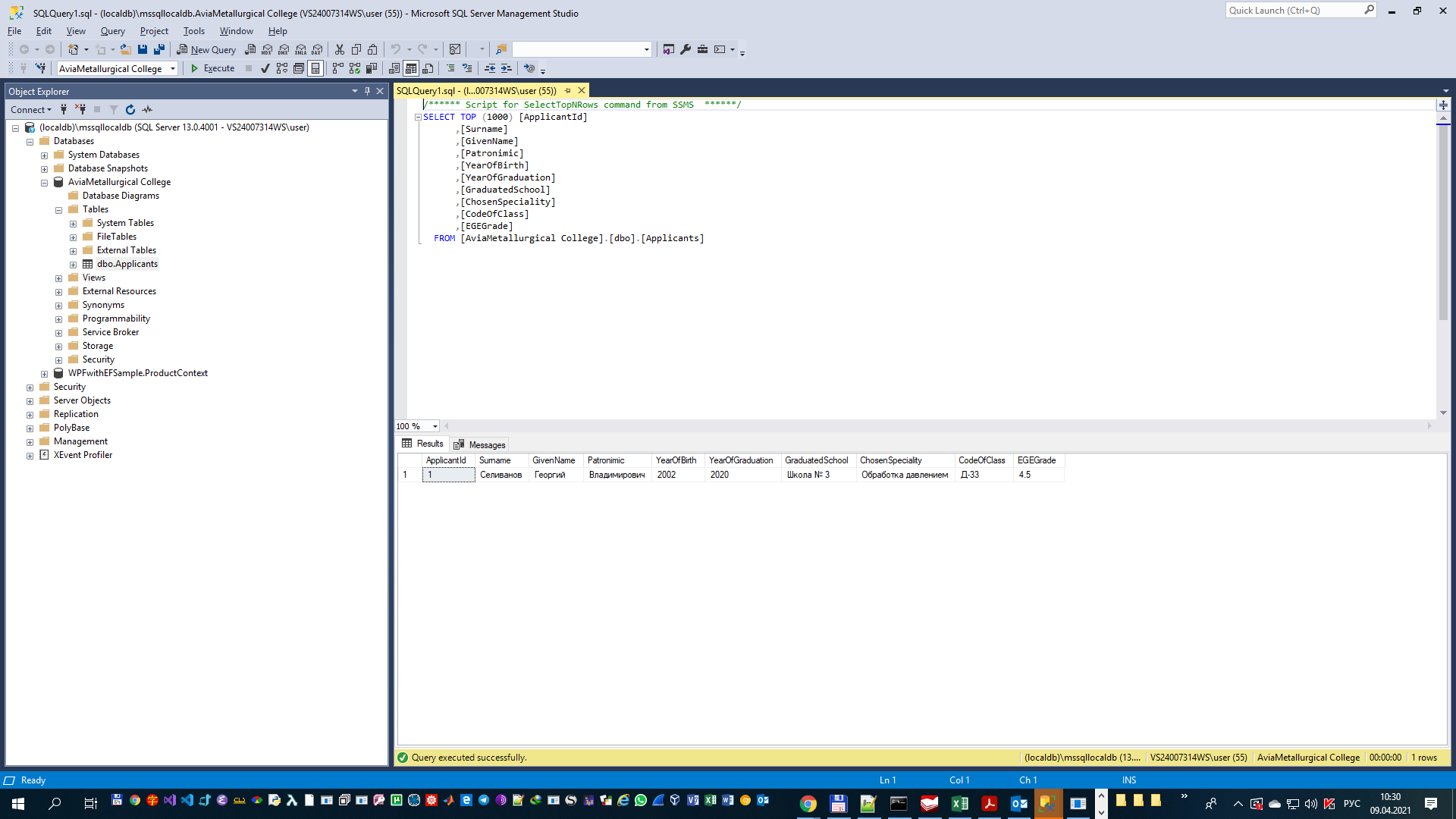
}

}

### 3.4 Тестирование программного модуля

Тестирование программы проводим, контролируя появление информации, которая вводится в поля ввода TextBox, в списке элемента управления DataGrid. Новая запись записывается из полей ввода в поля таблицы базы данных после нажатия кнопки Создать Кроме этого, удобно наблюдать за результатами произведённых над базой данных действий через программу Microsoft SQL Server Management Studio.

После использования нашей программы там появляется база данных AviaMetallurgical College, в составе которой находится таблица dbo.Applicants. Мы можем проверять содержимое этой таблицы, запуская запросы SQL, а также удалять базу данных, проверяя режим её создания в нашей программе.



## 4 РАЗРАБОТКА ДОКУМЕНТАЦИИ

## 4.1 Руководство пользователя

Использование программы благодаря интуитивно понятному для современного пользователя интерфейсу как самой программы, так и операционной системы Windows 10, под которой рекомендуется её эксплуатация, не требует особых навыков или объяснений.

После установки программы на компьютер пользователя её запуск будет производиться двойным щелчком левой клавиши мыши по иконке, выведенной на Рабочий стол компьютера. Ввод информации производится через стандартное использование клавиатуры и мыши на элементах управления главного окна программы. Прекращение работы программы производится стандартным закрытием окна или выбором пункта меню Выход.

При дальнейшем развитии и усложнении программы может потребоваться дополнение данного руководства, учитывающее появившиеся особенности, отличающие действия с программой от типичных.

## 4.2 Руководство программиста

Исходный код программы находится в репозитории GitHub по адресу

<https://github.com/Erroman/Applicant.git>. Заинтересованные в доработке лица должны иметь предпочтительно установленную на компьютере систему управления версиями Git ,с сайта <https://git-scm.com/>, а также Visual Studio 2019, где и был разработан указанный код. При установке Visual Studio 2019 должна быть выбрана загрузка «Хранение и обработка данных», после чего ожидается, что на компьютер будет установлена версия SQL Server LocalDB, с которой и будет работать наша программа.

Для целей отладки, чтобы убедиться, например, что версия SQL Server LocalDB действительно установлена, рекомендуется установить на компьютер программу «Microsoft SQL Server Management Studio 18.8». При запуске программы следует ввести в качестве имени сервера (localdb)\mssqllocaldb, для аутентификации оставив строку Windows Authentication.

Текст самой программы написан на языке C#, для продолжения разработки рекомендуется применять последнюю версию языка C# 9.0, которая поставляется вместе с последними выпусками Visual Studio 2019 с библиотекой Net 5.0 и Net 6.0 (пробная версия). При разработке проекта использовалась версия NET 6.0, которая осенью 2021 года должна стать основной средой разработки для языка C# на длительный срок.

В состав программы загружаются дополнительные библиотеки Microsoft.Entity Framework Core 5.0.5, Microsoft.EntityFrameworkCore.Proxies 5.0.5, Microsoft.EntityFrameworkCore.SqlServer 5.0.5 через диспетчер пакетов NuGet, которые позволяют осуществлять работу с SQL Server с помощью объектной модели данных Entity Framework.

Интерфейс программы написан на библиотеке WPF (Windows Presentation Foundation), переписанной для применения с версиями .NET Core, .NET 5, .NET 6. В качестве основной темы используется элемент управления Ribbon. До фиксации db72f830 этот элемент управления присутствовал в качестве образца в виде отдельного проекта для удобства разработки. Этот проект можно назначать исполняемым и изучать работы с Ribbon. В нашем проекте функции Ribbon почти не используются, но они могут понадобится в будущем. Здесь при разработке мы опираемся на принцип расширяемости, избыточности функций.

# ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы мною изучены и продемонстрированы возможности среды программирования Visual Studio 2019, входящего в её состав языка C# новейшей версии C# 9.0, приёмы отладки и тестирования программного обеспечения, приёмы работы с базой данных SQL Server, в том числе с применением объектной модели данных в виде сущностей, создания графической оболочки для взаимодействия с пользователем на основе библиотек WPF, новейшего элемента управления Ribbon. Получены навыки и продемонстрированы приёмы разработки технической документации, разработки алгоритма программной реализации поставленной задачи, создания программного модуля по разработанному алгоритму.

Программное обеспечение создавалось с применением распределённой системы управления версиями Git, что позволяет отследить все стадии выполнения проекта, в дальнейшем продолжить его развитие с привлечением новых участников.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. C# 9.0. Справочник. Полное описание языка Албахари Джозеф 2021 г.
2. Про C# 8.0 c .NET Core Основные принципы и практика программирования 9 издание Андрю Троелсен Фил Джапикс 2020 г.
3. Орлов С.А., Цилькер Б.Я. Технологии разработки программного обеспечения: Учебник для вузов. /4-е изд. 2012 г.
4. И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров Технология разработки программного обеспечения Учебное пособие по СПО 2019 г.

# ПРИЛОЖЕНИЕ А

Содержимое файла App.xaml, главного запускающего файла проекта:

<Application x:Class="Applicant.App"

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

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

xmlns:local="clr-namespace:Applicant"

StartupUri="MainWindow.xaml">

<Application.Resources>

</Application.Resources>

</Application>

Этот файл ссылается на главный файл окна, MainWindows.xaml, содержимое которого представлено ниже:

<Window x:Class=”CollegeApplicants.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:Applicant”

mc:Ignorable=”d”

Title=”Автоматизированная информационная система &quot;Абитуриент&quot;” Loaded=”Window\_Loaded”>

<Window.Resources>

<CollectionViewSource x:Key=”applicantsViewSource”/>

</Window.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height=”Auto”/>

<RowDefinition/>

</Grid.RowDefinitions>

<Ribbon x:Name=”RibbonWin” SelectedIndex=”0”>

<!—Help Pane, located at the right-hand side 🡪

<Ribbon.HelpPaneContent>

<RibbonButton />

</Ribbon.HelpPaneContent>

<Ribbon.ApplicationMenu>

<RibbonApplicationMenu>

<RibbonApplicationMenuItem Header=”Настройки” />

<RibbonApplicationMenuItem Header=”Выход из программы” Click=”RibbonApplicationMenuItem\_Click” />

</RibbonApplicationMenu>

</Ribbon.ApplicationMenu>

<!—Ribbon Tab #1: Home 🡪

<RibbonTab Header=”Список поступающих”>

<!—Home group🡪

<RibbonGroup x:Name=”ClipboardGroup” Header=”Список поступающих”>

</RibbonGroup>

</RibbonTab>

<!—Ribbon Tab #2: Launch 🡪

<RibbonTab Header=”Личные данные” KeyTip=”I”>

<!—Launch/Applications group🡪

<!—Employee And Payroll group🡪

<RibbonGroup x:Name=»Employee» Header=»Подробные сведения о поступающих в наш колледж»>

</RibbonGroup>

</RibbonTab>

</Ribbon>

<Grid Grid.Row=”1”>

<!—Defined GridRow🡪

<Grid.RowDefinitions>

<RowDefinition Height=”Auto”></RowDefinition>

<RowDefinition Height=”Auto”></RowDefinition>

<RowDefinition Height=”Auto”></RowDefinition>

<RowDefinition Height=”Auto”></RowDefinition>

<RowDefinition Height=”Auto”></RowDefinition>

<RowDefinition Height=”Auto”></RowDefinition>

<RowDefinition Height=”Auto”></RowDefinition>

<RowDefinition Height=”Auto”></RowDefinition>

<RowDefinition Height=”Auto”></RowDefinition>

<RowDefinition Height=”Auto”></RowDefinition>

<RowDefinition Height=”Auto”></RowDefinition>

<RowDefinition Height=”Auto”></RowDefinition>

<RowDefinition Height=”Auto”></RowDefinition>

<RowDefinition Height=”Auto”/>

<RowDefinition Height=”Auto”></RowDefinition>

<RowDefinition Height=”Auto”></RowDefinition>

<RowDefinition Height=”\*”></RowDefinition>

</Grid.RowDefinitions>

<!—Defined Column🡪

<Grid.ColumnDefinitions>

<ColumnDefinition Width=”Auto”></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition Width=”Auto”></ColumnDefinition>

</Grid.ColumnDefinitions>

<!—Stack Panel by Default Verticale🡪

<StackPanel Grid.Row=”0” Grid.Column=”0” Grid.ColumnSpan=”3”>

<Label Content=»Информация о поступающих в Верхнесалдинский металлургический колледж» FontSize=»20» ></Label>

<Canvas >

<Path Height=”5” Stroke=”Red” Width=”730” Canvas.Top=”-5” Canvas.Left=”5”

Stretch=”Fill” StrokeThickness=”5” Fill=”Blue” Data=”M61,125 L193,28”/>

</Canvas>

</StackPanel>

<!—First Column Content🡪

<Label Content=”Фамилия” Grid.Column=”0” Grid.Row=”1” Margin=”5”></Label>

<Label Content=”Имя” Grid.Column=”0” Grid.Row=”2” Margin=”5”></Label>

<Label Content=”Отчество” Grid.Column=”0” Grid.Row=”3” Margin=”5”></Label>

<Label Content=”Год рождения” Grid.Column=”0” Grid.Row=”4” Margin=”5”></Label>

<Label Content=”Год окончания школы” Grid.Column=”0” Grid.Row=”5” Margin=”5”></Label>

<Label Content=”Какую школу закончил” Grid.Column=”0” Grid.Row=”6” Margin=”5”></Label>

<Label Content=”Избранная специальность” Grid.Column=”0” Grid.Row=”7” Margin=”5”></Label>

<Label Content=”Код группы” Grid.Column=”0” Grid.Row=”8” Margin=”8”></Label>

<Label Content=”Балл ЕГЭ” Grid.Column=”0” Grid.Row=”9” Margin=”9”></Label>

<!—Second Column Content🡪

<TextBox Name=”txtNumber” Grid.Column=”1” Grid.Row=”1” Margin=”5”></TextBox>

<TextBox Name=”txtSurname” Grid.Column=”1” Grid.Row=”1” Margin=”5”></TextBox>

<TextBox Name=”txtFirstName” Grid.Column=”1” Grid.Row=”2” Margin=”5”></TextBox>

<TextBox Name=”txtGivenName” Grid.Column=”1” Grid.Row=”2” Margin=”5”></TextBox>

<TextBox Name=”txtPatronimic” Grid.Column=”1” Grid.Row=”3” Margin=”5”></TextBox>

<TextBox Name=”txtYearOfBirth” Grid.Column=”1” Grid.Row=”4” Margin=”5”></TextBox>

<TextBox Name=”txtYearOfGraduation” Grid.Column=”1” Grid.Row=”5” Margin=”5”></TextBox>

<TextBox Name=”txtGraduatedSchool” Grid.Column=”1” Grid.Row=”6” Margin=”5”></TextBox>

<TextBox Name=”txtChosenSpeciality” Grid.Column=”1” Grid.Row=”7” Margin=”5”></TextBox>

<TextBox Name=”txtCodeOfClass” Grid.Column=”1” Grid.Row=”8” Margin=”5”></TextBox>

<TextBox Name=”txtEGEGrade” Grid.Column=”1” Grid.Row=”9” Margin=”5”></TextBox>

<!—Stack Panel Horizental orientation🡪

<!—Wrap Panel List of button by default hoizental🡪

<WrapPanel Grid.Column=”0” Margin=”80,0,0,0” Grid.Row=”10” Grid.ColumnSpan=”4”>

<Button Content=”Сохранить” Width=”70” Height=”24” Margin=”5” Click=”Button\_Click”></Button>

<Button Content=”Удалить” Width=”70” Height=”24” Margin=”5”></Button>

<Button Content=”Изменить” Width=”70” Height=”24” Margin=”5”></Button>

<Button Content=”Предыдущий студент” Width=”141” Height=”24”/>

<Button Content=”Следующий студент” Width=”132” Height=”24”/>

</WrapPanel>

<!—DataGrid for Show the Employee Info🡪

<DataGrid ItemsSource=”{Binding Source={StaticResource applicantsViewSource}}” AutoGenerateColumns=”False” Grid.Row=”11” Grid.ColumnSpan=”4” Grid.Column=”0” Margin=”5”>

<DataGrid.Columns>

<DataGridTextColumn Header=”№” Binding=”{Binding ApplicantId}”></DataGridTextColumn>

<DataGridTextColumn Header=”Фамилия” Binding=”{Binding Surname}”></DataGridTextColumn>

<DataGridTextColumn Header=”Имя” Binding=”{Binding GivenName}”></DataGridTextColumn>

<DataGridTextColumn Header=”Отчество” Binding=”{Binding Patronimic}”></DataGridTextColumn>

<DataGridTextColumn Header=”Год рождения” Binding=”{Binding YearOfBirth}”></DataGridTextColumn>

<DataGridTextColumn Header=”Год окончания школы” Binding=”{Binding Surname}”></DataGridTextColumn>

<DataGridTextColumn Header=”Какую школу закончил” Binding=”{Binding YearOfGraduation}”></DataGridTextColumn>

<DataGridTextColumn Header=”Какую школу закончил” Binding=”{Binding GraduatedSchool}”></DataGridTextColumn>

<DataGridTextColumn Header=”Избранная специальность” Binding=”{Binding ChosenSpeciality}”></DataGridTextColumn>

<DataGridTextColumn Header=”Код группы” Binding=”{Binding CodeOfClass}”></DataGridTextColumn>

<DataGridTextColumn Header=”Балл ЕГЭ” Binding=”{Binding EGEGrade}”></DataGridTextColumn>

</DataGrid.Columns>

</DataGrid>

<!—Third Column Content🡪

<Image Grid.Column=”2” Grid.Row=”1” Grid.RowSpan=”5” Margin=”21,10” Width=”100” />

</Grid>

</Grid>

</Window>

Основной код программы на языке C# находится в файле кодовой поддержки окна с расширением .cs, так называемом code behind file, MainWindow.xaml.cs:

using Microsoft.EntityFrameworkCore;

using System;

using System.Collections.Generic;

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 CollegeApplicants

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

private CollectionViewSource applicantsViewSource;

public MainWindow()

{

InitializeComponent();

using (var context = new ApplicantContext())

{

context.Database.EnsureCreated();

}

applicantsViewSource = (CollectionViewSource)FindResource(nameof(applicantsViewSource));

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

using (var context = new ApplicantContext())

{

var newApplicant = new Applicant()

{

Surname = txtSurname.Text,

GivenName = txtGivenName.Text,

Patronimic = txtPatronimic.Text,

YearOfBirth = txtYearOfBirth.Text,

YearOfGraduation = txtYearOfGraduation.Text,

GraduatedSchool= txtGraduatedSchool.Text,

ChosenSpeciality=txtChosenSpeciality.Text,

CodeOfClass=txtCodeOfClass.Text,

EGEGrade = txtEGEGrade.Text

};

context.Applicants.Add(newApplicant);

context.SaveChanges();

}

}

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

using (var context = new ApplicantContext())

{

context.Applicants.Load<Applicant>();

applicantsViewSource.Source = context.Applicants.Local.ToObservableCollection();

}

}

private void RibbonApplicationMenuItem\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

}

}

Ниже приведён текст файлов, в которых расположены классы Applicant.cs и ApplicantContext.cs, отвечающих за взаимодействие с базой данных:

/\*Applicant.cs\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Security.Policy;

using System.Text;

using System.Threading.Tasks;

namespace CollegeApplicants

{

class Applicant

{

public int ApplicantId { get; set; }

public string Surname { get; set; }

public string GivenName { get; set; }

public string Patronimic { get; set; }

public string YearOfBirth { get; set; }

public string YearOfGraduation { get; set; }

public string GraduatedSchool { get; set; }

public string ChosenSpeciality { get; set; }

public string CodeOfClass { get; set; }

public string EGEGrade { get; set; }

}

}

/\*ApplicantContext.cs\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Microsoft.EntityFrameworkCore;

namespace CollegeApplicants

{

class ApplicantContext:DbContext

{

public DbSet<Applicant> Applicants { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseSqlServer(@"Server=(localdb)\mssqllocaldb;Database=AviaMetallurgical College");

}

}

}