МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное

учреждение высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности

**ОТЧЕТ**

О прохождении учебной практики

**Место прохождения** **практики** - *НОЦ «Математическое и программное обеспечение информационных систем реального времени»*

Вид практики Учебная практика

Тип практики Практика по получению первичных профессиональных умений и навыков

Способ проведения практики Стационарная

Форма проведения практики Дискретная (по видам и периодам практик)

**Обучающийся** Текутов Антон Павлович, очной формы обучения, направления 09.03.04 «Программная инженерия»

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Текутов А.П./*

**Руководитель практики от структурного подразделения ЮФУ** старший преподаватель кафедры МОП ЭВМ ИКТИБ ЮФУ, Родзина Ольга Николаевна

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Родзина О.Н./*

ОГЛАВЛЕНИЕ

[ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ 3](#_Toc515283520)

[ВВЕДЕНИЕ 4](#_Toc515283521)

[1 ПРОЕКТИРОВАНИЕ 5](#_Toc515283522)

[2 РЕАЛИЗАЦИЯ ПРОГРАММНОГО РЕШЕНИЯ 6](#_Toc515283523)

[2.1 Выбор инструментария 6](#_Toc515283524)

[2.2 Проектирование и планирование работы 6](#_Toc515283525)

[2.3 Проектирование и разработка пользовательского интерфейса 6](#_Toc515283526)

[2.4 База данных АИС оптимизации учебного процесса 12](#_Toc515283527)

[2.5 Реализация функции печати 13](#_Toc515283528)

[ЗАКЛЮЧЕНИЕ 14](#_Toc515283529)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 15](#_Toc515283530)

[Приложение 1. Файл StudentUsr.cs 16](#_Toc515283531)

[Приложение 2. Файл Tasks.cs 17](#_Toc515283532)

[Приложение 3. Файл TeacherUsr.cs 23](#_Toc515283533)

[Приложение 4. Файл TasksViewModel.cs 25](#_Toc515283534)

[Приложение 5. Файл TasksViewModelStudent.cs 28](#_Toc515283535)

[Приложение 6. Файл MainWindow.xaml.cs 30](#_Toc515283536)

[Приложение 7. Файл RegistrationWindow.xaml.cs 32](#_Toc515283537)

[Приложение 8. Файл WorkWindow.xaml.cs 35](#_Toc515283538)

[Приложение 9. Файл WorkWindowStudent.xaml.cs 39](#_Toc515283539)

# ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

АИС – автоматизированная информационная система

БД – база данных

ПО – программное обеспечение

СУБД – система управления базой данных

# ВВЕДЕНИЕ

Основными целями учебной практики при её прохождении являются расширение и закрепление теоретических знаний через получение первичных профессиональных навыков, ознакомление студентов с характером и спецификацией будущей деятельности.

Согласно индивидуальному заданию на практику, которое описано в дневнике практики, требуется спроектировать и разработать АИС для организации учебного процесса в школах.

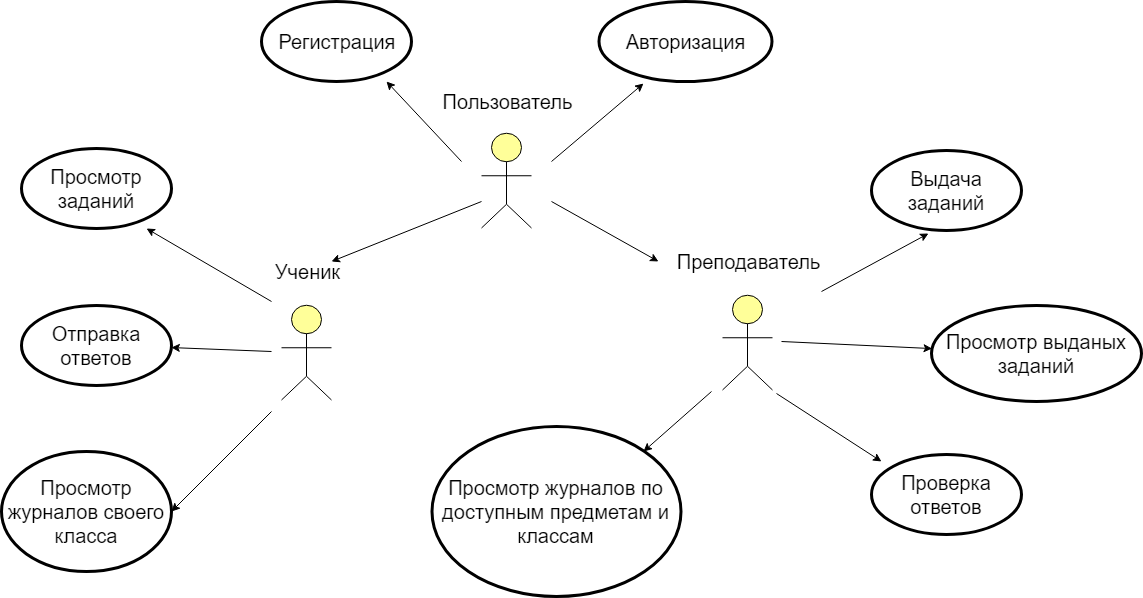
В настоящее время во многих школах все еще используется метод организации учебного процесса без использования информационных технологий для оптимизации процесса выдачи заданий и отслеживания их выполнения. Предлагается разработать АИС, которая будет помогать преподавателям и ученикам в организации учебного процесса, а также помогать составлять необходимые документы.

Разработка любой АИС с использованием БД включает в себя как минимум 3 этапа разработки: проектирование БД, разработка программного решения, тестирование разработанной системы.

# ПРОЕКТИРОВАНИЕ

Согласно индивидуальному заданию, требуется разработать АИС для организации учебного процесса в школах. Данная АИС должна обеспечивать возможность выдавать задания и отслеживать их выполнение и другие функции управления.

В результате анализа требований, проведённого на основе методических указаний из учебного пособия «Анализ требований к АИС», была составлена диаграмма прецедентов, представленная на рисунке 1.

   
Рисунок 1 – Диаграмма вариантов использования.

В системе должно быть предусмотрено 2 вида пользователей: *ученик* и *преподаватель*. Ученик может выполнять следующие функции: функция просмотра выданных преподавателями заданий с возможностью фильтрации по предмету; функция отправки ответа на задание с возможностью загрузить дополнительные файлы; функция просмотра журнала выполнения заданий по классу, в котором обучается пользователь. Преподаватель может выполнять следующие функции: функция выдачи заданий по доступным для преподавателя предметам и классам; функция просмотра выданных данным пользователем заданий; функция проверки ответов учеников на выданные пользователем задания; функция просмотра журналов выполнения заданий по доступным классам и предметам Последующая разработка АИС будет опираться на составленную диаграмму вариантов использования.

# РЕАЛИЗАЦИЯ ПРОГРАММНОГО РЕШЕНИЯ

## Выбор инструментария

Для разработки данного проекта была выбрана среда разработки Visual Studio 2017 [2], использовался язык С# [3], потому что в учебном процессе уже был опыт выполнения заданий на данном языке, кроме того был использован язык разметки XAML[4]. При разработке использовался паттерн MVVM[6], цель использования – отделить логику работы программы и дынные от их визуальной составляющей, что позволяет гибко менять пользовательский интерфейс, а значит подготавливает продукт к изменениям с течением времени. Хранение данных осуществляется с помощью базы данных на основе системы управления базами данных Microsoft Office Access [5].

## Проектирование и планирование работы

Одним из этапов индивидуального задания является – спроектировать и разработать программное решение, оптимизирующее процесс обучения в школах, а именно позволяет преподавателю быстро выдавать задания, проверять ответы на них и отслеживать выполнение в журнале журнале, а ученикам быстро сориентироваться в учебном процессе. Данное решение должно предоставлять возможность удобно использовать все его функции, а также разграничивать некоторые функции по уровню доступа. То есть ученик не может использовать функции, доступные только преподавателю (подробнее см. п.1.1).

Разработка программного решения производится в несколько этапов: проектирование и разработка пользовательского интерфейса (View); создание описания фундаментальных данных (Model), необходимых для работы приложения, а также их модели отображения (ViewModel) в соответствии с паттерном проектирования MVVM и реализация их загрузки из базы данных; реализация функции печати данных. Подробная информация по каждому из этапов представлена ниже.

## Проектирование и разработка пользовательского интерфейса

На данном этапе был разработан пользовательский интерфейс системы организации учебного для удобного взаимодействия с разрабатываемой АИС. В соответствии с требованиями, интерфейс должен быть органичен и все окна должны быть выполнены в одном стиле. Архитектура интерфейса должна быть интуитивно понятной и максимально простой.

При запуске программы открывается окно авторизации (рис. 2) с полями для ввода логина и пароля. Пользователю необходимо ввести свои данные, а затем нажать на кнопку «Войти»

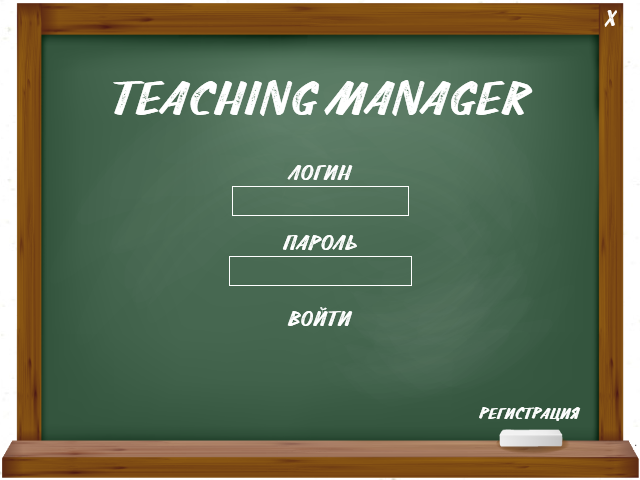


Рисунок 2 – Окно авторизации.

Если пользователь еще не зарегистрирован, то ему следует нажать на кнопку «Регистрация», после этого откроется окно регистрации (рис. 3). Необходимо заполнить все поля и нажать на кнопку «Зарегистрироваться», иначе появится сообщение об ошибке, также можно вернуться на страницу авторизации нажав кнопку «Авторизация».

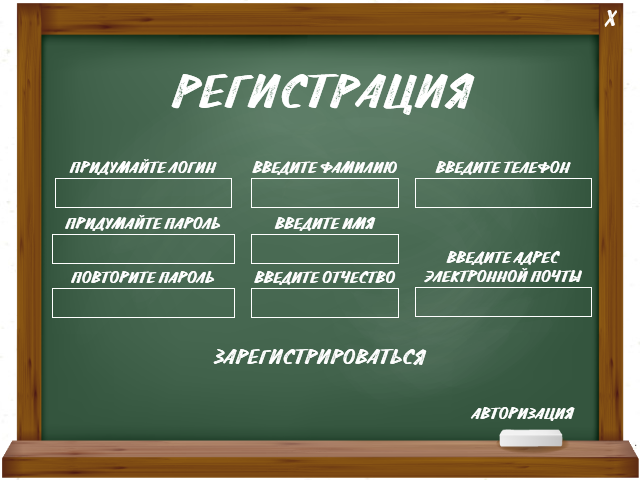


Рисунок 3 – Окно регистрации.

Если авторизуется пользователь с типом доступа «Преподаватель», то в открывшемся окне ему будут доступны следующий функционал: форма выдачи задания (рис. 4), форма просмотра выданных заданий (рис. 5), форма проверки ответов (рис. 6) и журнал выполнения заданий (рис. 7). Переключение производится при помощи кнопок на панели слева.

Чтобы выдать задание необходимо выбрать класс и предмет из доступных данному преподавателю, выбрать дату сдачи задания (дата выдачи проставляется автоматически), указать тему работы и текст задания, также по желанию можно добавить дополнительные файлы изображений.

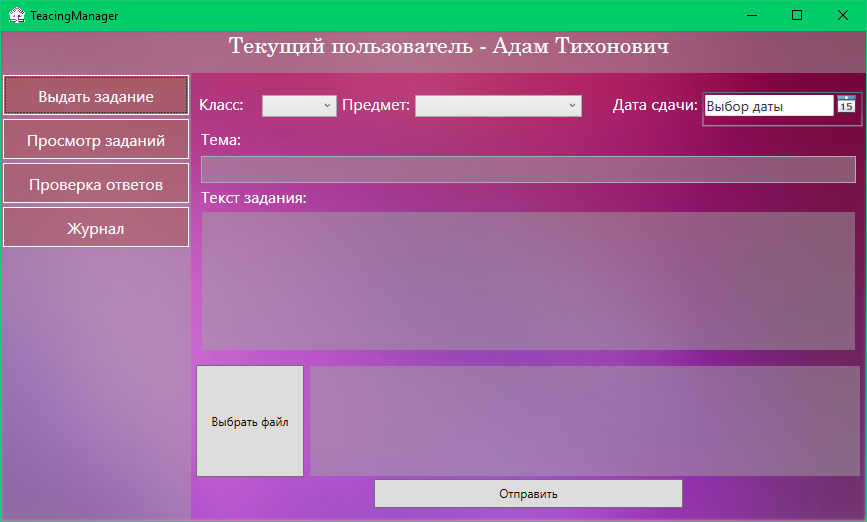


Рисунок 4 – Окно для пользователя с типом доступа «Преподаватель»; форма выдачи задания.

На форме просмотра выданных заданий можно в хронологическом порядке посмотреть выданные данным пользователем задания, в информации о каждом задании указывается класс, которому оно было выдано, предмет, тема задания, дата выдачи и дата сдачи. При нажатии на кнопку слева от информации о задании можно просмотреть текст задания, файлы изображений, а также приступить к проверке этого задания нажав кнопку «Проверить». Примечательно, что изображения можно увеличить кликнув по ним. Кроме этого, можно отфильтровать задания по классу либо предмету, либо классу и предмету одновременно. Чтобы снова показать все доступные задания необходимо нажать на кнопку «Сброс фильтра».

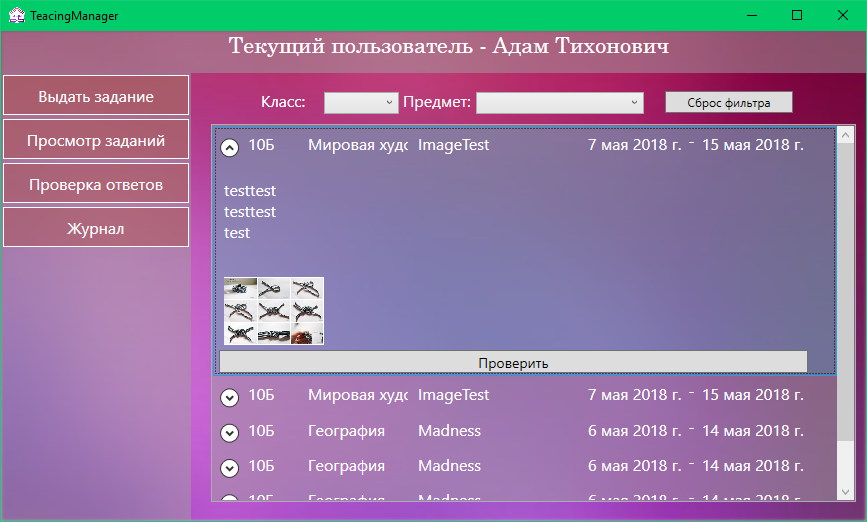


Рисунок 5 – Окно для пользователя с типом доступа «Преподаватель»; форма просмотра выданных заданий.

По запросу на проверку какого-либо задания пользователь попадает на форму проверки ответов, где может посмотреть и проверить ответы учеников. Для удобства в верхней части окна отображается само задание. Чтобы оценить работу ученика преподавателю нужно на форме слева указать оценку, по желанию можно написать комментарий. Изображения, загруженные учениками также можно увеличивать.

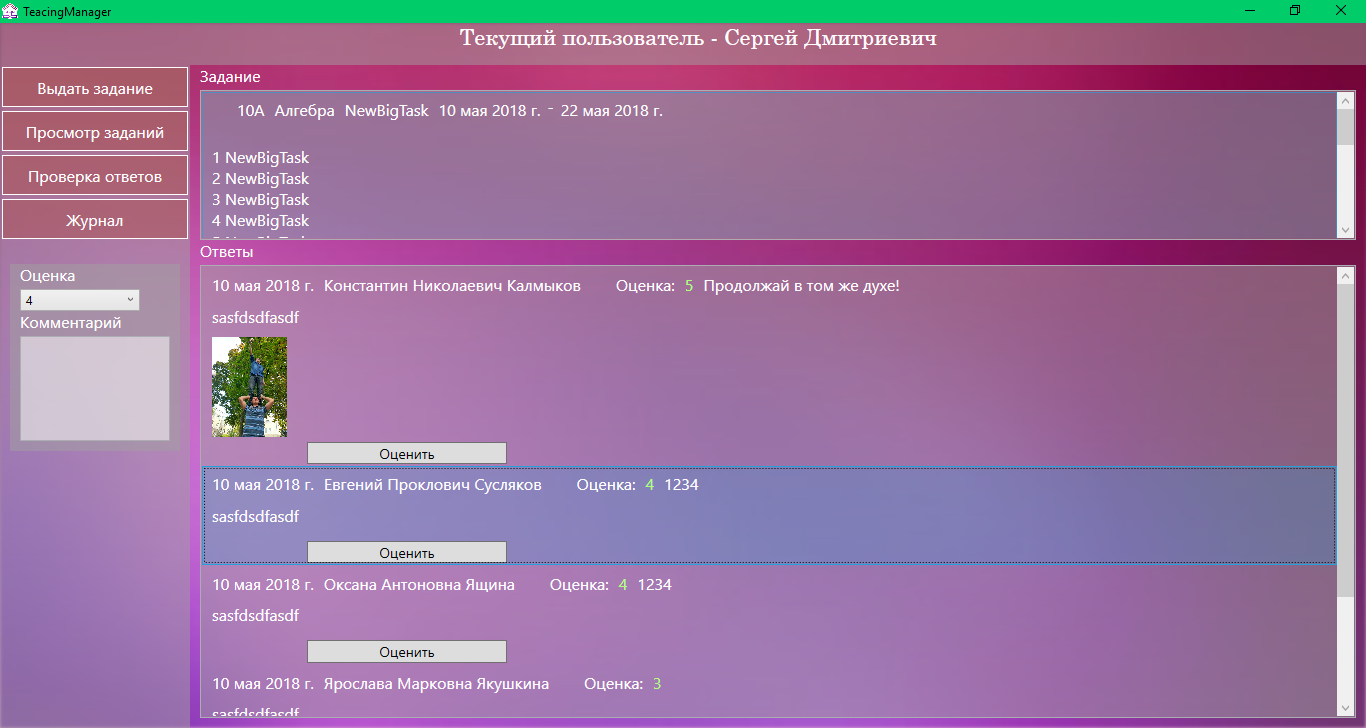


Рисунок 6 – Окно для пользователя с типом доступа «Преподаватель»; форма проверки ответов.

Чтобы просмотреть статистику выполнения заданий по какому-либо классу преподавателю необходимо нажать на кнопку «Журнал», выбрать класс и предмет, а затем нажать кнопку «Обновить». Преподаватель может распечатать текущий журнал кликнув по кнопке «Распечатать»



Рисунок 7 – Окно для пользователя с типом доступа «Преподаватель»; журнал выполнения заданий.

Если авторизуется пользователь с типом доступа «Ученик», то в открывшемся окне ему будут доступны следующий функционал: форма отправки ответа (рис. 9), форма просмотра заданий (рис. 8) и журнал выполнения заданий (рис. 10). Переключение производится при помощи кнопок на панели слева.

На форме просмотра заданий можно в хронологическом порядке посмотреть выданные данным пользователем задания, в информации о каждом задании указывается класс, которому оно было выдано, предмет, тема задания, дата выдачи и дата сдачи. При нажатии на кнопку слева от информации о задании можно просмотреть текст задания, файлы изображений, а также отправить ответ на выбранное задание нажав кнопку «Проверить». Кроме этого, можно отфильтровать задания предмету, Чтобы снова показать все доступные задания необходимо нажать на кнопку «Сброс фильтра».

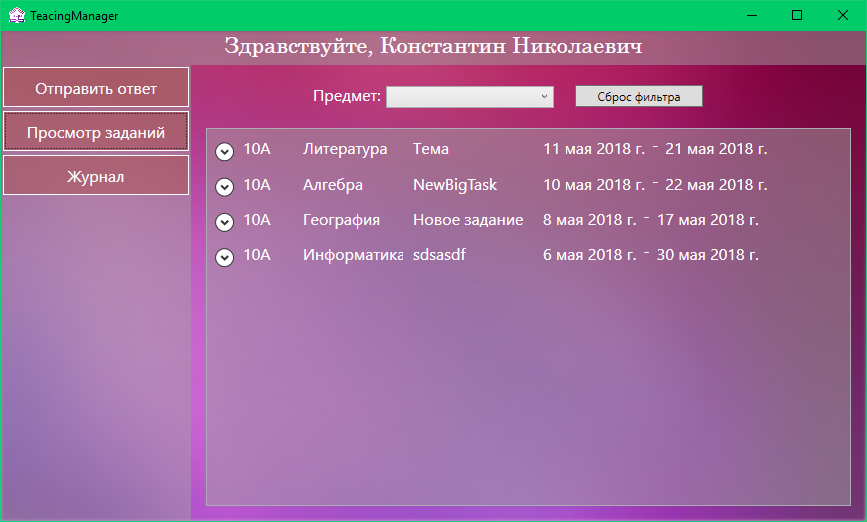


Рисунок 8 – Окно для пользователя с типом доступа «Ученик»; форма просмотра заданий.

На форме отправки ответа для удобства отвечающего находится текст задания, поле для ввода текста ответа, осуществлена возможность загрузки дополнительных файлов изображений. Ответ будет загружен в базу данных по нажатию на кнопку «Отправить», причем если ответ на данное задание уже был отправлен этим учеником, то он заменится более поздней версией

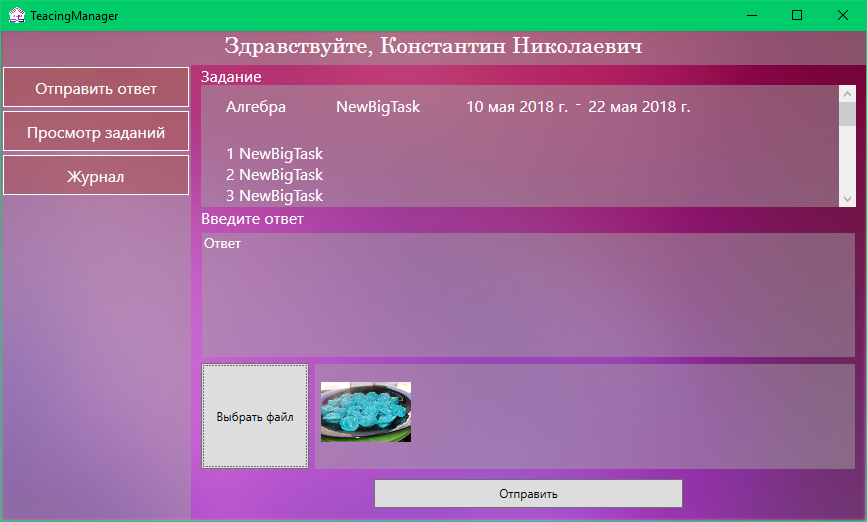


Рисунок 9 – Окно для пользователя с типом доступа «Ученик»; форма отправки ответа.

Журнал выполнения заданий аналогичен журналу учителя, за исключением того, что нет фильтрации по классу – обусловлено это тем, что ученик обучается только в одном конкретном классе, а значит информация о других классах ему не нужна.

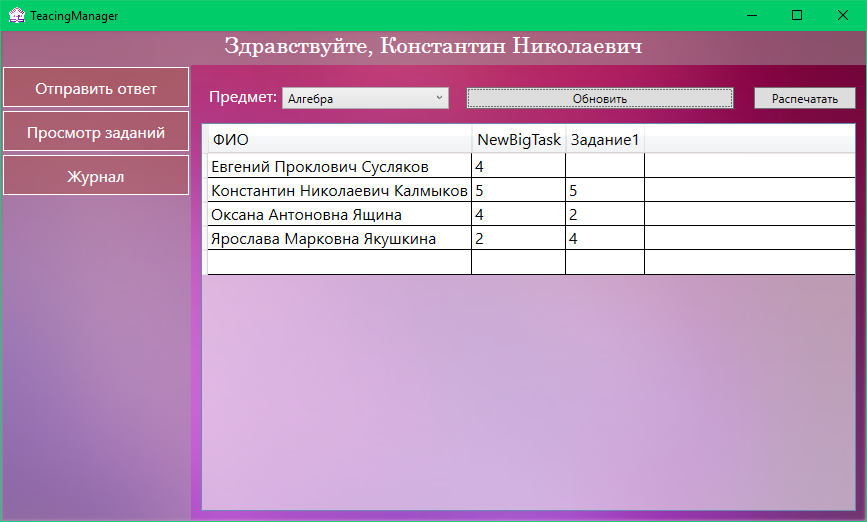


Рисунок 10 – Окно для пользователя с типом доступа «Ученик»; журнал выполнения заданий.

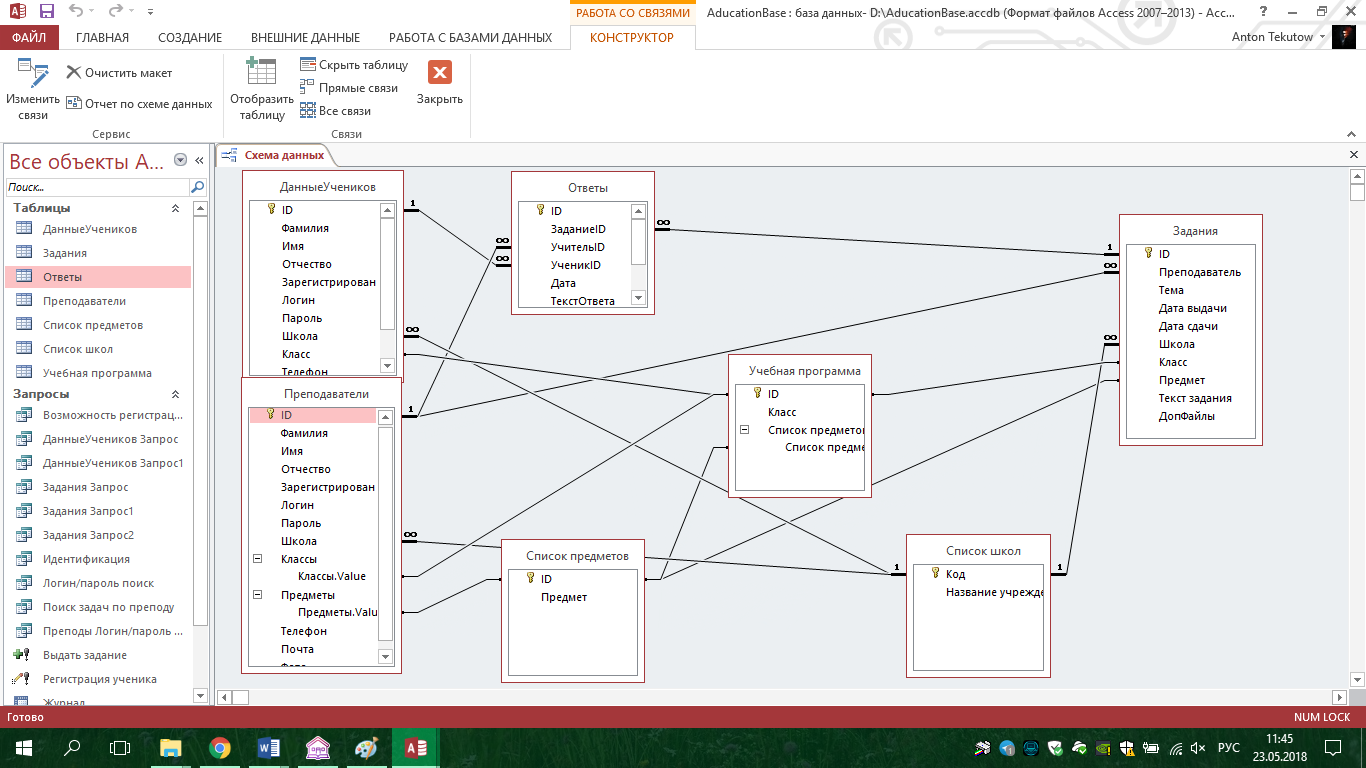
В результате проектирования и разработки пользовательского интерфейса получен практически полностью разработанный интерфейс системы.

## База данных АИС оптимизации учебного процесса

Все данные, которые нужно хранить в базе данных (перечень предметов, вопросы по тестам и результаты прохождения тестов) сохраняются в программе Microsoft Access. Связь программы с базой данных происходит с помощью запросов SQL.

Для работы разрабатываемой АИС была спроектирована и разработана база данных, которая содержит в себе всю необходимую информацию преподавателях, учениках, учебной программе, школах, выданных зданиях и ответах на них.

В результате проектирования БД, была составлена диаграмма сущностей (рис. 11), по которой была разработана сама БД.

  
Рисунок 11 – Диаграмма сущностей

В спроектированной БД составлены 7 таблиц:

* Данные Учеников – В данной таблице храниться информация об учениках, как зарегистрированных, так и просто числящихся в организации, но еще не использовавших данную АИС.
* Преподаватели – В данной таблице храниться информация о преподавателях, как зарегистрированных, так и просто числящихся в организации, но еще не использовавших данную АИС.
* Список школ – В данной таблице храниться информации об учебных учреждениях.
* Список предметов – Таблица содержит список предметов, которые могут преподаваться в школах.
* Учебная программа – Таблица содержит информацию о списках предметов, которые преподаются в различных классах.
* Задания – В данной таблице хранится информация о заданиях.
* Ответы – Таблица содержит ответы учеников.

## Реализация функции печати

В соответствии с индивидуальным заданием, последним этапом разработки является реализация печати данных. Данная функция была реализована для вывода на печать информации из журналов выполнения заданий

# ЗАКЛЮЧЕНИЕ

В ходе учебной практики была реализована АИС организации учебного процесса в школах. Были изучены навыки работы с базой данных, языком разметки XAML, и паттерном проектирования MVVM, закреплены знания в области программирования на C#. Все цели, задачи и условия прохождения практики были выполнены в полном объеме.

Программа полностью готова для эксплуатирования. В дальнейшем есть возможность расширения функционал, не входящие в условия поставленного технического задания.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Учебная практика - <https://methodological_terms.academic.ru/2186/%D0%A3%D0%A7%D0%95%D0%91%D0%9D%D0%90%D0%AF_%D0%9F%D0%A0%D0%90%D0%9A%D0%A2%D0%98%D0%9A%D0%90> (Интернет-ресурс)
2. Visual Studio 2017 - <https://www.microsoft.com/ru-ru/SoftMicrosoft/vs2017> (Microsoft)
3. C Sharp - <https://ru.wikipedia.org/wiki/C_Sharp> (Википедия)
4. XAML <https://docs.microsoft.com/ru-ru/windows/uwp/xaml-platform/xaml-overview>
5. Создание таблиц в базе данных - <https://support.office.com/ru-ru/article/%D0%A1%D0%BE%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5-%D1%82%D0%B0%D0%B1%D0%BB%D0%B8%D1%86-%D0%B2-%D0%B1%D0%B0%D0%B7%D0%B5-%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85-8fdc65f9-8d40-4ff5-9212-80e6545e8d87> (Microsoft)
6. Паттерн MVVM <https://ru.wikipedia.org/wiki/Model-View-ViewModel>(Википедия)

## Приложение 1. Файл StudentUsr.cs

using System;

using System.Collections.Generic;

using System.Data.OleDb;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace StartWindow.Data

{

class StudentUsr

{

public static int ID { get; set; }

public static string Surname { get; set; }

public static string Name { get; set; }

public static string Patronymic { get; set; }

public static int SchoolID { get; set; }

public static string SchoolStr { get; set; }

public static int KlassID { get; set; }

public static string Klass { get; set; }

public static List<int> SubjectsID { get; set; }

public static List<string> Subjects { get; set; }

public static System.Windows.Controls.Image Photo { get; set; }

public static List<int> StringToListInt(string buff)

{

List<int> currentList = new List<int>();

string substr = "";

while (buff.Length > 0)

{

if (buff[0] >= '0' && buff[0] <= '9')

{

substr += buff[0];

buff = buff.Substring(1);

}

else if (buff[0] == ';')

{

currentList.Add(Convert.ToInt32(substr));

substr = "";

buff = buff.Substring(1);

}

}

try

{

currentList.Add(Convert.ToInt32(substr));

substr = "";

buff = buff.Substring(1);

}

catch { }

return currentList;

}

public static string GetTextWithID(string TableName, string columtName, int ID)

{

string connectString = "provider=Microsoft.ACE.Oledb.12.0;Data Source= D:\\AducationBase.accdb;";

OleDbConnection connection = new OleDbConnection(connectString);

string query = " SELECT [" + TableName + "]." + columtName +

" FROM[" + TableName + "]" +

" WHERE((([" + TableName + "].ID) = " + Convert.ToString(ID) + "));";

OleDbCommand command = new OleDbCommand(query, connection);

connection.Open();

string answer = command.ExecuteScalar().ToString();

connection.Close();

return answer;

}

public static int GetIDWithText(string TableName, string columtName, string Text)

{

string connectString = "provider=Microsoft.ACE.Oledb.12.0;Data Source= D:\\AducationBase.accdb;";

OleDbConnection connection = new OleDbConnection(connectString);

string query = " SELECT [" + TableName + "].ID" +

" FROM[" + TableName + "]" +

" WHERE( ([" + TableName + "]." + columtName + ") = \"" + Text + "\");";

OleDbCommand command = new OleDbCommand(query, connection);

connection.Open();

int answer = (int)command.ExecuteScalar();

connection.Close();

return answer;

}

public static void GetStudentUsr(string id)

{

string connectString = "provider=Microsoft.ACE.Oledb.12.0;Data Source= D:\\AducationBase.accdb;";

OleDbConnection connection = new OleDbConnection(connectString);

string query = "SELECT ДанныеУчеников.ID, ДанныеУчеников.Фамилия, ДанныеУчеников.Имя, ДанныеУчеников.Отчество, ДанныеУчеников.Школа, ДанныеУчеников.Класс " +

"FROM ДанныеУчеников WHERE(((ДанныеУчеников.ID) = "+id+" ));";

OleDbCommand command = new OleDbCommand(query, connection);

connection.Open();

OleDbDataReader dataReader = command.ExecuteReader();

dataReader.Read();

ID = (int)dataReader["ID"];

Name = (string)dataReader["Имя"];

Surname = (string)dataReader["Фамилия"];

Patronymic = (string)dataReader["Отчество"];

SchoolID = (int)dataReader["Школа"];

KlassID = (int)dataReader["Класс"];

Klass = GetTextWithID("Учебная программа", "Класс", KlassID);

Subjects = new List<string>();

query = "SELECT [Учебная программа].[Список предметов] FROM [Учебная программа] WHERE(([Учебная программа].ID) = " + KlassID + ");";

command = new OleDbCommand(query, connection);

dataReader = command.ExecuteReader();

dataReader.Read();

SubjectsID = StringToListInt((string)dataReader["Список предметов"]);

for (int i = 0; i < SubjectsID.Count; i++)

{

Subjects.Add(GetTextWithID("Список предметов", "Предмет", SubjectsID[i]));

}

connection.Close();

}

}

}

## Приложение 2. Файл Tasks.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.OleDb;

using System.Windows.Media;

using System.Windows.Controls;

namespace StartWindow.Data

{

public class IDWithText

{

public int ID { get; set; }

public String Text { get; set; }

}

public class TaskFileImage

{

public string directory { get; set; }

public Image image { get; set; }

public ImageSource IMGS { get; set; }

public TaskFileImage(string dirBuff)

{

directory = dirBuff;

ImageSourceConverter imageSourceConverter = new ImageSourceConverter();

image = new Image();

image.SetValue(System.Windows.Controls.Image.SourceProperty, imageSourceConverter.ConvertFromString(directory));

IMGS = image.Source;

}

}

public static class NewTask

{

public static int TeacherID { get; set; }

public static string Title { get; set; }

public static DateTime DateEnd { get; set; }

public static int SchoolID { get; set; }

public static int KlassID { get; set; }

public static string Klass { get; set; }

public static int SubjectID { get; set; }

public static string Subject { get; set; }

public static string TaskText { get; set; }

public static List<TaskFileImage> TaskFiles { get; set; } = new List<TaskFileImage>();

}

public static class NewTaskAnswer

{

public static int TeacherID { get; set; } = -1;

public static int StudentID { get; set; } = -1;

public static int TaskID { get; set; } = -1;

public static string AnswerText { get; set; } = "";

public static List<TaskFileImage> AnswerFiles { get; set; } = new List<TaskFileImage>();

public static bool UnloadAnswer()

{

if (!(TeacherID!= -1&& StudentID != -1 && TaskID != -1 && AnswerText!= ""))

{

return false;

}

string connectString = "provider=Microsoft.ACE.Oledb.12.0;Data Source= D:\\AducationBase.accdb;";

OleDbConnection connection = new OleDbConnection(connectString);

string query = " SELECT Ответы.ID " +

" FROM Ответы " +

" WHERE( ((Ответы.ЗаданиеID) = "+TaskID + " ) AND ( (Ответы.УченикID) = " + StudentID + " ));";

OleDbCommand command = new OleDbCommand(query, connection);

connection.Open();

string answerID = "";

try

{

answerID = command.ExecuteScalar().ToString();

}

catch

{}

if (answerID != "")

{

query = "DELETE Ответы.ID" +

" FROM Ответы" +

" WHERE((Ответы.ID) = "+answerID+");";

command = new OleDbCommand(query, connection);

command.ExecuteScalar();

}

connection.Close();

if (AnswerFiles.Count > 0)

{

string Date = Convert.ToString(DateTime.Today.ToString("dd.MM.yyyy"));

query = "INSERT INTO Ответы(ЗаданиеID, УчительID, УченикID, Дата, ТекстОтвета, ДопФайлы) " +

"values( '" + TaskID + "', '" + TeacherID + "', '" + StudentID + "', '" + Date + "', '" + AnswerText + "' , ? )";

if (answerID == "")

{

}

command = new OleDbCommand(query, connection);

OleDbParameter oleDbParameter = new OleDbParameter("ДопФайлы", OleDbType.VarBinary);

string fileName = AnswerFiles[0].directory; //Путь к файлу

System.Drawing.Image image = System.Drawing.Image.FromFile(fileName); //Изображение из файла.

System.IO.MemoryStream memoryStream = new System.IO.MemoryStream(); //Поток в который запишем изображение

image.Save(memoryStream, System.Drawing.Imaging.ImageFormat.Bmp); //Сохраняем изображение в поток.

oleDbParameter.Value = memoryStream.ToArray(); //Устанавливаем значение параметра

command.Parameters.Add(oleDbParameter);

connection.Open();

command.ExecuteNonQuery();

connection.Close();

memoryStream.Dispose();

}

else

{

string Date = Convert.ToString(DateTime.Today.ToString("dd.MM.yyyy"));

query = "INSERT INTO Ответы(ЗаданиеID, УчительID, УченикID, Дата, ТекстОтвета) " +

"values( '" + TaskID + "', '" + TeacherID + "', '" + StudentID + "', '" + Date + "', '" + AnswerText + "' )";

command = new OleDbCommand(query, connection);

connection.Open();

command.ExecuteNonQuery();

connection.Close();

}

return true;

}

}

class Answer

{

public int ID { get; set; }

public int TeacherID { get; set; }

public string TeacherFIO { get; set; } = "";

public int StudentID { get; set; }

public string StudentFIO { get; set; } = "";

public int TaskID { get; set; }

public DateTime Date { get; set; }

public string DateStr { get; set; }

public string AnswerText { get; set; }

public Image AnswerImage { get; set; }

public ImageSource IMGS { get; set; }

public string Mark { get; set; }

public string Comment { get; set; }

public void GetFIO()

{

string connectString = "provider=Microsoft.ACE.Oledb.12.0;Data Source= D:\\AducationBase.accdb;";

OleDbConnection connection = new OleDbConnection(connectString);

string query = "SELECT ДанныеУчеников.Фамилия, ДанныеУчеников.Имя, ДанныеУчеников.Отчество" +

" FROM ДанныеУчеников WHERE(((ДанныеУчеников.ID) = " + StudentID + " ));";

OleDbCommand command = new OleDbCommand(query, connection);

connection.Open();

OleDbDataReader dataReader = command.ExecuteReader();

while (dataReader.Read())

{

StudentFIO += (string)dataReader["Имя"] + " " + (string)dataReader["Отчество"] + " " + (string)dataReader["Фамилия"];

}

query = "SELECT Преподаватели.Фамилия, Преподаватели.Имя, Преподаватели.Отчество" +

" FROM Преподаватели WHERE(((Преподаватели.ID) = " + TeacherID + " ));";

command = new OleDbCommand(query, connection);

dataReader = command.ExecuteReader();

while (dataReader.Read())

{

TeacherFIO += (string)dataReader["Имя"] + " " + (string)dataReader["Отчество"] + " " + (string)dataReader["Фамилия"];

}

connection.Close();

}

public static List<Answer> GetAnswers()

{

Tasks buff = ViewModel.TasksViewModel.CurrentTask;

string connectString = "provider=Microsoft.ACE.Oledb.12.0;Data Source= D:\\AducationBase.accdb;";

OleDbConnection connection = new OleDbConnection(connectString);

string query = "SELECT Ответы.ID, Ответы.ЗаданиеID, Ответы.УчительID, Ответы.УченикID, Ответы.Дата, Ответы.ТекстОтвета, Ответы.ДопФайлы, Ответы.Оценка, Ответы.Комментарий" +

" FROM Ответы WHERE( ((Ответы.ЗаданиеID) = " + buff.ID + " ) AND((Ответы.УчительID) = " + Data.TeacherUsr.ID + " ) );";

OleDbCommand command = new OleDbCommand(query, connection);

connection.Open();

OleDbDataReader dataReader = command.ExecuteReader();

List<Answer> result = new List<Answer>();

System.IO.Directory.CreateDirectory("ImageBuffFolder");

while (dataReader.Read())

{

try

{

Answer buffAnswer = new Answer();

buffAnswer.ID = (int)dataReader["ID"];

buffAnswer.TeacherID = (int)dataReader["УчительID"];

buffAnswer.StudentID = (int)dataReader["УченикID"];

buffAnswer.TaskID = (int)dataReader["ЗаданиеID"];

buffAnswer.Date = (DateTime)dataReader["Дата"];

buffAnswer.DateStr = ((DateTime)(dataReader["Дата"])).ToLongDateString();

buffAnswer.AnswerText = (string)dataReader["ТекстОтвета"];

try

{

buffAnswer.Mark = (string)dataReader["Оценка"];

}

catch

{

buffAnswer.Mark = "Не оценено";

}

try

{

buffAnswer.Comment = (string)dataReader["Комментарий"];

}

catch

{

buffAnswer.Comment = "";

}

buffAnswer.GetFIO();

try

{

System.IO.MemoryStream memoryStream = new System.IO.MemoryStream();

memoryStream.Write((byte[])dataReader["ДопФайлы"], 0, ((byte[])dataReader["ДопФайлы"]).Length);

System.Drawing.Image image = System.Drawing.Image.FromStream(memoryStream);

image.Save(@"ImageBuffFolder\" + buffAnswer.ID + "answ.BMP");

memoryStream.Dispose();

ImageSourceConverter imageSourceConverter = new ImageSourceConverter();

buffAnswer.AnswerImage = new Image();

buffAnswer.AnswerImage.SetValue(System.Windows.Controls.Image.SourceProperty, imageSourceConverter.ConvertFromString(@"ImageBuffFolder\" + buffAnswer.ID + "answ.BMP"));

buffAnswer.IMGS = buffAnswer.AnswerImage.Source;

}

catch { }

result.Add(buffAnswer);

}

catch { }

}

try

{

NewTaskAnswer.TaskID = result[0].ID;

}

catch

{

NewTaskAnswer.TaskID = -1;

}

return result;

}

public void MarkTheAnswer(string mark, string comment)

{

Mark = mark;

Comment = comment;

string connectString = "provider=Microsoft.ACE.Oledb.12.0;Data Source= D:\\AducationBase.accdb;";

OleDbConnection connection = new OleDbConnection(connectString);

string query = "UPDATE Ответы SET Ответы.Оценка = \"" + mark + "\", Ответы.Комментарий = \"" + comment + "\" WHERE((Ответы.ID) = " + ID + ");";

OleDbCommand command = new OleDbCommand(query, connection);

connection.Open();

command.ExecuteNonQuery();

connection.Close();

}

}

class Tasks

{

public int ID { get; set; }

public int TeacherID { get; set; }

public string Title { get; set; }

public DateTime DateBegin { get; set; }

public DateTime DateEnd { get; set; }

public string DateBeginStr { get; set; }

public string DateEndStr { get; set; }

public int SchoolID { get; set; }

public int KlassID { get; set; }

public string Klass { get; set; }

public int SubjectID { get; set; }

public string Subject { get; set; }

public string TaskText { get; set; }

public object ImageBuff { get; set; }

public Image TaskImage { get; set; }

public ImageSource IMGS { get; set; }

public static List<Tasks> GetTasks()

{

string connectString = "provider=Microsoft.ACE.Oledb.12.0;Data Source= D:\\AducationBase.accdb;";

OleDbConnection connection = new OleDbConnection(connectString);

string query = "SELECT Задания.ID, Задания.Преподаватель, Задания.Тема, Задания.[Дата выдачи], Задания.[Дата сдачи], Задания.Школа, Задания.Класс, Задания.Предмет, Задания.[Текст задания], Задания.ДопФайлы FROM Задания ORDER BY Задания.[Дата выдачи] DESC ;";

OleDbCommand comand = new OleDbCommand(query, connection);

connection.Open();

OleDbDataReader dataReader = comand.ExecuteReader();

List<Tasks> result = new List<Tasks>();

System.IO.Directory.CreateDirectory("ImageBuffFolder");

while (dataReader.Read())

{

try

{

result.Add

(new Tasks

{

ID = (int)dataReader["ID"],

TeacherID = (int)dataReader["Преподаватель"],

Title = (string)dataReader["Тема"],

DateBegin = (DateTime)dataReader["Дата выдачи"],

DateEnd = (DateTime)dataReader["Дата сдачи"],

DateBeginStr = ((DateTime)(dataReader["Дата выдачи"])).ToLongDateString(),

DateEndStr = ((DateTime)dataReader["Дата сдачи"]).ToLongDateString(),

SchoolID = (int)dataReader["Школа"],

KlassID = (int)dataReader["Класс"],

Klass = StartWindow.Data.TeacherUsr.GetTextWithID("Учебная программа", "Класс", (int)dataReader["Класс"]),

SubjectID = (int)dataReader["Предмет"],

Subject = StartWindow.Data.TeacherUsr.GetTextWithID("Список предметов", "Предмет", (int)dataReader["Предмет"]),

TaskText = (string)dataReader["Текст задания"],

});

System.IO.MemoryStream memoryStream = new System.IO.MemoryStream();

memoryStream.Write((byte[])dataReader["ДопФайлы"], 0, ((byte[])dataReader["ДопФайлы"]).Length);

System.Drawing.Image image = System.Drawing.Image.FromStream(memoryStream);

image.Save(@"ImageBuffFolder\" +result[result.Count - 1].ID+".BMP");

memoryStream.Dispose();

ImageSourceConverter imageSourceConverter = new ImageSourceConverter();

result[result.Count - 1].TaskImage = new Image();

result[result.Count - 1].TaskImage.SetValue(System.Windows.Controls.Image.SourceProperty, imageSourceConverter.ConvertFromString(@"ImageBuffFolder\" + result[result.Count - 1].ID + ".BMP"));

result[result.Count - 1].IMGS = result[result.Count - 1].TaskImage.Source;

}

catch { }

}

try

{

NewTaskAnswer.TaskID = result[0].ID;

}

catch

{

NewTaskAnswer.TaskID = -1;

}

return result;

}

}

public static class CurrentTask

{

private static List<Tasks> currentTaskList = new List<Tasks>();

}

}

## Приложение 3. Файл TeacherUsr.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.OleDb;

namespace StartWindow.Data

{

public class TeacherUsr

{

public static int ID { get; set; }

public static string Surname { get; set; }

public static string Name { get; set; }

public static string Patronymic { get; set; }

public static int SchoolID { get; set; }

public static string SchoolStr { get; set; }

public static List<int> KlassID { get; set; }

public static List<string> Klass { get; set; }

public static List<int> SubjectsID { get; set; }

public static List<string> Subjects { get; set; }

public static System.Windows.Controls.Image Photo { get; set; }

public static List<int> StringToListInt(string buff)

{

List<int> currentList = new List<int>();

string substr = "";

while (buff.Length > 0)

{

if (buff[0] >= '0' && buff[0] <= '9')

{

substr += buff[0];

buff = buff.Substring(1);

}

else if (buff[0] == ';')

{

currentList.Add(Convert.ToInt32(substr));

substr = "";

buff = buff.Substring(1);

}

}

try

{

currentList.Add(Convert.ToInt32(substr));

substr = "";

buff = buff.Substring(1);

}

catch { }

return currentList;

}

public static string GetTextWithID(string TableName, string columtName, int ID)

{

string connectString = "provider=Microsoft.ACE.Oledb.12.0;Data Source= D:\\AducationBase.accdb;";

OleDbConnection connection = new OleDbConnection(connectString);

string query = " SELECT [" + TableName + "]." + columtName +

" FROM[" + TableName + "]" +

" WHERE((([" + TableName + "].ID) = " + Convert.ToString(ID) + "));";

OleDbCommand command = new OleDbCommand(query, connection);

connection.Open();

string answer = command.ExecuteScalar().ToString();

connection.Close();

return answer;

}

public static int GetIDWithText(string TableName, string columtName, string Text)

{

string connectString = "provider=Microsoft.ACE.Oledb.12.0;Data Source= D:\\AducationBase.accdb;";

OleDbConnection connection = new OleDbConnection(connectString);

string query = " SELECT [" + TableName + "].ID"+

" FROM[" + TableName + "]" +

" WHERE( ([" + TableName + "]."+ columtName +") = \"" + Text + "\");";

OleDbCommand command = new OleDbCommand(query, connection);

connection.Open();

int answer = (int)command.ExecuteScalar();

connection.Close();

return answer;

}

public static void GetTeacherUsr(string id)

{

string connectString = "provider=Microsoft.ACE.Oledb.12.0;Data Source= D:\\AducationBase.accdb;";

OleDbConnection connection = new OleDbConnection(connectString);

string query = "SELECT Преподаватели.ID, Преподаватели.Фамилия, Преподаватели.Имя, Преподаватели.Отчество, Преподаватели.Школа, Преподаватели.Классы, Преподаватели.Предметы, Преподаватели.Фото FROM Преподаватели WHERE((Преподаватели.ID) = " + id + ");";

OleDbCommand command = new OleDbCommand(query, connection);

connection.Open();

OleDbDataReader dataReader = command.ExecuteReader();

dataReader.Read();

ID = (int)dataReader["ID"];

Name = (string)dataReader["Имя"];

Surname = (string)dataReader["Фамилия"];

Patronymic = (string)dataReader["Отчество"];

SchoolID = (int)dataReader["Школа"];

KlassID = StringToListInt((string)dataReader["Классы"]);

SubjectsID = StringToListInt((string)dataReader["Предметы"]);

Klass = new List<string>();

Subjects = new List<string>();

for (int i = 0; i < KlassID.Count; i++)

{

Klass.Add(GetTextWithID("Учебная программа", "Класс", KlassID[i]));

}

for (

int i = 0; i < SubjectsID.Count; i++)

{

Subjects.Add(GetTextWithID("Список предметов", "Предмет", SubjectsID[i]));

}

connection.Close();

}

}

}

## Приложение 4. Файл TasksViewModel.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.ComponentModel;

using System.Windows.Data;

namespace StartWindow.ViewModel

{

class TasksViewModel : DependencyObject

{

private static string id;

public static void SetUsrID(string IDBuff)

{

id = IDBuff;

}

public static ICollectionView TaskFilesViewBuff;

public string SubjectFilter

{

get { return (string)GetValue(SubjectFilterProperty); }

set { SetValue(SubjectFilterProperty, value); }

}

// Using a DependencyProperty as the backing store for FilterText. This enables animation, styling, binding, etc...

public static readonly DependencyProperty SubjectFilterProperty =

DependencyProperty.Register("SubjectFilter", typeof(string), typeof(TasksViewModel), new PropertyMetadata("", FilterTextChanged));

public string KlassFilter

{

get { return (string)GetValue(KlassFilterProperty); }

set { SetValue(KlassFilterProperty, value); }

}

// Using a DependencyProperty as the backing store for FilterText. This enables animation, styling, binding, etc...

public static readonly DependencyProperty KlassFilterProperty =

DependencyProperty.Register("KlassFilter", typeof(string), typeof(TasksViewModel), new PropertyMetadata("", FilterTextChanged));

public ICollectionView Items

{

get { return (ICollectionView)GetValue(MyPropertyProperty); }

set { SetValue(MyPropertyProperty, value); }

}

public static readonly DependencyProperty MyPropertyProperty =

DependencyProperty.Register("MyProperty", typeof(ICollectionView), typeof(TasksViewModel), new PropertyMetadata(null));

public ICollectionView ItemsUsrKlass

{

get { return (ICollectionView)GetValue(UsrKlassProperty); }

set { SetValue(UsrKlassProperty, value); }

}

public static readonly DependencyProperty UsrKlassProperty = DependencyProperty.Register("UsrKlassProperty", typeof(ICollectionView), typeof(TasksViewModel), new PropertyMetadata(null));

public ICollectionView ItemsUsrSubjects

{

get { return (ICollectionView)GetValue(UsrSubjectsProperty); }

set { SetValue(UsrSubjectsProperty, value); }

}

public static readonly DependencyProperty UsrSubjectsProperty =

DependencyProperty.Register("UsrSubjectsProperty", typeof(ICollectionView), typeof(TasksViewModel), new PropertyMetadata(null));

public ICollectionView ItemsTaskFiles

{

get { return (ICollectionView)GetValue(ItemsTaskFilesProperty); }

set { SetValue(ItemsTaskFilesProperty, value); }

}

public static readonly DependencyProperty ItemsTaskFilesProperty =

DependencyProperty.Register("ItemsTaskFilesProperty", typeof(ICollectionView), typeof(TasksViewModel), new PropertyMetadata(null));

public TasksViewModel()

{

StartWindow.Data.TeacherUsr.GetTeacherUsr(id);

Items = CollectionViewSource.GetDefaultView(StartWindow.Data.Tasks.GetTasks());

Items.Filter = FilterTasks;

ItemsUsrKlass = CollectionViewSource.GetDefaultView( StartWindow.Data.TeacherUsr.Klass);

ItemsUsrSubjects = CollectionViewSource.GetDefaultView( StartWindow.Data.TeacherUsr.Subjects);

ItemsTaskFiles = CollectionViewSource.GetDefaultView(StartWindow.Data.NewTask.TaskFiles);

TaskFilesViewBuff = ItemsTaskFiles;

}

public static ICollectionView RefrashItems()

{

return CollectionViewSource.GetDefaultView(Data.Tasks.GetTasks());

}

public ICollectionView CurrentTaskItem

{

get { return (ICollectionView)GetValue(CurrentTaskItemProperty); }

set { SetValue(CurrentTaskItemProperty, value); }

}

public static readonly DependencyProperty CurrentTaskItemProperty =

DependencyProperty.Register("CurrentTaskItemProperty", typeof(ICollectionView), typeof(TasksViewModel), new PropertyMetadata(null));

public static Data.Tasks CurrentTask;

public static ICollectionView RefrashCurrentTask(Data.Tasks CurrentTaskBuff)

{

CurrentTask = CurrentTaskBuff;

return CollectionViewSource.GetDefaultView(new List<Data.Tasks> {CurrentTask} );

}

public ICollectionView CurrentTaskAnswersItem

{

get { return (ICollectionView)GetValue(CurrentTaskAnswersProperty); }

set { SetValue(CurrentTaskAnswersProperty, value); }

}

public static readonly DependencyProperty CurrentTaskAnswersProperty =

DependencyProperty.Register("CurrentTaskAnswersProperty", typeof(ICollectionView), typeof(TasksViewModel), new PropertyMetadata(null));

public static ICollectionView RefrashCurrentTaskAnswers()

{

return CollectionViewSource.GetDefaultView(Data.Answer.GetAnswers());

}

bool FilterTasks(object obj)

{

bool result = true;

StartWindow.Data.Tasks current = obj as StartWindow.Data.Tasks;

if(current== null)

{

result = false;

}

if (SubjectFilter!="" && current.Subject != SubjectFilter)

{

result = false;

}

try

{

if (current.TeacherID != Data.TeacherUsr.ID)

{

result = false;

}

}

catch { }

if (KlassFilter != "" && current.Klass != KlassFilter)

{

result = false;

}

return result;

}

private static void FilterTextChanged(DependencyObject d, DependencyPropertyChangedEventArgs e)

{

var current = d as TasksViewModel;

if (current != null)

{

current.Items.Filter = null;

current.Items.Filter = current.FilterTasks;

}

}

}

}

## Приложение 5. Файл TasksViewModelStudent.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.ComponentModel;

using System.Windows.Data;

namespace StartWindow.ViewModel

{

class TasksViewModelStudent : DependencyObject

{

private static string id;

public static void SetUsrID(string IDBuff)

{

id = IDBuff;

}

public static ICollectionView TaskFilesViewBuff;

public string SubjectFilter

{

get { return (string)GetValue(SubjectFilterProperty); }

set { SetValue(SubjectFilterProperty, value); }

}

// Using a DependencyProperty as the backing store for FilterText. This enables animation, styling, binding, etc...

public static readonly DependencyProperty SubjectFilterProperty =

DependencyProperty.Register("SubjectFilter", typeof(string), typeof(TasksViewModelStudent), new PropertyMetadata("", FilterTextChanged));

public ICollectionView Items

{

get { return (ICollectionView)GetValue(MyPropertyProperty); }

set { SetValue(MyPropertyProperty, value); }

}

public static readonly DependencyProperty MyPropertyProperty =

DependencyProperty.Register("MyProperty", typeof(ICollectionView), typeof(TasksViewModelStudent), new PropertyMetadata(null));

public ICollectionView ItemsUsrSubjects

{

get { return (ICollectionView)GetValue(UsrSubjectsProperty); }

set { SetValue(UsrSubjectsProperty, value); }

}

public static readonly DependencyProperty UsrSubjectsProperty =

DependencyProperty.Register("UsrSubjectsProperty", typeof(ICollectionView), typeof(TasksViewModelStudent), new PropertyMetadata(null));

public ICollectionView ItemsAnswerFiles

{

get { return (ICollectionView)GetValue(ItemsAnswerFilesProperty); }

set { SetValue(ItemsAnswerFilesProperty, value); }

}

public static readonly DependencyProperty ItemsAnswerFilesProperty =

DependencyProperty.Register("ItemsAnswerFilesProperty", typeof(ICollectionView), typeof(TasksViewModel), new PropertyMetadata(null));

public ICollectionView ItemAnswer

{

get { return (ICollectionView)GetValue(ItemAnswerProperty); }

set { SetValue(ItemAnswerProperty, value); }

}

public static DependencyProperty ItemAnswerProperty =

DependencyProperty.Register("ItemAnswerProperty", typeof(ICollectionView), typeof(TasksViewModelStudent), new PropertyMetadata(null));

public TasksViewModelStudent()

{

StartWindow.Data.StudentUsr.GetStudentUsr(id);

Items = CollectionViewSource.GetDefaultView(StartWindow.Data.Tasks.GetTasks());

Items.Filter = FilterTasks;

ItemsUsrSubjects = CollectionViewSource.GetDefaultView(StartWindow.Data.StudentUsr.Subjects);

ItemsAnswerFiles = CollectionViewSource.GetDefaultView(StartWindow.Data.NewTaskAnswer.AnswerFiles);

TaskFilesViewBuff = ItemsAnswerFiles;

}

bool FilterTasks(object obj)

{

bool result = true;

StartWindow.Data.Tasks current = obj as StartWindow.Data.Tasks;

if(current== null)

{

result = false;

}

if (current.SchoolID != Data.StudentUsr.SchoolID)

{

result = false;

}

if (current.KlassID != Data.StudentUsr.KlassID)

{

result = false;

}

if (SubjectFilter!="" && current.Subject != SubjectFilter)

{

result = false;

}

return result;

}

private static void FilterTextChanged(DependencyObject d, DependencyPropertyChangedEventArgs e)

{

var current = d as TasksViewModelStudent;

if (current != null)

{

current.Items.Filter = null;

current.Items.Filter = current.FilterTasks;

}

}

}

}

## Приложение 6. Файл MainWindow.xaml.cs

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;

using System.Data.OleDb;

namespace StartWindow

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

private OleDbConnection myConnection;

public static string connectString = "provider=Microsoft.ACE.Oledb.12.0;Data Source= D:\\AducationBase.accdb;";

public static string userID;

public MainWindow()

{

InitializeComponent();

myConnection = new OleDbConnection(connectString);

myConnection.Open();

}

public bool LoginCheck(string login, string password)

{

string query = "SELECT DISTINCTROW ДанныеУчеников.ID FROM ДанныеУчеников WHERE(((ДанныеУчеников.Зарегистрирован) = True)) GROUP BY ДанныеУчеников.ID, ДанныеУчеников.Логин, ДанныеУчеников.Пароль HAVING(((ДанныеУчеников.Логин) =\"" + login + "\") AND((ДанныеУчеников.Пароль) =\"" + password + "\"));";

OleDbCommand command = new OleDbCommand(query, myConnection);

string answer;

try

{

answer = command.ExecuteScalar().ToString();

}catch

{

answer = "0";

}

if (answer != "0")

{

userID = answer;

return true;

}

query = "SELECT DISTINCTROW Преподаватели.ID FROM Преподаватели WHERE(((Преподаватели.Зарегистрирован) = True)) GROUP BY Преподаватели.ID, Преподаватели.Логин, Преподаватели.Пароль HAVING(((Преподаватели.Логин) =\"" + login + "\") AND((Преподаватели.Пароль) =\"" + password + "\"));";

command = new OleDbCommand(query, myConnection);

try

{

answer = command.ExecuteScalar().ToString();

}

catch

{

answer = "0";

}

if (answer != "0")

{

userID = 't'+answer;

return true;

}

else return false;

}

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

{

}

private void Window\_Closed(object sender, EventArgs e)

{

// заркываем соединение с БД

myConnection.Close();

}

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

{

this.Close();

}

private void Rectangle\_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

this.DragMove();

}

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

{

bool checkData = LoginCheck(LoginBox.Text, PasswordBox.Password);

if (checkData)

{

if (userID[0] != 't')

{

StartWindow.ViewModel.TasksViewModelStudent.SetUsrID(userID);

StartWindow.WorkWindowStudent Window = new StartWindow.WorkWindowStudent();

Window.Show();

this.Close();

}

else

{

userID = userID.Substring(1);

StartWindow.ViewModel.TasksViewModel.SetUsrID(userID);

WorkWindow Window = new WorkWindow();

Window.Show();

this.Close();

}

}

else

{

MessageBox.Show("Введены некорректные данные", "!!!Внимание!!!");

}

}

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

{

Window1 Window = new Window1();

Window.Show();

this.Close();

}

}

}

## Приложение 7. Файл RegistrationWindow.xaml.cs

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.Shapes;

using System.Data.OleDb;

namespace StartWindow

{

/// <summary>

/// Логика взаимодействия для Window1.xaml

/// </summary>

public partial class Window1 : Window

{

private OleDbConnection myConnection;

public static string connectString = "provider=Microsoft.ACE.Oledb.12.0;Data Source= D:\\AducationBase.accdb;";

public static string userID;

private bool teacher = false;

public Window1()

{

InitializeComponent();

myConnection = new OleDbConnection(connectString);

myConnection.Open();

}

public bool RegistrCheck(string name, string surname, string patronymic)

{

string query = "SELECT DISTINCTROW ДанныеУчеников.Зарегистрирован FROM ДанныеУчеников GROUP BY ДанныеУчеников.Зарегистрирован, ДанныеУчеников.Имя, ДанныеУчеников.Фамилия, ДанныеУчеников.Отчество HAVING(((ДанныеУчеников.Имя) =\"" + name + "\") AND((ДанныеУчеников.Фамилия) =\"" + surname + "\") AND((ДанныеУчеников.Отчество) =\"" + patronymic + "\"));";

OleDbCommand command = new OleDbCommand(query, myConnection);

string answer;

try

{

answer = command.ExecuteScalar().ToString();

}

catch

{

answer = "no inform";

}

if (answer == "False") return true;

query = "SELECT DISTINCTROW Преподаватели.Зарегистрирован FROM Преподаватели GROUP BY Преподаватели.Зарегистрирован, Преподаватели.Имя, Преподаватели.Фамилия, Преподаватели.Отчество HAVING(((Преподаватели.Имя) =\"" + name + "\") AND((Преподаватели.Фамилия) =\"" + surname + "\") AND((Преподаватели.Отчество) =\"" + patronymic + "\"));";

command = new OleDbCommand(query, myConnection);

try

{

answer = command.ExecuteScalar().ToString();

}

catch

{

answer = "no inform";

}

if (answer == "False")

{

teacher = true;

return true;

}

else return false;

}

public bool LoginCheck(string login, string password)

{

string query = "SELECT DISTINCTROW Count(ДанныеУчеников.Зарегистрирован) AS [Count-Зарегистрирован] FROM ДанныеУчеников WHERE(((ДанныеУчеников.Зарегистрирован) = True)) GROUP BY ДанныеУчеников.Логин, ДанныеУчеников.Пароль HAVING(((ДанныеУчеников.Логин) =\"" + login + "\") AND((ДанныеУчеников.Пароль) =\"" + password + "\"));";

OleDbCommand command = new OleDbCommand(query, myConnection);

string answer;

try

{

answer = command.ExecuteScalar().ToString();

}

catch

{

answer = "0";

}

if (answer != "0") return true;

query = "SELECT DISTINCTROW Count(Преподаватели.Зарегистрирован) AS [Count-Зарегистрирован] FROM Преподаватели WHERE(((Преподаватели.Зарегистрирован) = True)) GROUP BY Преподаватели.Логин, Преподаватели.Пароль HAVING(((Преподаватели.Логин) =\"" + login + "\") AND((Преподаватели.Пароль) =\"" + password + "\"));";

command = new OleDbCommand(query, myConnection);

try

{

answer = command.ExecuteScalar().ToString();

}

catch

{

answer = "0";

}

if (answer != "0") return true;

else return false;

}

public void Registr()

{

string query = "UPDATE ДанныеУчеников SET ДанныеУчеников.Зарегистрирован = True, ДанныеУчеников.Логин = \"" + LoginBox.Text + "\", ДанныеУчеников.Пароль = \"" + PasswordBox.Password + "\", ДанныеУчеников.Телефон = \"" + PhoneBox.Text + "\", ДанныеУчеников.Почта = \"" + MailBox.Text + "\" WHERE(((ДанныеУчеников.Имя) = \"" + NameBox.Text + "\") AND((ДанныеУчеников.Фамилия) = \"" + SurnameBox.Text + "\") AND((ДанныеУчеников.Отчество) = \"" + PatronymicBox.Text + "\"));";

if (teacher) query = "UPDATE Преподаватели SET Преподаватели.Зарегистрирован = True, Преподаватели.Логин = \"" + LoginBox.Text + "\", Преподаватели.Пароль = \"" + PasswordBox.Password + "\", Преподаватели.Телефон = \"" + PhoneBox.Text + "\", Преподаватели.Почта = \"" + MailBox.Text + "\" WHERE(((Преподаватели.Имя) = \"" + NameBox.Text + "\") AND((Преподаватели.Фамилия) = \"" + SurnameBox.Text + "\") AND((Преподаватели.Отчество) = \"" + PatronymicBox.Text + "\"));";

OleDbCommand command = new OleDbCommand(query, myConnection);

command.ExecuteScalar();

MessageBox.Show("Регистрация прошла успешно!", "!!!Поздравляю!!!");

}

private void Window\_Closed(object sender, EventArgs e)

{

// заркываем соединение с БД

myConnection.Close();

}

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

{

this.Close();

}

private void Rectangle\_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

this.DragMove();

}

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

{

if (NameBox.Text.Length == 0 || SurnameBox.Text.Length == 0 || PatronymicBox.Text.Length == 0 || PhoneBox.Text.Length == 0 || MailBox.Text.Length == 0)

{

MessageBox.Show("Вам нужно заполнить все поля!", "!!!Внимание!!!");

return;

}

if (PasswordBox.Password.Length < 5)

{

MessageBox.Show("Пароль должен содержать минимум 5 символов!", "!!!Внимание!!!");

return;

}

if (PasswordBox.Password != PasswordBox1.Password)

{

MessageBox.Show("Пароли не совпадают!", "!!!Внимание!!!");

return;

}

bool checkData = RegistrCheck(NameBox.Text, SurnameBox.Text, PatronymicBox.Text);

if (!checkData)

{

MessageBox.Show("Человека с таким ФИО нет в списке, либо он уже зарегистрирован!", "!!!Внимание!!!");

return;

}

checkData = LoginCheck(LoginBox.Text, PasswordBox.Password);

if (checkData)

{

MessageBox.Show("Логин занят!", "!!!Внимание!!!");

return;

}

Registr();

MainWindow Window = new MainWindow();

Window.Show();

this.Close();

}

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

{

MainWindow Window = new MainWindow();

Window.Show();

this.Close();

}

}

}

## Приложение 8. Файл WorkWindow.xaml.cs

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.Shapes;

using System.Data.OleDb;

using System.IO;

using System.Drawing;

using System.Windows.Forms;

using System.ComponentModel;

using System.Drawing.Printing;

namespace StartWindow

{

/// <summary>

/// Логика взаимодействия для WorkWindow.xaml

/// </summary>

public partial class WorkWindow : Window

{

public OleDbConnection myConnection;

public static string connectString = "provider=Microsoft.ACE.Oledb.12.0;Data Source= D:\\AducationBase.accdb;";

public WorkWindow()

{

InitializeComponent();

myConnection = new OleDbConnection(connectString);

myConnection.Open();

HalloBox.Text += StartWindow.Data.TeacherUsr.Name + " " + StartWindow.Data.TeacherUsr.Patronymic;

}

private void Window\_Closed(object sender, EventArgs e)

{

myConnection.Close();

}

private void HideAllGrids()

{

GiveTaskGrid.Visibility = Visibility.Hidden;

TasksViewGrid.Visibility = Visibility.Hidden;

AnswerGrid.Visibility = Visibility.Hidden;

MarkGrid.Visibility = Visibility.Hidden;

JournalGrid.Visibility = Visibility.Hidden;

}

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

{

HideAllGrids();

GiveTaskGrid.Visibility = Visibility.Visible;

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

HideAllGrids();

UsrIDBuff.Text = "1";

TasksViewGrid.Visibility = Visibility.Visible;

}

private void SandTask(object sender, RoutedEventArgs e)

{

if (KlassComboBox.Text!="" && SubjectComboBox.Text!= "" && DatePic.Text!="" && TitleBox.Text!="" && TaskText.Text!="")

{

int idKlass, idSubject, idTeacher, idSchool;

idKlass = Data.TeacherUsr.GetIDWithText("Учебная программа", "Класс", KlassComboBox.Text);

idSubject = Data.TeacherUsr.GetIDWithText("Список предметов", "Предмет", SubjectComboBox .Text);

idTeacher = Data.TeacherUsr.ID;

string DateBegin = Convert.ToString(DateTime.Today.ToString("dd.MM.yyyy"));

string DateEnd = DatePic.Text;

idSchool = Data.TeacherUsr.SchoolID;

string title = TitleBox.Text;

string task = TaskText.Text;

if (Data.NewTask.TaskFiles.Count > 0)

{

OleDbConnection connection = new OleDbConnection(connectString);

string query = "INSERT INTO Задания(Тема, [Дата выдачи], [Дата сдачи], Школа, Предмет, [Текст задания], Класс, Преподаватель, ДопФайлы) " +

"values( '" + title + "', '" + DateBegin + "', '" + DateEnd + "', '" + idSchool + "', '" + idSubject + "', '" + task + "', '" + idKlass + "', '" + idTeacher + "' , ? )";

OleDbCommand command = new OleDbCommand(query, connection);

OleDbParameter oleDbParameter = new OleDbParameter("ДопФайлы", OleDbType.VarBinary);

string fileName = StartWindow.Data.NewTask.TaskFiles[0].directory; //Путь к файлу

System.Drawing.Image image = System.Drawing.Image.FromFile(fileName); //Изображение из файла.

MemoryStream memoryStream = new MemoryStream(); //Поток в который запишем изображение

image.Save(memoryStream, System.Drawing.Imaging.ImageFormat.Bmp); //Сохраняем изображение в поток.

oleDbParameter.Value = memoryStream.ToArray(); //Устанавливаем значение параметра

command.Parameters.Add(oleDbParameter);

connection.Open();

command.ExecuteNonQuery();

connection.Close();

memoryStream.Dispose();

}

else

{

OleDbConnection connection = new OleDbConnection(connectString);

string query = "INSERT INTO Задания(Тема, [Дата выдачи], [Дата сдачи], Школа, Предмет, [Текст задания], Класс, Преподаватель) " +

"values( '" + title + "', '" + DateBegin + "', '" + DateEnd + "', '" + idSchool + "', '" + idSubject + "', '" + task + "', '" + idKlass + "', '" + idTeacher + "')";

OleDbCommand command = new OleDbCommand(query, connection);

connection.Open();

command.ExecuteNonQuery();

connection.Close();

}

KlassComboBox.Text = "";

SubjectComboBox.Text = "" ;

DatePic.Text = "" ;

TitleBox.Text = "" ;

TaskText.Text = "";

Data.NewTask.TaskFiles.Clear();

TaskViewList.ItemsSource = ViewModel.TasksViewModel.RefrashItems();

(TaskViewList.ItemsSource as ICollectionView).Refresh();

System.Windows.MessageBox.Show("Задание успешно выдано!", "Поздравляем!");

}

}

private void AddFileGialog(object sender, RoutedEventArgs e)

{

OpenFileDialog myDialog = new OpenFileDialog();

myDialog.Filter = "Картинки(\*.JPG;\*.PNG)|\*.JPG;\*.PNG" + "|Все файлы (\*.\*)|\*.\* ";

myDialog.CheckFileExists = true;

myDialog.Multiselect = true;

myDialog.ShowDialog();

string buff = myDialog.FileName;

if (buff != "")

{

StartWindow.Data.TaskFileImage buffImage = new Data.TaskFileImage(buff);

StartWindow.Data.NewTask.TaskFiles.Add(buffImage);

StartWindow.ViewModel.TasksViewModel.TaskFilesViewBuff.Refresh();

}

}

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

{

SubjectComboBox1.Text = "";

KlassComboBox1.Text = "";

}

private void ViewImageFromBuff(object sender, RoutedEventArgs e)

{

System.Windows.Controls.Image b = sender as System.Windows.Controls.Image;

if (b != null)

{

ImageForView.Source = b.Source;

ImageViewGrid.Visibility = Visibility.Visible;

}

}

private void ImageForView\_MouseDown(object sender, MouseButtonEventArgs e)

{

ImageViewGrid.Visibility = Visibility.Hidden;

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

HideAllGrids();

MarkGrid.Visibility = Visibility.Visible;

AnswerGrid.Visibility = Visibility.Visible;

}

private void MarkAnswer(object sender, RoutedEventArgs e)

{

Data.Tasks buff = (sender as System.Windows.Controls.Button).DataContext as Data.Tasks;

if (buff != null)

{

CurrentTaskView.ItemsSource = ViewModel.TasksViewModel.RefrashCurrentTask(buff);

(CurrentTaskView.ItemsSource as ICollectionView).Refresh();

CurrentTaskAnswerView.ItemsSource = ViewModel.TasksViewModel.RefrashCurrentTaskAnswers();

(CurrentTaskAnswerView.ItemsSource as ICollectionView).Refresh();

HideAllGrids();

AnswerGrid.Visibility = Visibility.Visible;

MarkGrid.Visibility = Visibility.Visible;

}

}

private void MarkTheAnswer(object sender, RoutedEventArgs e)

{

if (MarkComboBox.Text!="")

{

((sender as System.Windows.Controls.Button).DataContext as Data.Answer).MarkTheAnswer(MarkComboBox.Text, CommentBox.Text);

(CurrentTaskAnswerView.ItemsSource as ICollectionView).Refresh();

}

else

{

System.Windows.MessageBox.Show("Оцените ответ!");

}

}

private void LoadJournal(object sender, RoutedEventArgs e)

{

if (JKlassComboBox.Text == "" || JSubjectComboBox.Text == "")

{

System.Windows.MessageBox.Show("Вам нужно выбрать класс и предмет для просмотра!");

return;

}

int idKlass = Data.TeacherUsr.GetIDWithText("Учебная программа", "Класс", JKlassComboBox.Text);

int idSubject = Data.TeacherUsr.GetIDWithText("Список предметов", "Предмет", JSubjectComboBox.Text);

OleDbConnection connection = new OleDbConnection(connectString);

string query = "TRANSFORM First(Ответы.Оценка) AS [First-Оценка] " +

"SELECT[ДанныеУчеников]![Имя] + \" \" +[ДанныеУчеников]![Отчество] + \" \" +[ДанныеУчеников]![Фамилия] AS ФИО " +

"FROM Задания INNER JOIN(ДанныеУчеников INNER JOIN Ответы ON ДанныеУчеников.ID = Ответы.УченикID) ON Задания.ID = Ответы.ЗаданиеID " +

"WHERE(((ДанныеУчеников.Класс) = " + idKlass + ") AND((Задания.Предмет) = " + idSubject +")) " +

"GROUP BY[ДанныеУчеников]![Имя] + \" \" +[ДанныеУчеников]![Отчество] + \" \" +[ДанныеУчеников]![Фамилия], ДанныеУчеников.Класс, Задания.Предмет " +

"PIVOT Задания.Тема;";

OleDbCommand Cmd = new OleDbCommand();

Cmd.Connection = connection;

Cmd.CommandText = query;

OleDbDataAdapter DataAdap = new OleDbDataAdapter(Cmd);

System.Data.DataTable dtRecord = new System.Data.DataTable();

DataAdap.Fill(dtRecord);

JournalDataGrid.ItemsSource = dtRecord.DefaultView;

PrintButton.IsEnabled = true;

}

private void Button\_Click\_3(object sender, RoutedEventArgs e)

{

HideAllGrids();

JournalGrid.Visibility = Visibility.Visible;

}

private void PrintJ(object sender, RoutedEventArgs e)

{

System.Windows.Controls.PrintDialog printDialog = new System.Windows.Controls.PrintDialog();

if (printDialog.ShowDialog() == true)

{

printDialog.PrintVisual(JournalDataGrid, "Распечатка журнала");

}

}

}

}

## Приложение 9. Файл WorkWindowStudent.xaml.cs

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.Shapes;

using System.Data.OleDb;

using System.IO;

using System.Drawing;

using System.Windows.Forms;

using System.ComponentModel;

namespace StartWindow

{

/// <summary>

/// Логика взаимодействия для WorkWindow.xaml

/// </summary>

public partial class WorkWindowStudent : Window

{

public OleDbConnection myConnection;

public static string connectString = "provider=Microsoft.ACE.Oledb.12.0;Data Source= D:\\AducationBase.accdb;";

public WorkWindowStudent()

{

InitializeComponent();

myConnection = new OleDbConnection(connectString);

myConnection.Open();

HalloBox.Text += StartWindow.Data.StudentUsr.Name + " " + StartWindow.Data.StudentUsr.Patronymic;

}

private void Window\_Closed(object sender, EventArgs e)

{

myConnection.Close();

}

private void HideAllGrids()

{

AnswerGrid.Visibility = Visibility.Hidden;

TasksViewGrid.Visibility = Visibility.Hidden;

JournalGrid.Visibility = Visibility.Hidden;

}

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

{

HideAllGrids();

AnswerGrid.Visibility = Visibility.Visible;

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

HideAllGrids();

UsrIDBuff.Text = "1";

TasksViewGrid.Visibility = Visibility.Visible;

}

private void SandTask(object sender, RoutedEventArgs e)

{

Data.NewTaskAnswer.AnswerText = AnswerText.Text;

if (Data.NewTaskAnswer.UnloadAnswer())

{

System.Windows.MessageBox.Show("Ответ успешно отправлен!", "Поздравляем!");

}

else

{

System.Windows.MessageBox.Show("Заполните поле ответа!");

}

}

private void AddFileGialog(object sender, RoutedEventArgs e)

{

OpenFileDialog myDialog = new OpenFileDialog();

myDialog.Filter = "Картинки(\*.JPG;\*.PNG)|\*.JPG;\*.PNG" + "|Все файлы (\*.\*)|\*.\* ";

myDialog.CheckFileExists = true;

myDialog.Multiselect = true;

myDialog.ShowDialog();

string buff = myDialog.FileName;

if (buff != "")

{

StartWindow.Data.TaskFileImage buffImage = new Data.TaskFileImage(buff);

StartWindow.Data.NewTaskAnswer.AnswerFiles.Add(buffImage);

StartWindow.ViewModel.TasksViewModelStudent.TaskFilesViewBuff.Refresh();

}

}

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

{

SubjectComboBox1.Text = "";

}

private void ViewImageFromBuff(object sender, RoutedEventArgs e)

{

System.Windows.Controls.Image b = sender as System.Windows.Controls.Image;

if (b != null)

{

ImageForView.Source = b.Source;

ImageViewGrid.Visibility = Visibility.Visible;

}

}

private void ImageForView\_MouseDown(object sender, MouseButtonEventArgs e)

{

ImageViewGrid.Visibility = Visibility.Hidden;

}

private void AnswerTheTask(object sender, RoutedEventArgs e)

{

System.Windows.Controls.Button b = sender as System.Windows.Controls.Button;

Data.Tasks data = b.DataContext as Data.Tasks;

HideAllGrids();

Data.NewTaskAnswer.StudentID = Data.StudentUsr.ID;

Data.NewTaskAnswer.TaskID = data.ID;

Data.NewTaskAnswer.TeacherID = data.TeacherID;

CurrentDateBeginStr.Text = data.DateBeginStr;

CurrentDateEndStr.Text = data.DateEndStr;

CurrentIMGS.Source = data.IMGS;

CurrentSubject.Text = data.Subject;

CurrentTaskText.Text = data.TaskText;

CurrentTitle.Text = data.Title;

CurrentLine.Visibility = Visibility.Visible;

AnswerGrid.Visibility = Visibility.Visible;

}

private void LoadJournal(object sender, RoutedEventArgs e)

{

if ( JSubjectComboBox.Text == "")

{

System.Windows.MessageBox.Show("Вам нужно выбрать предмет для просмотра!");

return;

}

int idSubject = Data.TeacherUsr.GetIDWithText("Список предметов", "Предмет", JSubjectComboBox.Text);

OleDbConnection connection = new OleDbConnection(connectString);

string query = "TRANSFORM First(Ответы.Оценка) AS [First-Оценка] " +

"SELECT[ДанныеУчеников]![Имя] + \" \" +[ДанныеУчеников]![Отчество] + \" \" +[ДанныеУчеников]![Фамилия] AS ФИО " +

"FROM Задания INNER JOIN(ДанныеУчеников INNER JOIN Ответы ON ДанныеУчеников.ID = Ответы.УченикID) ON Задания.ID = Ответы.ЗаданиеID " +

"WHERE(((ДанныеУчеников.Класс) = " + Data.StudentUsr.KlassID + ") AND((Задания.Предмет) = " + idSubject +")) " +

"GROUP BY[ДанныеУчеников]![Имя] + \" \" +[ДанныеУчеников]![Отчество] + \" \" +[ДанныеУчеников]![Фамилия], ДанныеУчеников.Класс, Задания.Предмет " +

"PIVOT Задания.Тема;";

OleDbCommand Cmd = new OleDbCommand();

Cmd.Connection = connection;

Cmd.CommandText = query;

OleDbDataAdapter DataAdap = new OleDbDataAdapter(Cmd);

System.Data.DataTable dtRecord = new System.Data.DataTable();

DataAdap.Fill(dtRecord);

JournalDataGrid.ItemsSource = dtRecord.DefaultView;

PrintButton.IsEnabled = true;

}

private void Button\_Click\_3(object sender, RoutedEventArgs e)

{

HideAllGrids();

JournalGrid.Visibility = Visibility.Visible;

}

private void PrintJ(object sender, RoutedEventArgs e)

{

System.Windows.Controls.PrintDialog printDialog = new System.Windows.Controls.PrintDialog();

if (printDialog.ShowDialog() == true)

{

printDialog.PrintVisual(JournalDataGrid, "Распечатка журнала");

}

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

HideAllGrids();

JournalGrid.Visibility = Visibility.Visible;

}

}

}