### Обособленное структурное подразделение

### «Индустриальный техникум»

### Государственного образовательного учреждения

### высшего образования Луганской Народной Республики

### «Донбасский государственный технический институт»

### Методическая комиссия информатики и компьютерной техники

ВЫПУСКНая КВАЛИФИКАЦИОННая РАБОТа (ДИПЛОМНая работа)

### специалиста среднего звена

на тему: «Разработка информационной системы по формированию портфолио студента учебного заведения»

Выполнил студент 4 курса

группы ПКС-2018-1/9

специальности 09.02.03 «Программирование в компьютерных системах»

Раевский Алексей Александрович

Руководитель \_\_\_\_\_\_\_\_ В. А. Селезнев

Председатель методической

комиссии \_\_\_\_\_\_\_\_ О.Ю. Ленкова

Рецензент \_\_\_\_\_\_\_\_ Е. Ю.Позднякова

г. Алчевск – 2022 год

Обособленное структурное подразделение «Индустриальный техникум»

Государственного образовательного учреждения высшего образования

Луганской Народной Республики «Донбасский государственный технический институт»

Отделение Компьютерных и электрических систем

Методическая комиссия Информатики и компьютерной техники

Специальность 09.02.03 «Программирование в компьютерных системах»

**УТВЕРЖДАЮ**

врио заместителя директора по УПР

\_\_\_\_\_\_\_\_\_ С.И. Васильченко

«\_\_\_\_»\_\_\_\_\_\_\_\_\_ 2022 года

**ЗАДАНИЕ**

**ДЛЯ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ (ДИПЛОМНой работы)**

СТУДЕНТУ

Раевскому Алексею Александровичу

1. Тема работы «Разработка информационной системы по формированию портфолио студента учебного заведения.», утвержденная Приказом от 05.04.2022 года №31-ст.

Руководитель работы преподаватель высшей категории Селезнев В.А.

2 Срок предоставления студентом проекта 16.06.2022 года.

3. Исходные данные к работе: база данных по группам и студентам.

4. Содержание пояснительной записки (перечень вопросов, которые нужно разработать)

Введение

1 Обзор систем автоматизации работы тестирующей программы

1.1 Описание предметной области

1.2 Постановка задачи

1.3 Характеристика программных аналогов

1.4 Выбор программного обеспечения и языков разработки

2 Проектирование и реализация модулей систем автоматизации тестирующей программы

2.1 Проектирование базы данных

2.2 Описание программного обеспечения

3 Тестирование

Заключение

Список литературы

Приложение А – Техническое задание

Приложение Б – Листинг программы

5. Дата выдачи задания 20.04.2022 года.

**КАЛЕНДАРНЫЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Название этапов дипломного проекта | Срок выполнения этапов проекта | Примечание |
| 1 | Обзор систем автоматизации работы тестирующей программы | 20.04.22 |  |
| 1.1 | Описание предметной области | 24.04.22 |  |
| 1.2 | Постановка задачи | 28.04.22 |  |
| 1.3 | Характеристика программных аналогов | 30.04.22 |  |
| 1.4 | Выбор программного обеспечения и языков разработки | 5.05.22 |  |
| 2 | Проектирование и реализация модулей систем автоматизации тестирующей программы | 10.05.22 |  |
| 2.1 | Проектирование базы данных | 15.05.22 |  |
| 2.2 | Описание программного обеспечения | 20.05.22 |  |
| 3 | Тестирование | 25.05.22 |  |
|  | Заключение | 30.05.22 |  |
|  | Список литературы | 4.06.22 |  |
|  | Приложение А – Техническое задание | 5.06.22 |  |
|  | Приложение Б – Листинг программы | 10.06.21 |  |

Студент \_\_\_\_\_\_\_\_\_\_\_\_ Р.А. Алексей

Руководитель проекта \_\_\_\_\_\_\_\_\_\_\_\_ В.А. Селезнев

Согласовано на заседании цикловой комиссии

информатики и компьютерной техники

Протокол № 3 от 14.03.2022

Председатель комиссии \_\_\_\_\_\_\_\_\_\_\_\_ О.Ю.Ленкова

*Содержание*

Изм.

Лист

№ докум.

Подпись

Дата

Лист

*3*

*ДР 09.02.03.718.00.00*

Разраб.

*Раевский А.А.*

Провер.

*Селезнев В.А.*

Реценз.

Позднякова Е.Ю.

. Соболев

Н. Контр.

Утв.

*«Разработка информационной системы по формированию портфолио студента учебного заведения»*

*Пояснительная записка*

Лит.

Листов

*155*

*ИТ ДонГТИ группа ПКС-18-1/9*

|  |  |
| --- | --- |
| *Задание выпускной квалификационной работы (дипломной работы)* | *3* |
| *Введение* | *4* |
| *1 Обзор систем автоматизации работы тестирующей программы* | *6* |
| *1.1 Описание предметной области* | *6* |
| *1.2 Постановка задачи* | *8* |
| *1.3 Характеристика программных аналогов* | *10* |
| *1.4 Выбор программного обеспечения и языков разработки* | *13* |
| *2 Проектирование и реализация модулей систем автоматизации тестирующей программы* | *26* |
| *2.1 Проектирование базы данных* | *26* |
| *2.2 Описание программного обеспечения* | *45* |
| *3 Тестирование* | *63* |
| *Заключение* | *70* |
| *Список литературы* | *71* |
| *Приложение А – Техническое задание* |  |
| *Приложение Б – Листинг программы* |  |

*Введение*

*Информационные системы не так давно стали частью нашей жизни, но уже является неотъемлемой её частью. Сложно представить предприятие, офис, магазин и даже домашний компьютер, на котором в том или ином виде не встретишь информационную систему.*

*Информационная система – система, предназначенная для хранения, поиска и обработки информации, включающая соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.).*

*Основным преимуществом автоматизированной информационной системы для организации является возможность правильно составить декларацию, провести расчеты и, в конечном итоге, сформировать отчет, который наглядно демонстрирует информацию интересующую пользователя.*

*При работе с личной информацией человек часто сталкивается с проблемой заполнения большого количества однотипной информации в документе. Это может быть, как и простое указание даты так и дублирование целы выписок справок распоряжений. В это число входит такой документ как портфолио. Портфолио служит для организации данных включающих личные достижения человека. Не важно работает ли он на предприятии или учится в учебном заведении.*

*Принимая во внимание актуальность и востребованность создания информационной системы, принято решение о ее разработке.*

*Объектом исследования, является учебное заведение ОСП «ИТ» ГОУ ВО ЛНР «ДонГТИ».*

*Предметом исследования является автоматизация формирования документа «Портфолио», включающего в себя данные студента и группы.*

*Задачи проекта состоят в: описании предметной области, разработке базы данных.*

*Для достижения указанной цели следующие задачи:*

*– Изучение особенностей предоставленного образца документа, который в последствии должен будет формироваться;*

*– Сравнительный анализ существующих информационных связанных с формирование документа;*

*– Разработка требований к проектируемой информационной системе;*

*– Тестирование Базы данных;*

*Разработанная информационная система позволит значительно сократить трудозатраты сотрудников учебного заведения.*

*1 Обзор систем автоматизации работы тестирующей программы*

* 1. *Описание предметной области*

*В ходе работы на предприятии, обучении в учебном учреждении, ведении бизнеса, содержании собственных документов, человек сталкивается с созданием, редактированием, формировании новых документов. Данный процесс позволяет структурировать данные, получить быстрый и удобный доступ к ним, а также упрощается взаимодействие между людьми по вопросам обмена какой–либо информации, касающейся как возможно и лично человека, так и сферы его жизни, активности, работы.*

*В процессе работы в учебном заведении бухгалтера, преподаватели, сталкиваются с огромного количества документов, что значительно усложняет работу в этой сфере. Портфолио является важной частью документации на студента учебного заведения. Этот документ позволяет получить информацию касающеюся студента в удобном виде.*

*Портфолио должно включать:*

*­­– контактную информации студента;*

*– место предыдущего образования;*

*– список языков и уровень владения ими;*

*– приказы, полученные в адрес данного студента;*

*– темы курсовых и практик. Место проведения практики;*

*– данные внеаудиторной, научной деятельности студента;*

*С заполнением данного документа может не возникнуть проблем в случае формирования его для одного студента. Недопонимания, ошибки и тому прочие возникают вследствие формирования документа на целую группу или даже на целую специальность, даже с учётом того что данный документ имеет чёткую структуру она может быть нарушена, а изменение шаблона заполнения чревато огромными затратами во времени, потраченном на редактировании ранее сформированного портфолио, так как данный документ может заполняться в период до четырёх лет и дополнятся по мере обучения студента.*

*Информационная система должна включать в себя выше перечисленную информацию, а также связь данной информации между собой для соблюдения целостности данных.*

*1.2 Постановка задачи*

*Для учебных заведения, очень важно оперативно и своевременно вести учёт данных о студентах. Поэтому задача, автоматизации обработки данных студента, является одной из приоритетных:*

*Задачи, стоящие перед информационной системой портфолио, структурирование данных, а также формирования отчёта, именуемого «Портфолио».*

*Данные касающиеся группы это:*

*– содержание, дата и номер приказа на перевод;*

*– дисциплины, нагрузка и форма аттестации промежуточного контроля;*

*– дисциплина и семестр курсовых работ;*

*– вид, должность и сроки прохождения практик;*

*Данные касающиеся студента это:*

*– ФИО и личная информация;*

*– список языков и уровень владения;*

*– номер, дата и содержание приказа на: перерыв (в академическом отпуске), поощрение, взыскание;*

*– сведения о предыдущем образовании;*

*– оценки по промежуточному контролю;*

*– тема и оценка по курсовой работе;*

*– база и оценка по практикам;*

*– тема и оценка по выпускной квалификационной работе;*

*– сведенья о деятельности вне учебного заведения;*

*Данные группы должны заполняться в отрыве от данных самого студента, что позволит упростить заполнение отчёта и предотвратить ошибки в заполнении дублирующийся информации такой как: наименование дисциплин, даты проведения практик, курсовых и так далее.*

*Ведение данных в таком виде позволит в итоге должно позволить разово заполнять данные о группе или специальности и в дальнейшем обращать внимание лишь на данные касающиеся студента, не боясь нарушить целостность или допустить ошибку в формировании отчёта.*

*Так же данный способ ведения информации позволит быстро вносить изменения в уже существующие данные.*

*1.3 Характеристика программных аналогов*

*«1С: Бухгалтерия 8»*

*Программный продукт «1С: Бухгалтерия 8» включает технологическую платформу «1С: Предприятие 8» и конфигурацию (прикладное решение) «Бухгалтерия предприятия».*

*Конфигурация предназначена для автоматизации бухгалтерского и налогового учета, включая подготовку обязательной (регламентированной) отчетности в коммерческой организации, применяющей план счетов бухгалтерского учета.*

*Конфигурация обеспечивает решение всех задач бухгалтерской службы предприятия, если бухгалтерская служба полностью отвечает за учет на предприятии, включая, например, выписку первичных документов, учет продаж и т. д. Данное прикладное решение также можно использовать только для ведения бухгалтерского и налогового учета.*

*«Stimulsoft Reports.Web»*

*Stimulsoft Reports.Web предназначен для интеграции генератора отчётов в Web проекты, построенные с использованием ASP.NET, ASP.NET MVC и .NET Core платформ. Дополнительно, в состав продукта входит кроссплатформенный набор инструментов для работы совместно с Angular фреймворком. Доступны события для контролирования всех этапов работы с отчётом – от загрузки и подключения данных, до его просмотра и печати. Помимо широкого применения визуальных компонентов, любые действия над отчётом могут быть выполнены при помощи кода на стороне сервера.*

*1.4 Выбор программного обеспечения и языков разработки*

*В разработке приложения участвуют множество программ и программного обеспечения, но в основе информационной системы основным всё же остаётся среда разработки и СУБД. В случае создания данной информационной системы использовался следующий инструментарий:*

*– система управления базой данных SQL Server;*

*– среда разработки Visual Studio;*

*Система управления базами данных Microsoft SQL Server*

*Microsoft SQL Server – система управления реляционными базами данных, разработанная корпорацией Microsoft. Обычно используется для работы с базами данных большого размера. Лицензирование осуществляется на платной основе.*

*SQL Server – это реляционная СУБД, которая использует язык Transact-SQL для пересылки сообщений между компьютером клиента и сервером СУБД.*

*SQL Server является одной из наиболее популярных систем управления базами данных (СУБД) в мире. Данная СУБД подходит для самых различных проектов: от небольших приложений до больших высоконагруженных проектов.*

*SQL Server характеризуется такими особенностями как:*

*– производительность – SQL Server работает очень быстро;*

*– надежность и безопасность – SQL Server предоставляет шифрование данных;*

*– простота – с данной СУБД относительно легко работать и вести администрирование.*

*Реляционная СУБД отвечает за поддержку структуры базы данных и решает следующие задачи:*

*– поддерживает связи между данными в базе;*

*– гарантирует корректное хранение данных и выполнение правил, которые регламентируют связи между ними;*

*– восстанавливает данные после аварии системы, переводя их в согласованное состояние, зафиксированное до сбоя.*

*Центральным аспектом в Microsoft SQL Server, как и в любой СУБД, является база данных. База данных представляет хранилище данных, организованных определенным способом. Нередко физически база данных представляет файл на жестком диске, хотя такое соответствие необязательно. Для хранения и администрирования баз данных применяются системы управления базами данных (database management system) или СУБД (DBMS). Microsoft SQL Server является одной из такой СУБД.*

*Для организации баз данных Microsoft SQL Server использует реляционную модель. Эта модель баз данных была разработана еще в 1970 году Эдгаром Коддом. А на сегодняшний день она фактически является стандартом для организации баз данных.*

*Реляционная модель предполагает хранение данных в виде таблиц, каждая из которых состоит из строк и столбцов. Каждая строка хранит отдельный объект, а в столбцах размещаются атрибуты этого объекта.*

*Для идентификации каждой строки в рамках таблицы применяется первичный ключ (primary key). В качестве первичного ключа может выступать один или несколько столбцов. Используя первичный ключ, мы можем ссылаться на определенную строку в таблице. Соответственно две строки не могут иметь один и тот же первичный ключ.*

*Через ключи одна таблица может быть связана с другой, то есть между двумя таблицами могут быть организованы связи. А сама таблица может быть представлена в виде отношения («relation»).*

*Для взаимодействия с базой данных применяется язык SQL (Structured Query Language). Клиент (например, внешняя программа) отправляет запрос на языке SQL посредством специального API. СУБД должным образом интерпретирует и выполняет запрос, а затем посылает клиенту результат выполнения.*

*Изначально язык SQL был разработан в компании IBM для системы баз данных, которая называлась System/R. При этом сам язык назывался SEQUEL (Structured English Query Language). Хотя в итоге ни база данных, ни сам язык не были впоследствии официально опубликованы, по традиции сам термин SQL нередко произносят как «сиквел».*

*В 1979 году компания Relational Software Inc. разработала первую систему управления баз данных, которая называлась Oracle и которая использовала язык SQL. В связи с успехом данного продукта компания была переименована в Oracle.*

*Впоследствии стали появляться другие системы баз данных, которые использовали SQL. В итоге в 1989 году Американский Национальный Институт Стандартов (ANSI) кодифицировал язык и опубликовал его первый стандарт. После этого стандарт периодически обновлялся и дополнялся. Последнее его обновление состоялось в 2011 году. Но, несмотря на наличие стандарта, нередко производители СУБД используют свои собственные реализации языка SQL, которые немного отличаются друг от друга.*

*Выделяются две разновидности языка SQL: PL-SQL и T-SQL. PL-SQL используется в таких СУБД как Oracle и MySQL. T-SQL (Transact-SQL) применяется в SQL Server.*

*В зависимости от задачи, которую выполняет команда T-SQL, он может принадлежать к одному из описанных ниже типов.*

*DDL (Data Definition Language – Язык определения данных). К этому типу относятся различные команды, которые создают базу данных, таблицы, индексы, хранимые процедуры и т.д. В частности, к этому типу можно отнести следующие команды:*

*– CREATE: создает объекты базы данных (саму базу данных, таблицы, индексы и т.д.);*

*– ALTER: изменяет объекты базы данных;*

*– DROP: удаляет объекты базы данных;*

*– TRUNCATE: удаляет все данные из таблиц.*

*DML (Data Manipulation Language – Язык манипуляции данными). К этому типу относят команды на выборку данных, их обновление, добавление, удаление – команды управления данными. К этому типу относятся следующие команды:*

*– SELECT: извлекает данные из БД;*

*– UPDATE: обновляет данные;*

*– INSERT: добавляет новые данные;*

*– DELETE: удаляет данные.*

*DCL (Data Control Language – Язык управления доступа к данным). К этому типу относят команды, которые управляют правами доступа к данным. В частности, это следующие команды:*

*– GRANT: предоставляет права для доступа к данным;*

*– REVOKE: отзывает права на доступ к данным.*

*SQL Server имеет ряд возможностей:*

*– легкость установки, настройки и развертывания. SQL Server предоставляет администраторам баз данных полный набор инструментов, необходимых для тонкой настройки, как в составе промышленных систем, так и небольших однопользовательских системах. Установка или обновление SQL Server происходит под управлением стандартного графического приложения инсталляции программ;*

*– SQL Server автоматически и динамически меняет конфигурацию в процессе работы. По мере роста числа пользователей, подключенных к SQL Server, он может динамически выделять необходимые ресурсы, например память. При снижении загруженности SQL Server освобождает ресурсы и возвращает их системе. Если на сервере одновременно запускаются другие приложения, SQL Server выявит выделение для них дополнительной виртуальной памяти и уменьшит объем используемой им виртуальной памяти. SQL Server также способен автоматически увеличивать или уменьшать размер базы данных по мере добавления или удаления информации;*

*– некоторые функции SQL Server увеличивают масштабируемость системы, SQL Server способен планировать параллельное выполнение, при котором обработка оператора SQL разделается на несколько частей. Каждая часть может быть выполнена на отдельном процессоре, в этом случае формирования полного результирующего набора осуществляется быстрее, чем в том случае, когда отдельные части операторов выполняются последовательно;*

*– SQL Server поддерживает аутентификацию Windows, что позволяет применять в качестве учетных записей SQL Server пользовательские и доменные учетные записи Windows. Если пользователь вошел в систему Windows, введя логин и пароль, то эти же данные используются и для подключения к SQL Server. Таким образом SQL Server сам не выполняет проверку пользователей, а пользователям не нужны отдельные имена и пароли для подключения к системе SQL Server.*

*Доступны разные редакции SQL Server, способные удовлетворить требования заказчиков (организаций и отдельных лиц) к производительности и стоимости системы:*

*– Express Edition. Облегченная бесплатная версия SQL Server для обучения и создания простых приложений;*

*– Workgroup Edition. Версия разработана для малого и среднего бизнеса, также может быть использована на уровне отделов. Эта редакция оказывает поддержку реляционных баз данных без бизнес-аналитики и средств высокой готовности. Эта редакция поддерживает до двух процессоров и максимум 2 Гбайт RAM;*

*– Standard Edition. Предоставляет 32-разрядную версию и 64-разрядную поддержку для систем х64 и Itanium. Также поддерживает до четырех процессоров и включает полный ряд функциональности бизнес-аналитики;*

*– Enterprise Edition. Специальная форма системы SQL Server, которая предназначена для приложений, критических к времени выполнения и с огромным количеством пользователей. В отличии от Standard Edition эта редакция содержит дополнительные возможности, которые могут быть полезными для очень мощных инсталляций на оборудовании с симметричными мультипроцессорами или кластерами;*

*– Developer Edition. Дает возможность разработчикам создавать и тестировать любые типы приложений с SQL Server на 32- и 64-разядных плаформах. Включает всю функциональность Enterprise Edition, но версия лицензирована только для использования в разработке, тестировании и демонстрации. Каждая лицензия на Developer Edition предоставляет право одному разработчику использовать эту систему на таком количестве систем, сколько ему необходимо. Для быстрого перевода в производственную эксплуатацию система базы данных Developer Edition может быть дегко обновлена до Enterprise Edition;*

*– Compact Edition. Скромная версия SQL Server, которую вы можете использовать в карманных PC, мобильных телефонах, планшетных и настольных компьютерах.*

*Среда разработки Visual Studio.*

*Microsoft Visual Studio – линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств.*

*Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense (технология авто-дополнения при вводе), визуальный редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных.*

*При запуске среды по центру экрана отображается окно «Начальная страница». В этом окне можно создать новый проект, открыть проект с диска или выбрать из списка проект, с которым работали ранее.*

*После создания или открытия проекта по центру окна будут на разных закладках отображаться файлы проекта. Слева и справа будут отображаться вспомогательные панели.*

*Каждую панель можно перетащить в любое место экрана. При этом будут появляться вспомогательные окна с возможностью указания позиции, на которой вы хотите поместить перетаскиваемую панель. Если какой-то из панелей нет на экране, то отобразить ее можно с помощью команды меню «Вид» или с помощью специальных кнопок на панели задач.*

*На панели «Обозреватель решений» отображаются все файлы, входящие в проект. Двойным щелчком можно открыть любой файл для редактирования. В контекстном меню любого файла имеются стандартные команды переименования, удаления и т.п. На этой же панели в контекстном меню проекта имеются команды для добавления новых или существующих на диске элементов проекта (форм, классов, библиотек и т.п.)*

*На панели «Свойства» отображаются свойства выделенного компонента. При этом в нижней части панели отображается краткое описание выбранного свойства. На этой же панели можно задать тип группировки свойств (по категориям, в алфавитном порядке), а также переключиться на отображение списка событий. В режиме редактирования кода на данной панели никаких свойств или событий не отображается.*

*На панели «Панель элементов» отображается список компонентов, которые можно разместить на форме проекта. Компоненты могут быть видимыми и невидимыми. Если нанести на форму невидимый элемент, то под формой автоматически создается отдельная область, на которой размещаются все невидимые элементы.*

*Если Вы находитесь в режиме редактирования кода, то компилятор автоматически проверяет правильность кода при вводе. При обнаружении ошибки ее место будет подчеркнуто красной линией. При вводе функций, методов, редактор автоматически «подсказывает» правильность написания. Если функция или метод имеет несколько вариантов вызова, то стрелками вверх/вниз можно перейти на нужный вариант вызова.*

*Для запуска программы на выполнение используется специальная кнопка на панели инструментов «Начать отладку» или клавиша F5. Если компилятор обнаружит ошибку, то он выдаст запрос, выполнить ли программу без учета ошибки. Если ответить «Да», то программа выполнит последнюю успешную версию до внесенных изменений, но ошибочный код работать не будет. Если ответить «Нет», то программа не выполняется.*

*2 Проектирование и реализация модулей программной системы*

*2.1 Проектирование данных*

*При анализе предметной области были выделены следующие сущности:*

*«Специальность», «Группа», «Приказы на перевод», «Дисциплины практик», «Дисциплины курсовых», «Дисциплины промежуточного контроля», «Студент», «Приказы на студента», «Результаты курсовых», «Результаты практик», «Результаты промежуточного контроля», «Научная работа», «Внеаудиторная деятельность», «Дополнительное образование», «Языки», «Предыдущее образование», «Квалификационная работа»,*

*Для каждой из сущностей были определены атрибуты.*

*Сущность «Специальность» содержит атрибуты:*

*– код специальности;*

*– название специальности;*

*Первичный ключ – «Код специальности».*

*Сущность «Группа» содержит атрибуты:*

*– код группы;*

*– код специальности;*

*– форма обучения;*

*– название дисциплины;*

*Первичный ключ – «Код группы».*

*Сущность «Приказы на перевод» содержит атрибуты:*

*– код приказа не перевод;*

*– курс;*

*– номер приказа;*

*– дата приказа;*

*– содержание;*

*– код группы;*

*Первичный ключ – «Код приказа не перевод».*

*Сущность «Дисциплины практик» содержит атрибуты»:*

*– код дисциплины практик;*

*– вид практики;*

*– должность на время прохождения практики;*

*– дата начала;*

*– дата конца;*

*– код группы;*

*Первичный ключ – «Код дисциплины».*

*Сущность «Дисциплины курсовых» содержит атрибуты:*

*– код дисциплины курсовой;*

*– название дисциплины;*

*– семестр;*

*– код группы;*

*Первичный ключ – «Код дисциплины».*

*Сущность «Дисциплины промежуточного контроля» содержит атрибуты»:*

*– код дисциплины промежуточного контроля;*

*– название дисциплины;*

*– нагрузка;*

*– форма обучения;*

*– код группы;*

*– семестр;*

*Первичный ключ – «Код дисциплины».*

*Сущность «Студент» содержит атрибуты»:*

*– код студента;*

*– код группы;*

*– фамилия;*

*– имя;*

*– отчество;*

*– дата рождения;*

*– дата зачисления;*

*– номер зачисления;*

*– телефон;*

*– почта;*

*– уровень владения языками;*

*– фото;*

*Первичный ключ – «Код студента».*

*Сущность «Приказы на студента» содержит атрибуты:*

*– код приказов студента;*

*– курс;*

*– номер;*

*– содержание;*

*– код студента;*

*– тип приказа;*

*– дата приказа;*

*Первичный ключ – «Код приказов студента».*

*Сущность «Результаты практик» содержит атрибуты»:*

*– код результата практики;*

*– вид практики;*

*– должность на время прохождения практики;*

*– дата начала;*

*– дата конца;*

*– код студента;*

*– база прохождения;*

*– оценка;*

*Первичный ключ – «код результата практики».*

*Сущность «Результаты курсовых» содержит атрибуты»:*

*– код результата курсовой;*

*– название дисциплины;*

*– семестр;*

*– тема;*

*– оценка;*

*– код студента;*

*Первичный ключ – «Код результата курсовой».*

*Сущность «Результаты промежуточного контроля»:*

*– код результата промежуточного контроля;*

*– название дисциплины;*

*– нагрузка;*

*– форма обучения;*

*– итог;*

*– код студента;*

*– семестр;*

*Первичный ключ – «Код результата промежуточного контроля».*

*Сущность «Научная работа»:*

*– код научной работы;*

*– код студента:*

*– тип;*

*– результат;*

*– форма;*

*– уровень;*

*– тема;*

*– дата начала;*

*– дата конца;*

*– место проведения;*

*– фото;*

*– расширение фото;*

*Первичный ключ – «Код научной работы».*

*Сущность «Внеаудиторная деятельность»:*

*– код внеаудиторной деятельности;*

*– код студента;*

*– название;*

*– результат;*

*– форма;*

*– дата начала;*

*– дата конца;*

*– фото;*

*– расширение фото;*

*Первичный ключ – «код внеаудиторной деятельности».*

*Сущность «Дополнительное образование»:*

*– код дополнительного образования;*

*– код студента;*

*– результат;*

*– учреждение;*

*– объём;*

*– направление;*

*– фото;*

*– расширение фото;*

*Первичный ключ – «Код дополнительного образования».*

*Сущность «Языки»:*

*– код языка;*

*– код студента;*

*– название языка;*

*Первичный ключ – «код языка».*

*Сущность «Квалификационная работа»:*

*– тема;*

*– оценка;*

*– код квалификационной работы;*

*– код студента;*

*Первичный ключ – «код квалификационной работы».*

*Сущность «Предыдущее образование»:*

*– код предыдущего образования;*

*– дата начала;*

*– дата конца;*

*– учреждение;*

*– квалификация;*

*– код студента;*

*Первичный ключ – «код предыдущего образования».*

*На основании сущностей спроектируем базу данных «portfolio», состоящую из 17 таблиц, структура которых приведена в таблицах 1–17.*

*Таблица 1 – Структура таблицы «spec»*

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Типы данных, длина* | *Ограничение* | *Описание* |
| *id* | *int* | *первичный ключ* | *Код отделения* |
| *naz* | *varchar(150)* | *–* | *Название* |

*Таблица 2 – Структура таблицы «group»*

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Типы данных, длина* | *Ограничение* | *Описание* |
| *id* | *int* | *первичный ключ* | *Код отделения* |
| *idSpec* | *int* | *–* | *Код специальности* |
| *formO* | *varchar(150)* | *–* | *Форма* |
| *naz* | *varchar(150)* | *–* | *Название* |

*Таблица 3 – Структура таблицы «stud»*

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Типы данных, длина* | *Ограничение* | *Описание* |
| *id* | *int* | *первичный ключ* | *код отделения* |
| *idGroup* | *int* | *–* | *код группы* |
| *fam* | *varchar(150)* | *–* | *Фамилия* |
| *name* | *varchar(150)* | *–* | *Имя* |
| *otch* | *varchar(150)* | *–* | *Отчество* |
| *dateR* | *date* | *–* | *Дата рождения* |
| *dateZachisl* | *date* | *–* | *Дата зачисления* |
| *nomZachisl* | *varchar(150)* | *–* | *Номер зачисления* |
| *tel* | *varchar(150)* | *–* | *Телефон* |
| *email* | *varchar(150)* | *–* | *Почта* |
| *urVlad* | *varchar(150)* | *–* | *Владение языками* |
| *foto* | *image* | *–* | *Фото* |

*Таблица 4 – Структура таблицы «prikazPerevod»*

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Типы данных, длина* | *Ограничение* | *Описание* |
| *id* | *int* | *первичный ключ* | *Код приказа* |
| *kurs* | *varchar(150)* | *–* | *Курс* |
| *nomer* | *varchar(150)* | *–* | *Номер* |
| *date* | *date* | *–* | *Дата приказа* |
| *soderj* | *varchar(150)* | *–* | *Содержание* |
| *idGroup* | *int* | *–* | *Код группы* |

*Таблица 5 – Структура таблицы «prikazStud»*

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Типы данных, длина* | *Ограничение* | *Описание* |
| *id* | *int* | *первичный ключ* | *Код приказа* |
| *kurs* | *int* | *–* | *Курс* |
| *nomer* | *varchar(150)* | *–* | *Номер* |
| *soderj* | *varchar(150)* | *–* | *Содержание* |
| *idStud* | *int* | *–* | *Код студента* |
| *tip* | *varchar(150)* | *–* | *Тип* |
| *date* | *date* | *–* | *Дата* |

*Таблица 6 – Структура таблицы «nauRab2»*

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Типы данных, длина* | *Ограничение* | *Описание* |
| *id* | *int* | *первичный ключ* | *Код нау. раб.* |
| *idStud* | *int* | *–* | *Код студента* |
| *tip* | *varchar(150)* | *–* | *Тип* |
| *rez* | *varchar(150)* | *–* | *Результат* |
| *forma* | *varchar(150)* | *–* | *Форма* |
| *uroven* | *varchar(150)* | *–* | *Уровень* |
| *tema* | *varchar(150)* | *–* | *Тема* |
| *dateOt* | *date* | *–* | *Дата начала* |
| *dateDo* | *date* | *–* | *Дата конца* |
| *mestoProv* | *varchar(150)* | *–* | *Место проведения* |
| *foto* | *image* | *–* | *Фото* |
| *fotoF* | *varchar(150)* | *–* | *Расширение* |

*Таблица 7 – Структура таблицы «vneRab3»*

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Типы данных, длина* | *Ограничение* | *Описание* |
| *id* | *int* | *первичный ключ* | *Код вне. раб.* |
| *idStud* | *int* | *–* | *Код студента* |
| *naz* | *varchar(150)* | *–* | *Название* |
| *rez* | *varchar(150)* | *–* | *Результат* |
| *forma* | *varchar(150)* | *–* | *Форма* |
| *dateOt* | *date* | *–* | *Дата начала* |
| *dateDo* | *date* | *–* | *Дата конца* |
| *foto* | *image* | *–* | *Фото* |
| *fotoF* | *varchar(150)* | *–* | *Расширение фото* |

*Таблица 8 – Структура таблицы «dopRab4»*

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Типы данных, длина* | *Ограничение* | *Описание* |
| *id* | *int* | *первичный ключ* | *Код доп. раб.* |
| *idStud* | *int* | *–* | *Код студента* |
| *rez* | *varchar(150)* | *–* | *Результат* |
| *uchir* | *varchar(150)* | *–* | *Учреждение* |
| *oby* | *varchar(150)* | *–* | *Объём* |
| *napr* | *varchar(150)* | *–* | *Направление* |
| *foto* | *image* | *–* | *Фото* |
| *fotoF* | *varchar(150)* | *–* | *Расширение фото* |

*Таблица 9 – Структура таблицы «prevObr»*

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Типы данных, длина* | *Ограничение* | *Описание* |
| *id* | *int* | *первичный ключ* | *Код доп. раб.* |
| *dateOt* | *date* | *–* | *Дата начала* |
| *dateDo* | *date* | *–* | *Дата конца* |
| *ucher* | *varchar(150)* | *–* | *Учреждение* |
| *kval* | *varchar(150)* | *–* | *Квалификация* |
| *idStud* | *int* | *–* | *Код студента* |

*Таблица 10 – Структура таблицы «kavlRab»*

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Типы данных, длина* | *Ограничение* | *Описание* |
| *id* | *int* | *первичный ключ* | *Код квал. раб.* |
| *tema* | *varchar(150)* | *–* | *Тема* |
| *ocenka* | *varchar(150)* | *–* | *Оценка* |
| *idStud* | *int* | *–* | *Код студента* |

*Таблица 11 – Структура таблицы «jaz»*

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Типы данных, длина* | *Ограничение* | *Описание* |
| *id* | *int* | *первичный ключ* | *Код языка* |
| *idStud* | *int* | *–* | *Код студента* |
| *nameJaz* | *varchar(50)* | *–* | *Название языка* |

*Таблица 12 – Структура таблицы «prakt»*

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Типы данных, длина* | *Ограничение* | *Описание* |
| *id* | *int* | *первичный ключ* | *Код практики* |
| *vid* | *varchar(150)* | *–* | *Вид* |
| *baza* | *varchar(150)* | *–* | *База* |
| *dolj* | *varchar(150)* | *–* | *Должность* |
| *dateOt* | *date* | *–* | *Дата начала* |
| *dateDo* | *date* | *–* | *Дата конца* |
| *ocenka* | *varchar(150)* | *–* | *Оценка* |
| *idStud* | *int* | *–* | *Код студента* |

*Таблица 13 – Структура таблицы «prPrakt»*

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Типы данных, длина* | *Ограничение* | *Описание* |
| *id* | *int* | *первичный ключ* | *Код практики* |
| *vid* | *varchar(150)* | *–* | *Вид* |
| *dolj* | *varchar(150)* | *–* | *Должность* |
| *dateOt* | *date* | *–* | *Дата начала* |
| *dateDo* | *date* | *–* | *Дата конца* |
| *idGroup* | *int* | *–* | *Код группы* |

*Таблица 14 – Структура таблицы «kurs»*

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Типы данных, длина* | *Ограничение* | *Описание* |
| *id* | *int* | *первичный ключ* | *Код курсовой* |
| *disc* | *varchar(150)* | *–* | *Дисциплина* |
| *sem* | *varchar(150)* | *–* | *Семестр* |
| *tema* | *varchar(150)* | *–* | *Тема* |
| *ocenka* | *varchar(150)* | *–* | *Оценка* |
| *idStud* | *int* | *–* | *Код студента* |

*Таблица 15 – Структура таблицы «prKurs»*

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Типы данных, длина* | *Ограничение* | *Описание* |
| *id* | *int* | *первичный ключ* | *Код курсовой* |
| *disc* | *varchar(150)* | *–* | *Дисциплина* |
| *sem* | *varchar(150)* | *–* | *Семестр* |
| *idGroup* | *int* | *–* | *Код группы* |

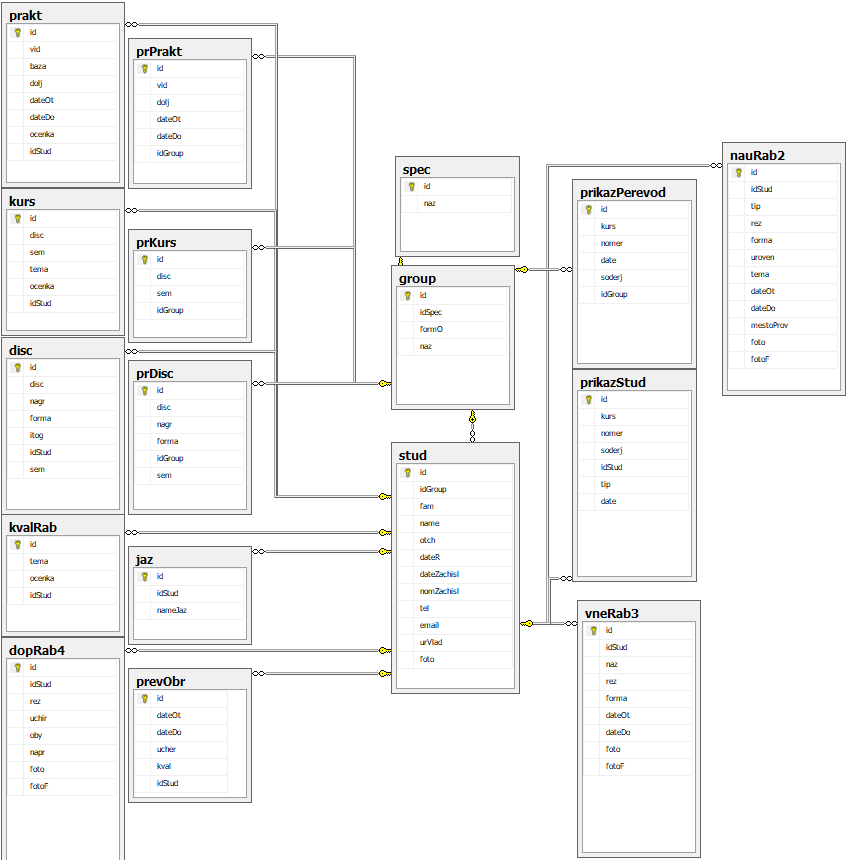
*Таблица 16 – Структура таблицы «disc»*

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Типы данных, длина* | *Ограничение* | *Описание* |
| *id* | *int* | *первичный ключ* | *Код дисциплины* |
| *disc* | *varchar(150)* | *–* | *Дисциплина* |
| *nagr* | *varchar(150)* | *–* | *Нагрузка* |
| *forma* | *varchar(150)* | *–* | *Форма* |
| *itog* | *varchar(150)* | *–* | *Оценка* |
| *idStud* | *int* | *–* | *Код студента* |
| *sem* | *varchar(150)* | *–* | *Семестр* |

*Таблица 17 – Структура таблицы «prDisc»*

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Типы данных, длина* | *Ограничение* | *Описание* |
| *id* | *int* | *первичный ключ* | *Код дисциплины* |
| *disc* | *varchar(150)* | *–* | *Дисциплина* |
| *nagr* | *varchar(150)* | *–* | *Нагрузка* |
| *forma* | *varchar(150)* | *–* | *Форма* |
| *idGroup* | *int* | *–* | *Код группы* |
| *sem* | *varchar(150)* | *–* | *Семестр* |

*Общая структура базы данных приведена на рис.1*

**

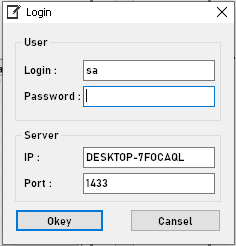
*Рисунок 1 – Структура базы данных*

*2.2 Описание программного обеспечения*

*Главная форма программы построена на основе MDI интерфейса. Приложение имеет интуитивно понятное меню для доступа ко всем необходимым функциям приложения.*

*Программа позволяет выполнять стандартный набор действий: просмотр и редактирование данных, составление отчетов.*

*Для запуска приложения необходимо выполнить подключение к серверу и ввести данные для авторизации. Форма авторизации представлена на рисунке 2.*



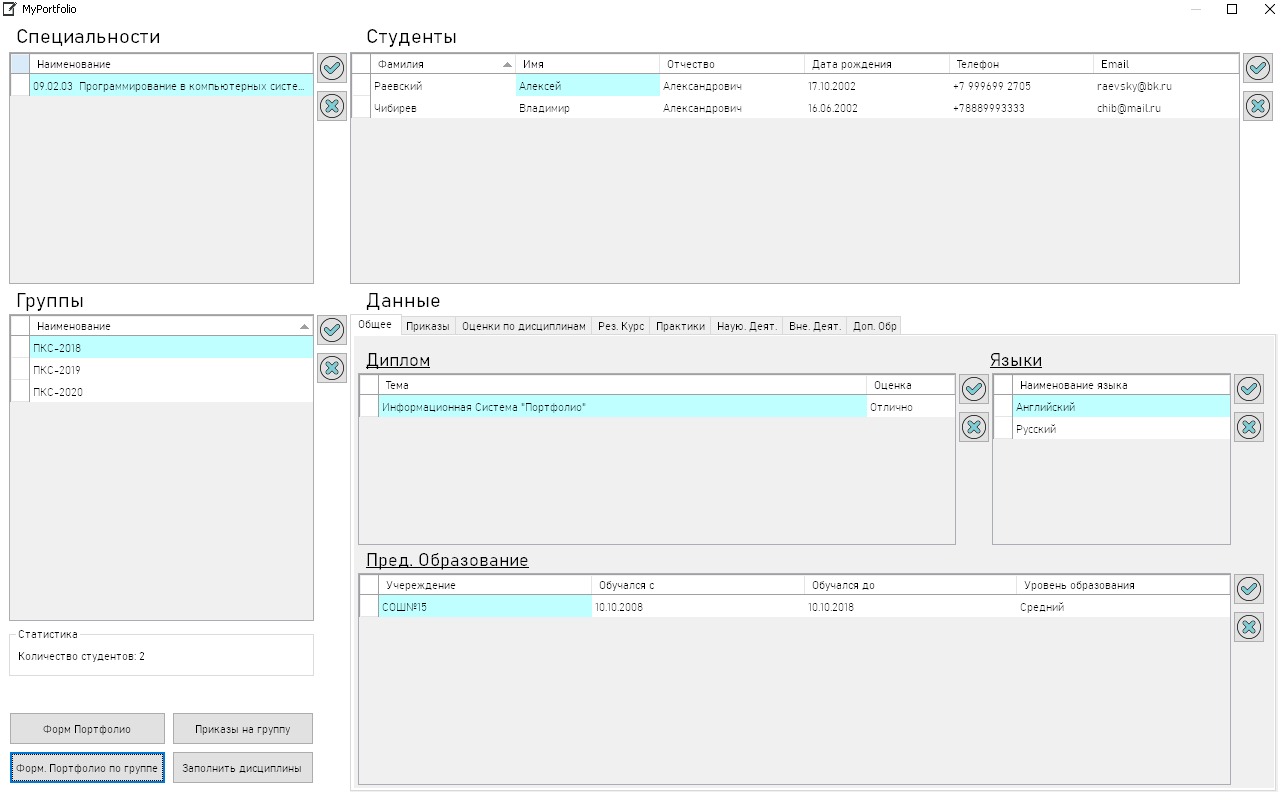
*Рисунок 2 – Форма авторизации*

*На данной форме не используются переменные, а данные хранятся в настройке самой программы (Setings.Setings). После успешной авторизации все данные сохраняются, за исключением пароля.*

*Таблица 16 – Структура таблицы «disc»*

|  |  |  |  |
| --- | --- | --- | --- |
| *Имя* | *Тип* | *Область* | *Назначение* |
| *user* | *string* | *Пользователь* | *Логин* |
| *port* | *string* | *Пользователь* | *Порт* |
| *server* | *string* | *Пользователь* | *IP адрес* |

*После запуска приложения и корректного ввода данных настройки сервера и авторизации приложение перейдет в полноэкранный режим и запустит главную форму. Главная форма представлена на рисунке 3.*



*Рисунок 3 – Главная форма*

*Таблица 17 – переменные модуля «General.cs»*

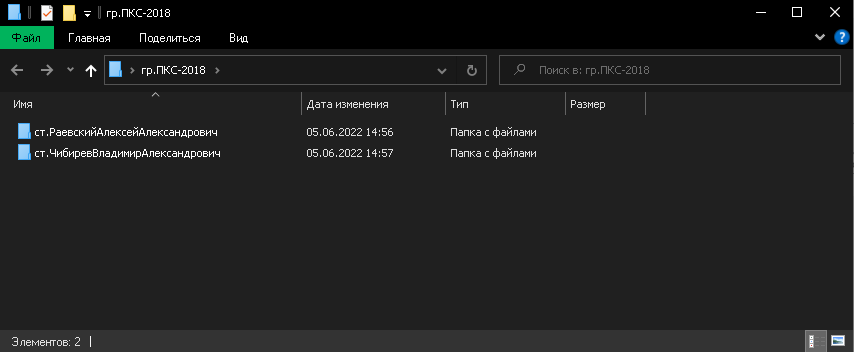
|  |  |  |
| --- | --- | --- |
| *Имя* | *Тип* | *Назначение* |
| *pos* | *Integer* | *Используется для хранения позиции выбранного элемента в таблице.* |
| *w* | *Word.Application* | *Объект класса Word для работы с приложение Word* |

*На главной форме представленные основные таблицы для работы с базой данных, а также вкладки для отображения остальных таблиц относящихся к конкретному студенту. На этой же форме мы можем добавлять новые специальности, группы, а также студентов. Редактировать уже существующие названия специальностей групп и некоторую информацию о студенте.*

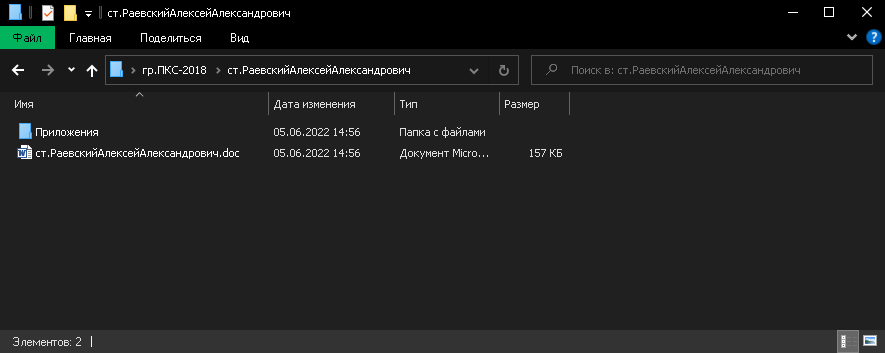
*Добавление и удаление данных в таблицы происходит по средствам нажатия кнопок по правую часть от таблиц, а изменение при помощи нажатия левой кнопкой мыши по выбранному полю в таблице.*

*На форме присутствуют четыре кнопки. Кнопки «Форм. портфолио» и «Форм. Портфолио по группе» используются для создания отчёта по выбранной записи в таблице. При нажатии на кнопку «Форм. портфолио» сформируется портфолио выбранного студента, в папку, выбранную из контекстного меню после нажатия. «Форм. Портфолио по группе» выполняет ту же функцию, но формирует портфолио по всей группе.*

*В результате формирования портфолио по группе в выбранной папке сформируется каталог (рисунок 4) включающий папки с именами студентов данной группы, содержащие «Портфолио» и изображения грамот, дипломов и так далее (рисунок 5).*



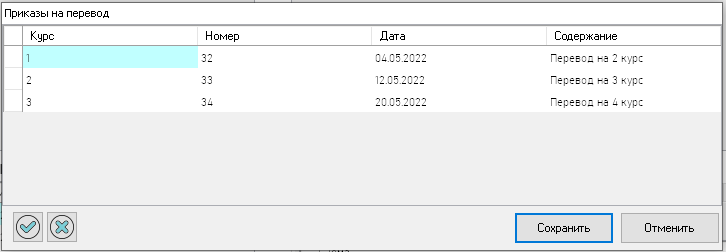
*Рисунок 4 – Каталог группы*



*Рисунок 5 – Каталог студента*

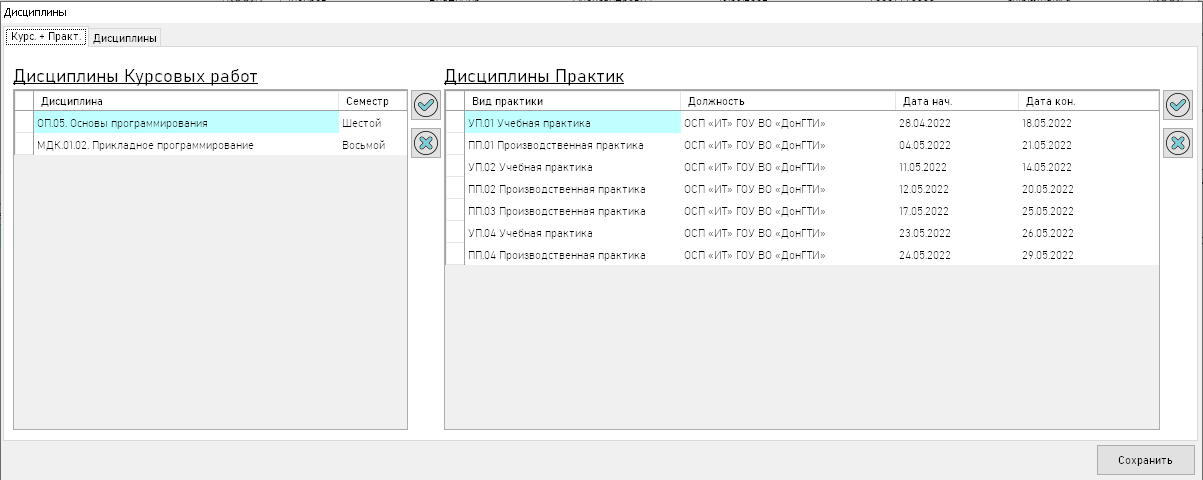
*Для заполнения общих данных для группы используются кнопки «Приказы на группу» и «Заполнить дисциплины».*

*Кнопка «Приказы на группу» открывает форму (рисунок 6) на которой пользователь может ввести, отредактировать, удалить, приказы, выданные для конкретной группы после чего эти данные, будут использоваться для формирования отчёта и будут общими для все группы что позволит сократить время на заполнения отчёта.*

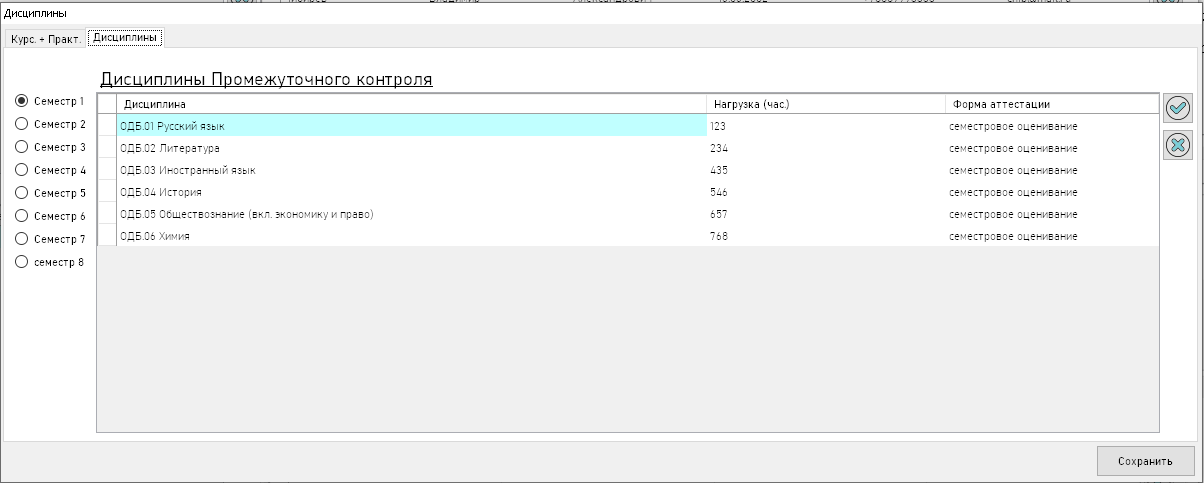


*Рисунок 6 – Каталог студента*

*Кнопка «Заполнить дисциплины» открывает форму (рисунок 7) на которой производится заполнение данных о курсовых, практиках, а также предметах, разбитых по семестрам (рисунок 8). Данные на этой форме являются общими для всех студентов конкретной группы.*

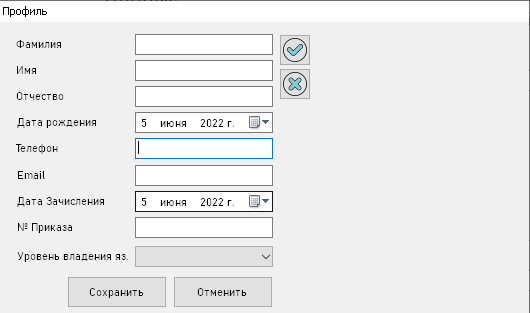


*Рисунок 7 – Таблицы с курсовыми и практиками*



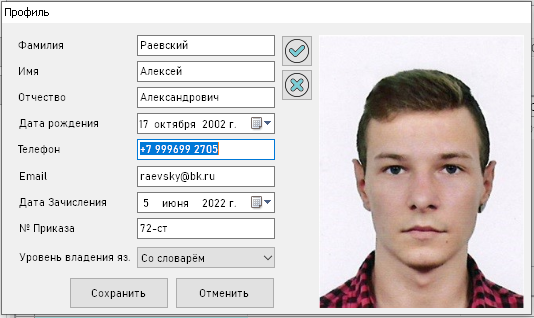
*Рисунок 8 – Каталог дисциплин*

*При добавлении нового студента в базу данных отображается форма (рисунок 9). На которой пользователь вводит всю личную информацию касающуюся студента, которого необходимо добавить. Также при помощи двух кнопок реализуется добавление фото студента.*



*Рисунок 9 – Добавление студента*

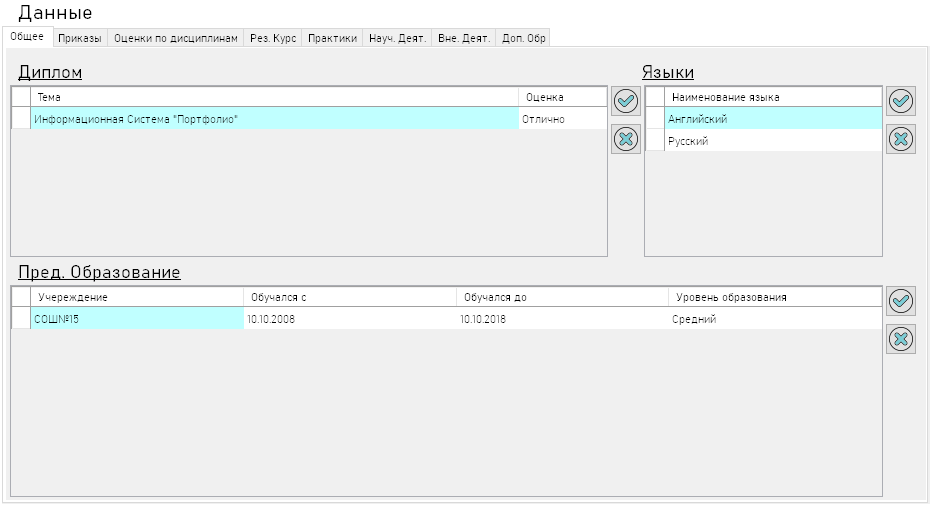
*При необходимости редактирования студента необходимо выполнить двойной клик по полю с нужным студентом после чего откроется форма (рисунок 10) аналогичная форме добавления, но с данными конкретного студента.*



*Рисунок 10 – Добавление студента*

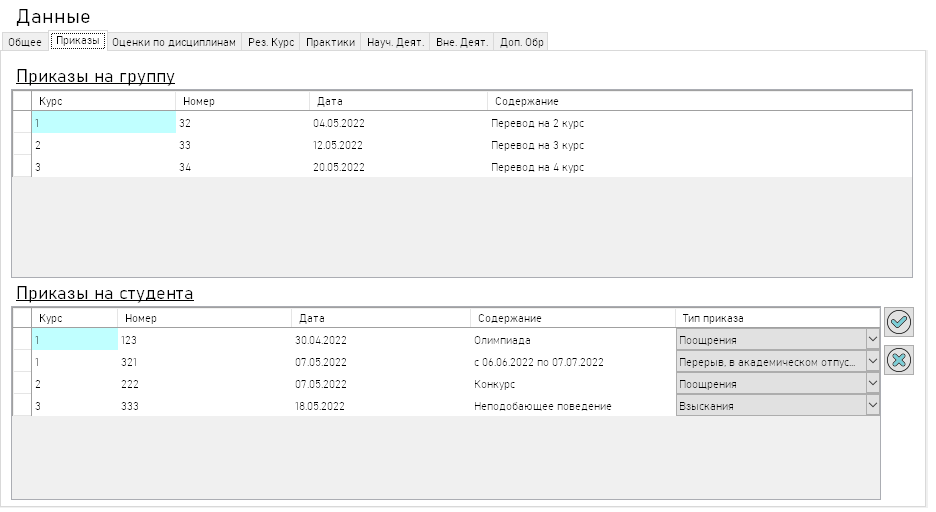
*На главной форме помимо кнопок присутствует область «Данные» (рисунок 9) содержащая в себе восемь вкладок («Общее», «Приказы», «Оценки по дисциплинам», «Рез. Курс», «Практики», «Науч. Деят», «Вне. Деят.», «Доп. Обр.»). Каждая вкладка служит для заполнения определённой информации о студенте.*

*Вкладка «Общее» (рисунок 11) хранит в себе данные о теме диплома и оценке, а также языках и предыдущем образовании.*



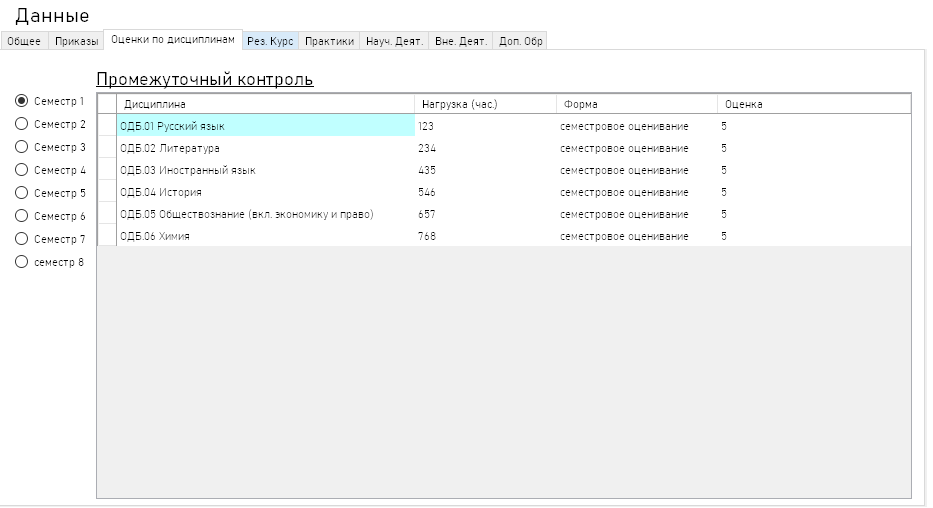
*Рисунок 11 – Область «Данные»*

*Вкладка «Приказы» (рисунок 12) хранит в себе данные о приказах, выданных на группу в которой состоит студент, а также приказы, выданные на имя этого студента. Данные о приказах на группу тут находятся в состоянии просмотра и не могут быть отредактированы.*



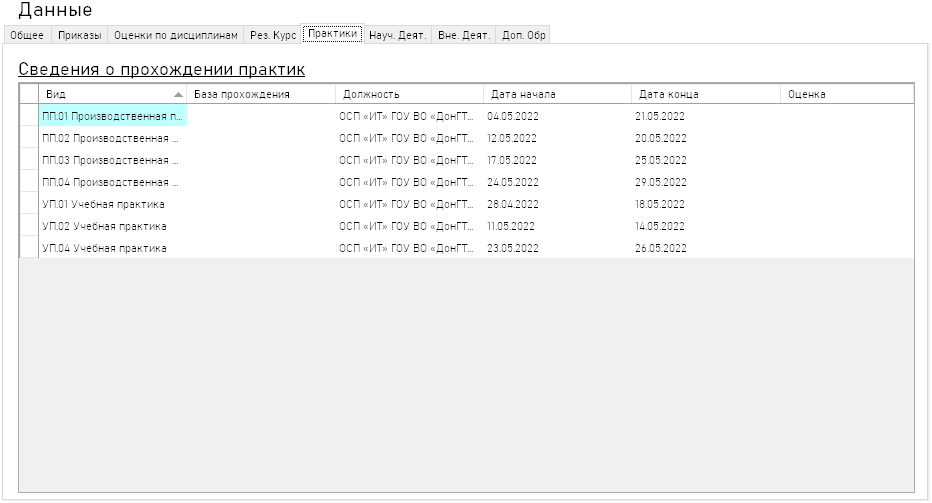
*Рисунок 12 – Область «Данные»*

*Вкладка «Оценки по дисциплинам» (рисунок 13) содержит данные о дисциплинах студента, а также его оценке. Поле оценка для каждого студента будет уникальным, а данные о дисциплине общими для всей группы.*



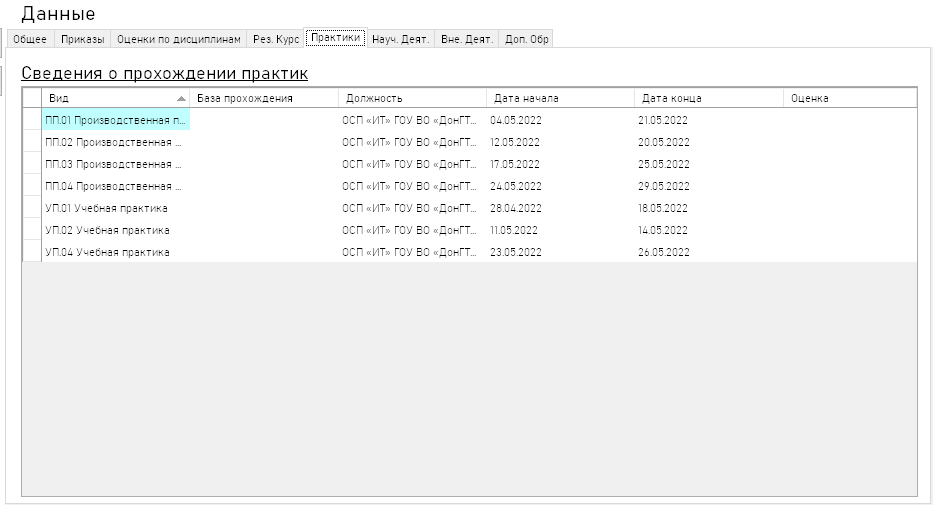
*Рисунок 12 – Область «Данные»*

*Вкладка «Рез. Курс» (рисунок 13) содержит данные о дисциплинах курсовых работ, а также теме и оценке выбранного студента. Поле оценка и тема для каждого студента будет уникальным, а данные о дисциплине общими для всей группы.*



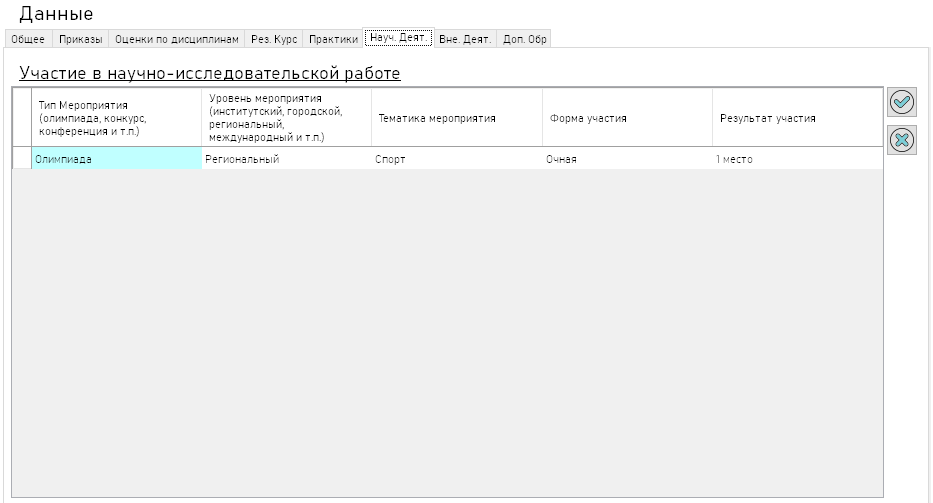
*Рисунок 13 – Область «Данные»*

*Вкладка «Практики» (рисунок 14) содержит данные о виде практике, должности, времени прохождения, а также базе и оценке выбранного студента.*



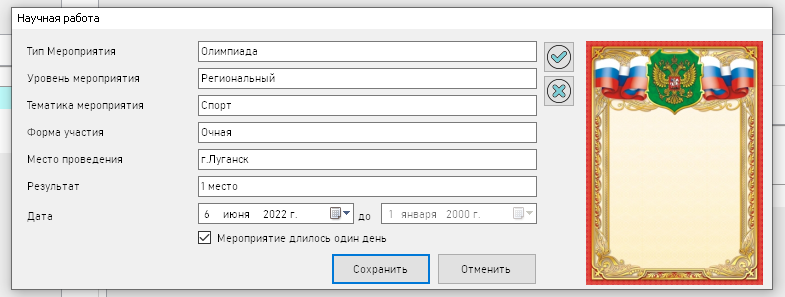
*Рисунок 14 – Область «Данные»*

*Вкладка «Науч. Деят.» (рисунок 15) хранит данные о участии студента в научно исследовательской работе.*



*Рисунок 15 – Область «Данные»*

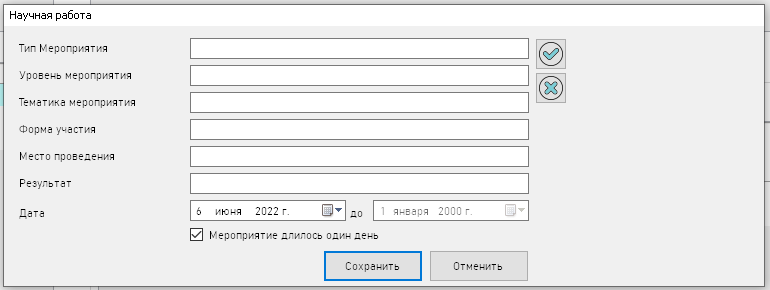
*Данные в таблице представлены в упрощённом виде для просмотра более детальной информации следует выполнить двойной клик по полю, после чего откроется форма (рисунок 16), на которой информация будет расширена полем с изображением грамоты, либо документом, удостоверяющим участие в научной деятельности.*



*Рисунок 16 – Форма «Научная деятельность»*

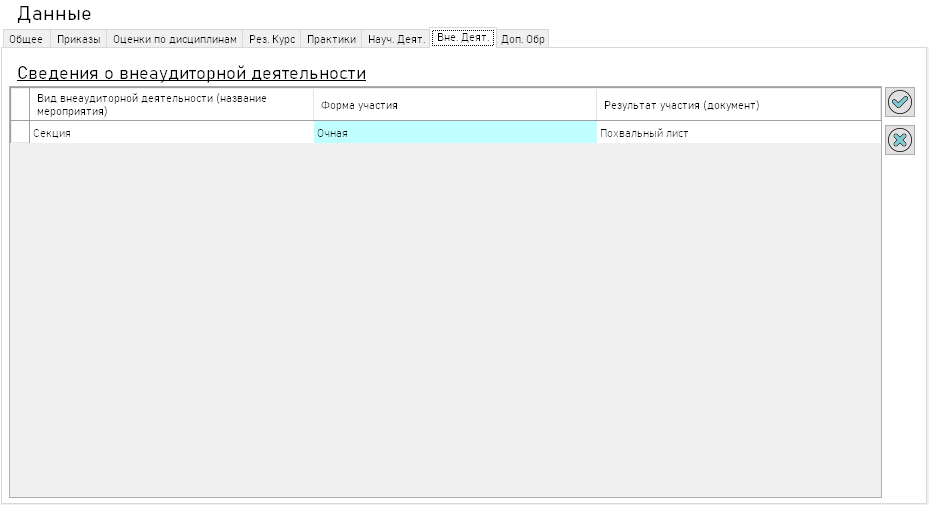
*Аналогичная форма будет открываться при добавлении новой формы (рисунок 17), но с пустыми полями.*

*Изображение на форме добавляется при помощи двух кнопок в правой верхней части формы.*



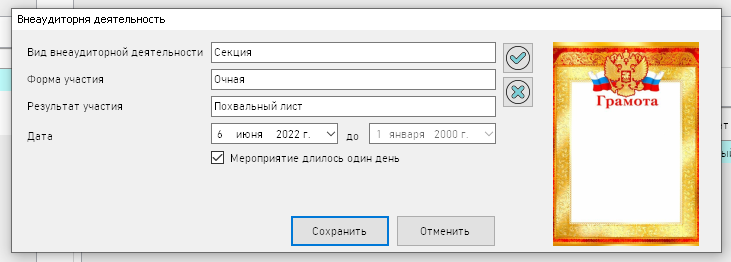
*Рисунок 17 – Форма «Научная деятельность»*

*Вкладка «Вне. Деят.» (рисунок 18) хранит данные о участии студента во внеаудиторной деятельности.*



*Рисунок 18 – Область «Данные»*

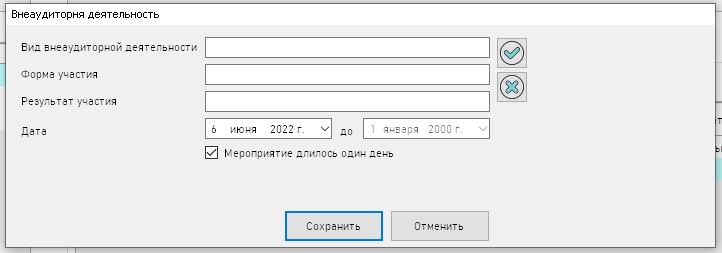
*Данные в таблице представлены в упрощённом виде для просмотра более детальной информации следует выполнить двойной клик по полю, после чего откроется форма (рисунок 19), на которой информация будет расширена полем с изображением грамоты, либо документом, удостоверяющим участие во внеаудиторной деятельности.*



*Рисунок 19 – Форма «Внеаудиторная деятельность»*

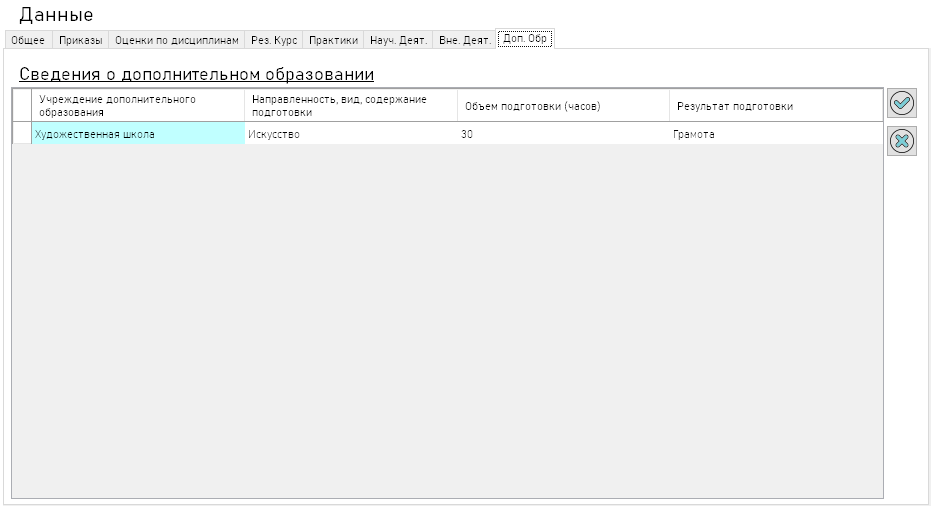
*Аналогичная форма будет открываться при добавлении новой формы (рисунок 20), но с пустыми полями.*

*Изображение на форме добавляется при помощи двух кнопок в правой верхней части формы.*



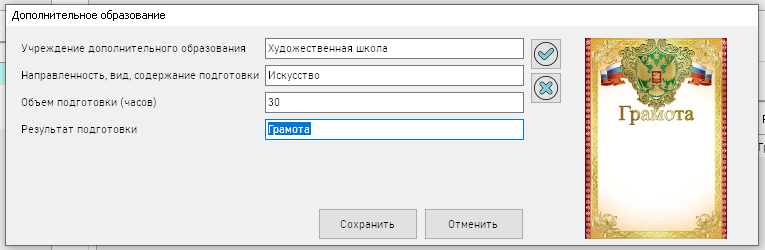
*Рисунок 20 – Форма «Внеаудиторная деятельность»*

*Вкладка «Доп. Обр.» (рисунок 21) хранит данные дополнительном образовании студента.*



*Рисунок 21 – Область «Дополнительное образование»*

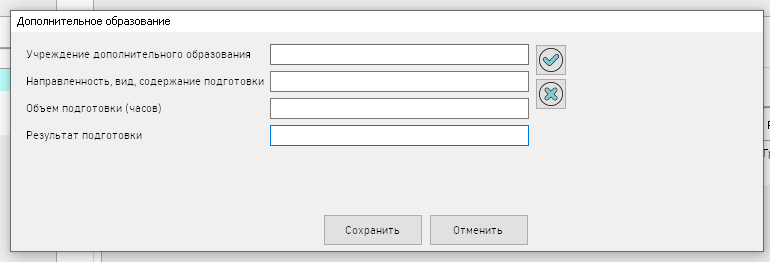
*Данные в таблице представлены в упрощённом виде для просмотра более детальной информации следует выполнить двойной клик по полю, после чего откроется форма (рисунок 22), на которой информация будет расширена полем с изображением грамоты, либо документом, о дополнительном образовании.*



*Рисунок 22 – Форма «Дополнительное образование»*

*Аналогичная форма будет открываться при добавлении новой формы (рисунок 23), но с пустыми полями.*

*Изображение на форме добавляется при помощи двух кнопок в правой верхней части формы.*



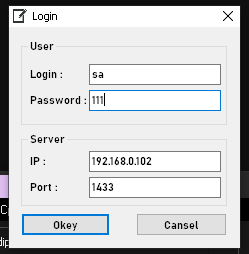
*Рисунок 23 – Форма «Дополнительное образование»*

*3 Тестирование*

*Для проверки работоспособности программной системы выполним несколько тестовых заданий.*

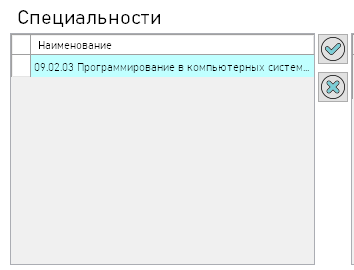
*Первым сценарием будет внесение студентов после зачисления их на первый курс. Данный сценарий подразумевает, что у студентов будут заполнены все поля касающиеся личной информации. После чего будет сформирован отчёт.*

*Откроем программу и выполним авторизацию (рисунок 24).*



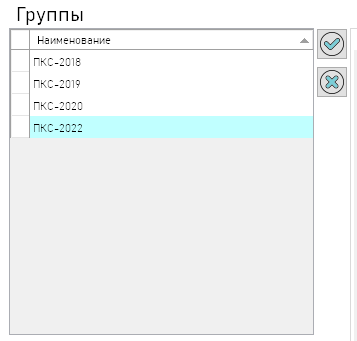
*Рисунок 24 – Форма «Авторизации»*

*После чего перед нами откроется главная форма где в левой верхней части мы должны будем либо внести новую специальность, либо выбрать уже существующую (рисунок 25). В данном случае будет внесена новая специальность «09.02.03 Программирование в компьютерных системах».*



*Рисунок 25 – Таблица со специальностями*

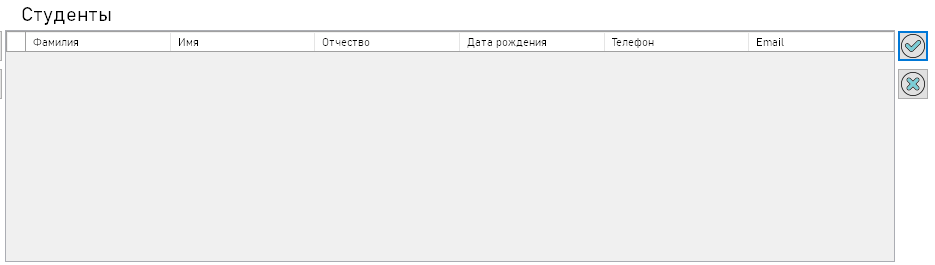
*После добавления специальности либо выбора уже существующих необходимо добавить новую группу или выбрать существующую. Для этого ниже таблицы «Специальности» находится таблица с группами (рисунок 26)*



*Рисунок 26 – Таблица с группами*

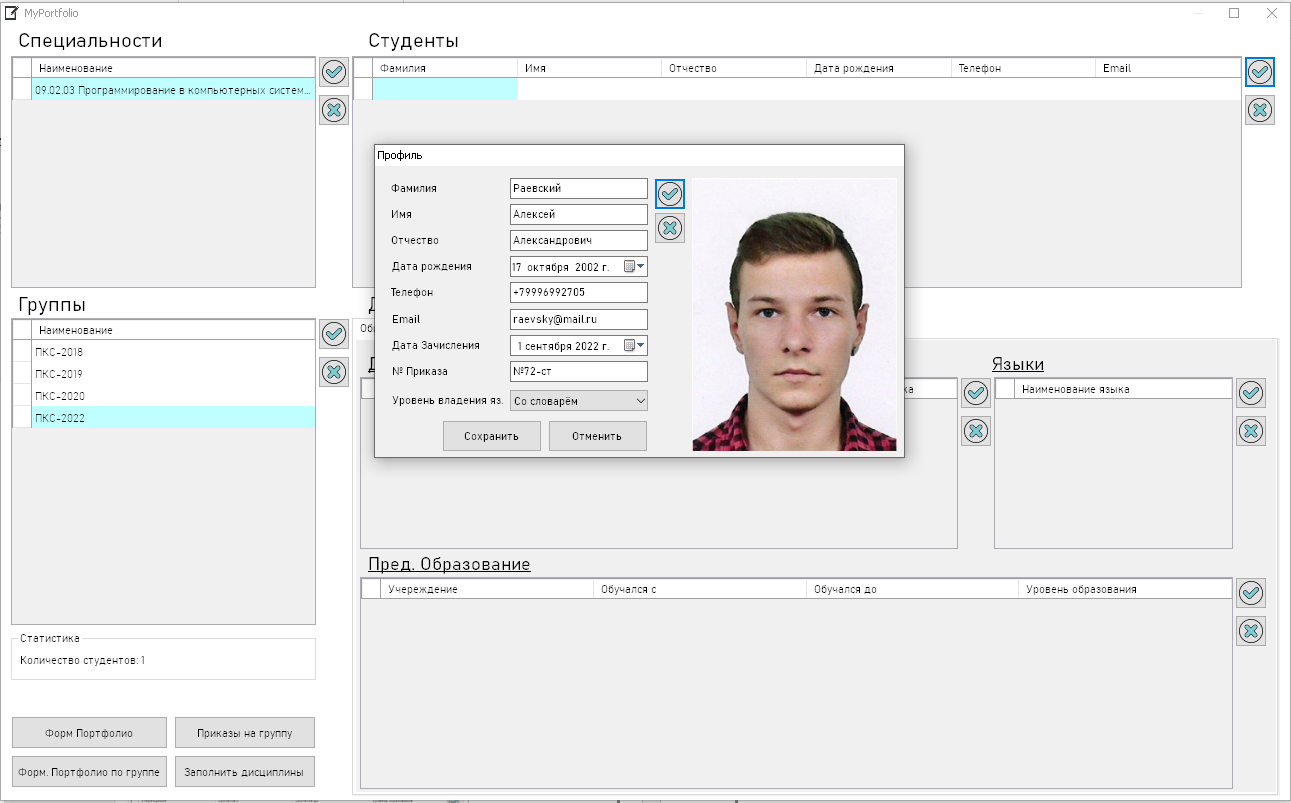
*В нашем случае будет внесена новая группа «ПКС-2022».*

*После создания специальности и группы можно перейти к добавлению студентов в таблицу студенты (рисунок 27).*



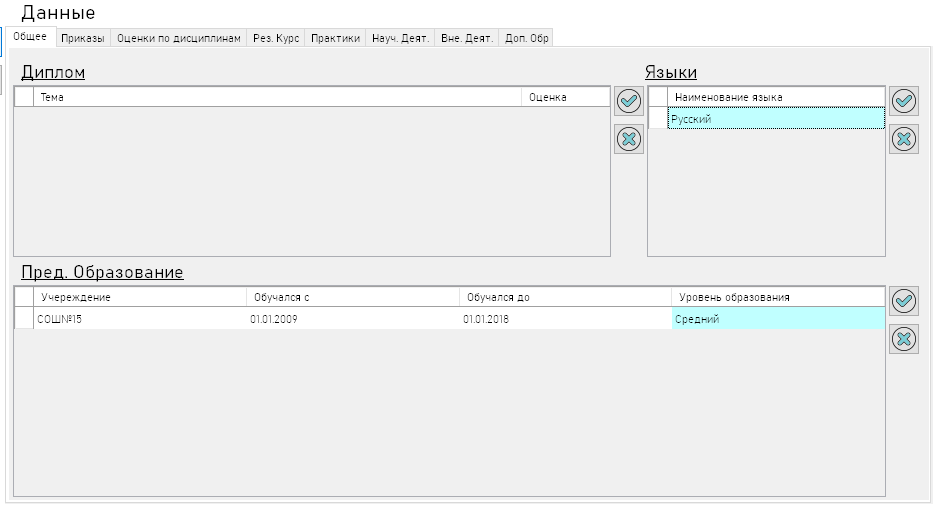
*Рисунок 27 – Таблица со студентами*

*После нажатия на кнопку добавления студента, открывается форма с личной информацией о нём (рисунок 28).*



*Рисунок 28 – Форма с личной информацией*

*После внесения студента, из личных данных о которые можно заполнить при поступлении остались только предыдущее образование и языки. Данные поля заполняются в таблицах, которые находятся в области «Данные» во вкладке «Общее» (рисунок 29).*



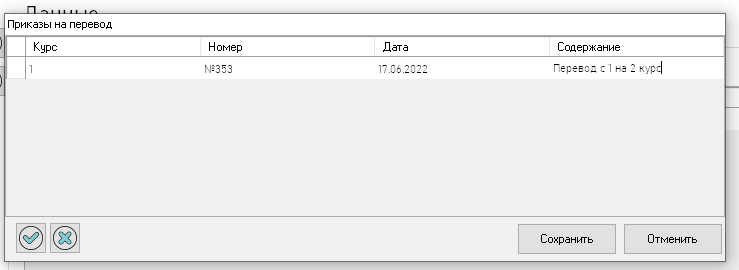
*Рисунок 29 – Вкладка «Общее»*

*После внесения языков и предыдущего образования можно перейти к добавлению следующего студента выполнив аналогичные действия начиная с таблицы «Студенты».*

*Следующим сценарием будет внесение: приказа на перевод, промежуточного контроля, курсовых, практик.*

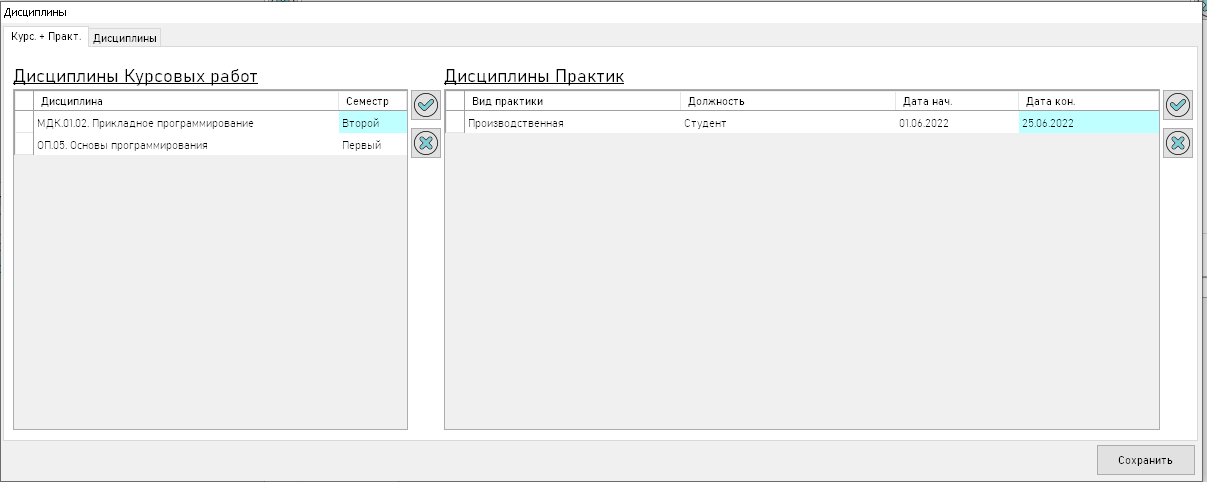
*Откроем программу и выполним авторизацию. Выберем интересующую группу в данном примере это будет «ПКС-2022».*

*Сначала заполним приказ на перевод с 1 на 2 курс. Для этого нажмём на кнопку «Приказы на группу» и в открывшейся форме (рисунок 30) заполним данные касающиеся этого приказа*



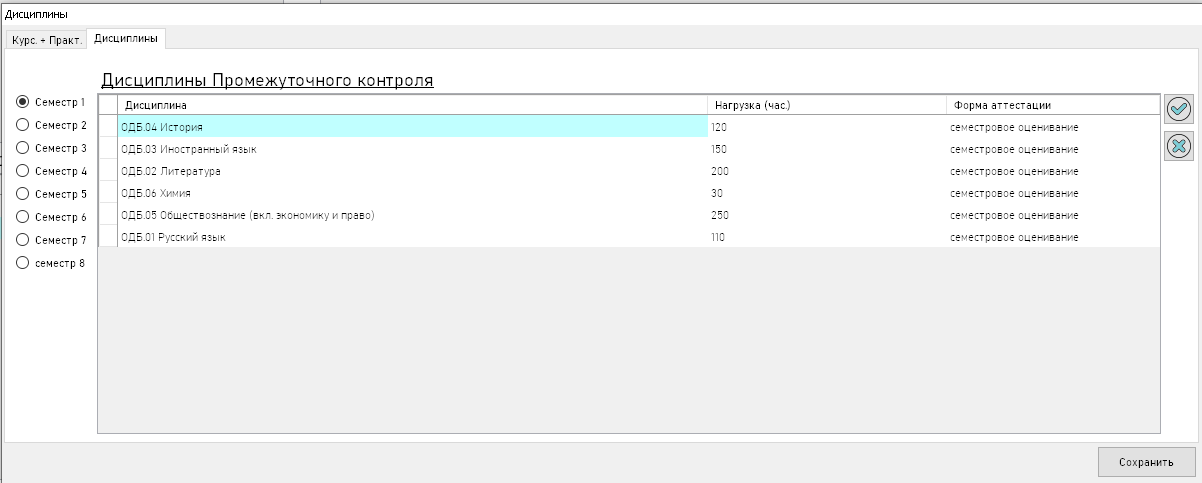
*Рисунок 30 – Форма «Приказы на перевод»*

*После заполнения приказа на перевод можно перейти к заполнению дисциплин по курсовым, практикам и промежуточному контролю. Для этого на главной форме нажмём кнопку «Заполнить дисциплины», в открывшейся форме (рисунок 31) на вкладке «Курс + Практ.» заполним данные касающиеся курсовых и практик.*



*Рисунок 31 – Форма «Дисциплины»*

*После заполнения практик и курсовых нужно перейти во вкладку «Дисциплины» (рисунок 32) и заполнить данные касающиеся промежуточного контроля.*



*Рисунок 32 – Вкладка «Дисциплины»*

*После заполнения данных на главной форме в соответствующих вкладках появятся дисциплины, нагрузки и так далее, останется только заполнить оценки.*

*Заключение*

*Неоспоримым преимуществом автоматизированных систем управления является то, что они не требуют никаких дополнительных вложений. Автоматизированные системы управления достаточно легко поддаются необходимым корректировкам в соответствии с требованиями рабочего процесса предприятия.*

*Для лучшего понимания принципов работы информационной системы была изучена ее предметная область. Также были изучены существующие программные аналоги, предназначенные для автоматизации работы в рассматриваемой предметной области, с целью расширения списка требований к интерфейсу и функциям создаваемой информационной системы.*

*На основании требований, сформированных при изучении предметной области, в ходе преддипломной практики мной была спроектирована и создана база данных SQL Server. В ходе выполнения дипломной работы мной было создано клиент-серверное приложение.*

*По итогам выполнения дипломной работы была завершена полнофункциональная информационная система. Созданная база данных и клиент-серверное приложение позволяют выполнить весь спектр задач, поставленных перед информационной системой.*

*Список литературы*

1. *Архангельский А.Я. Delphi 7. Справочное пособие – М.: ООО "Бином-Пресс"*
2. *Вендров А.М. Практикум по проектированию программного обеспечения экономических информационных систем: Учеб. пособие. - М.: Финансы и статистика, 2004. - 192 с.*
3. *Гофман В.Є., Хомоненко А.Д. Работа с базами данных в Visual Studio – 2-е изд. - Спб.: Бхв-Петербург*
4. *Мамаев Е. Microsoft SQL Server. Наиболее полное руководство – К.: BHV-Киев, 2002.*
5. *Н. Елманова, С. Трепалин, А. Тенцлер. Visual Studio и технология COM - СПБ.: Питер, 2003 - 698 с.*
6. *Роберт Виейра. Программирование баз данных Microsoft SQL Server 2005. Базовый курс = Beginning Microsoft SQL Server 2005 Programming. - М.: «Диалектика», 2007. - С. 832.*
7. *Тернстрем, Т. Microsoft SQL Server 2008. Разработка баз данных. Учебный курс Microsoft 2011 г.*
8. *Конспект лекций по дисциплине «Технология разработки и защиты баз данных».*
9. *Конспект лекций по дисциплине «Прикладное программирование».*

*Приложение А - Техническое задание*

*A.1 Общие положения*

*A.1.1 Полное наименование подсистемы и ее условное обозначение*

*«Разработка информационной системы «Портфолио»*

*A.1.2 Наименования организации-заказчика и организаций-участников работ*

*Заказчик системы: ОСП «ИТ» ГОУ ВО ЛНР «ДонГТИ»*

*Адрес заказчика: ЛНР, Алчевск, пр. Дунауварошская 14*

*Разработчик подсистемы: студент группы ПКС-2019 специальности 09.02.03 «Программирование в компьютерных системах» Раевский Алексей Александрович*

*A.1.3 Перечень документов, на основании которых создается система*

*Разработка ведется в соответствии с:*

*- основной профессиональной образовательной программой по подготовке специалистов среднего звена по специальности 09.02.03 «Информатика и вычислительная техника»;*

*- Приказом от 05.04.2021 № 31-ст «О назначении руководителей и консультантов выпускной квалификационной работы, утверждения тем выпускных квалификационных работ студентам очной формы обучения»*

*A.1.4 Плановые сроки начала и окончания работы по созданию системы*

*Плановые сроки указываются на примере таблицы А1.*

*Таблица А1 – Плановые сроки создания подсистемы*

|  |  |  |
| --- | --- | --- |
| *№* | *Наименование работ* | *Сроки выполнения* |
| *1* | *Изучение предметной области, разработка ТЗ* | *Апрель 2022г. – Май 2022г.* |
| *2* | *Разработка пользовательского интерфейса, программирование функциональных модулей, тестирование, отладка, документирование* | *Май 2022г. – Июнь 2022г.* |

*A.1.5 Порядок оформления и предъявления заказчику результатов работ по созданию подсистемы*

*Подсистема передается в виде функционирующего комплекса на базе средств вычислительной техники Заказчика в установленные сроки.*

*A.1.6 Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ*

*При разработке ТЗ использовался ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению.*

*A.2 Назначение и цели создания программы*

*A.2.1 Назначение программы*

*Формирования отчёта «Портфолио»*

*A.2.2 Цели создания подсистемы*

*Добавления, редактирование, удаление данных в информационной системе с целью дальнейшего формирования отчёта по эти данным*

A.3 *Требования к программе*

*A.3.1 Требования к программе в целом*

*– реализация программы представляет приложение, написанное в IDE Visual Studio.*

*– вся информация хранится в базе данных СУБД Sql Server;*

*A.3.2 Требования к функциональным характеристикам*

*Подсистема должна обеспечивать выполнение следующих функций:*

*добавление данных;*

*редактирование данных;*

*построение отчетов на основе данных.*

*A.3.3 Требования к исходным кодам и языкам программирования*

*Серверная часть системы должна быть реализована с использованием СУБД SQL Server.*

*Клиентская часть подсистемы должна быть реализована при помощи Visual Studio.*

*A.3.4 Требования к надёжности*

*В программе должны быть предусмотрены средства защиты от некорректных действий пользователя и вывод соответствующих сообщений.*

*A.3.5 Требования к безопасности*

*В программе должна быть реализована защита от входа пользователя не имеющего доступ к базе данных.*

*A.3.6 Требования к документации*

*В состав программной документации входят:*

*общее описание программы;*

*инструкция пользователя;*

*техническое задание.*

*A.4 Стадии и этапы разработки*

*A.4.1 Стадии разработки*

*Разработка должна быть проведена в три стадии:*

*техническое задание;*

*технический (и рабочий) проекты;*

*тестирование.*

*A.4.2 Этапы разработки*

*На стадии «Техническое задание» должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.*

*На стадии «Технический (и рабочий) проект» должны быть выполнены перечисленные ниже этапы работ:*

*разработка программы;*

*разработка программной документации;*

*испытание программы.*

*A.5 Порядок приема контроля*

*Сдача-приёмка работ производится в соответствии с разработанным планом, в установленные сроки.*

*Все создаваемые в рамках настоящей работы программные изделия (за исключением покупных) передаются Заказчику, как в виде готовых модулей, так и в виде исходных кодов, представляемых в электронной форме на машинном носителе.*

*Приложение Б – Листинг программы*

*Исходный код формы DopRab.cs:*

*using System;*

*using System.Collections.Generic;*

*using System.ComponentModel;*

*using System.Data;*

*using System.Drawing;*

*using System.Linq;*

*using System.Text;*

*using System.Windows.Forms;*

*namespace diplom*

*{*

*public partial class DopRab : Form*

*{*

*public DopRab()*

*{*

*InitializeComponent();*

*}*

*private void DopRab\_Load(object sender, EventArgs e)*

*{*

*textBox1.DataBindings.Add(new Binding("Text", Program.general.dopRab4BS, "uchir", true));*

*textBox2.DataBindings.Add(new Binding("Text", Program.general.dopRab4BS, "napr", true));*

*textBox3.DataBindings.Add(new Binding("Text", Program.general.dopRab4BS, "oby", true));*

*textBox4.DataBindings.Add(new Binding("Text", Program.general.dopRab4BS, "rez", true));*

*pictureBox1.DataBindings.Add(new Binding("Image", Program.general.dopRab4BS, "foto", true));*

*}*

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

*{*

*Program.general.Validate();*

*Program.general.dopRab4BS.EndEdit();*

*Program.general.dopRab4TA.Update(Program.general.portfolioDS.dopRab4);*

*Close();*

*Dispose();*

*}*

*private void button4\_Click(object sender, EventArgs e)*

*{*

*Program.general.dopRab4BS.CancelEdit();*

*Close();*

*Dispose();*

*Program.general.dopRab4BS.Position = Program.general.pos;*

*}*

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

*{*

*if (openFileDialog1.ShowDialog() == DialogResult.OK)*

*{*

*Bitmap pict = new Bitmap(openFileDialog1.FileName);*

*pictureBox1.Image = (Image)pict;*

*(Program.general.dopRab4BS.Current as*

*DataRowView)["fotoF"] = openFileDialog1.FileName.*

*Substring(openFileDialog1.FileName.LastIndexOf('.'));*

*}*

*}*

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

*{*

*pictureBox1.Image = null;*

*}*

*}*

*}*

*Исходный код формы NauRab:*

*using System;*

*using System.Collections.Generic;*

*using System.ComponentModel;*

*using System.Data;*

*using System.Drawing;*

*using System.Linq;*

*using System.Text;*

*using System.Windows.Forms;*

*namespace diplom*

*{*

*public partial class NauRab : Form*

*{*

*public NauRab()*

*{*

*InitializeComponent();*

*}*

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

*{*

*if (checkBox1.Checked)*

*{*

*dateTimePicker2.Value = Convert.ToDateTime("01.01.2000 0:00:00");*

*}*

*Program.general.Validate();*

*Program.general.nauRab2BS.EndEdit();*

*Program.general.nauRab2TA.Update(Program.general.portfolioDS.nauRab2);*

*Close();*

*Dispose();*

*}*

*private void button4\_Click(object sender, EventArgs e)*

*{*

*Program.general.nauRab2BS.CancelEdit();*

*Close();*

*Dispose();*

*Program.general.nauRab2BS.Position = Program.general.pos;*

*}*

*private void NauRab\_Load(object sender, EventArgs e)*

*{*

*textBox1.DataBindings.Add(new Binding("Text", Program.general.nauRab2BS, "tip", true));*

*textBox2.DataBindings.Add(new Binding("Text", Program.general.nauRab2BS, "uroven", true));*

*textBox3.DataBindings.Add(new Binding("Text", Program.general.nauRab2BS, "tema", true));*

*textBox4.DataBindings.Add(new Binding("Text", Program.general.nauRab2BS, "forma", true));*

*textBox5.DataBindings.Add(new Binding("Text", Program.general.nauRab2BS, "mestoProv", true));*

*textBox6.DataBindings.Add(new Binding("Text", Program.general.nauRab2BS, "rez", true));*

*pictureBox1.DataBindings.Add(new Binding("Image", Program.general.nauRab2BS, "foto", true));*

*dateTimePicker1.DataBindings.Add(new Binding("Value", Program.general.nauRab2BS, "dateOt", true));*

*dateTimePicker2.DataBindings.Add(new Binding("Value", Program.general.nauRab2BS, "dateDo", true));*

*if ((Program.general.nauRab2BS.Current as DataRowView)["dateDo"].ToString() == "01.01.2000 0:00:00"*

*||*

*(Program.general.nauRab2BS.Current as DataRowView)["dateDo"].ToString() ==""*

*)*

*{*

*dateTimePicker2.Enabled = false;*

*checkBox1.Checked = true;*

*}*

*else*

*{*

*dateTimePicker2.Enabled = true;*

*checkBox1.Checked = false;*

*}*

*}*

*private void checkBox1\_CheckedChanged(object sender, EventArgs e)*

*{*

*if (checkBox1.Checked)*

*{*

*dateTimePicker2.Enabled = false;*

*dateTimePicker2.Value = Convert.ToDateTime("01.01.2000 0:00:00");*

*}*

*if (!checkBox1.Checked)*

*{*

*dateTimePicker2.Enabled = true;*

*dateTimePicker2.Value = DateTime.Now;*

*}*

*}*

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

*{*

*if (openFileDialog1.ShowDialog() == DialogResult.OK)*

*{*

*Bitmap pict = new Bitmap(openFileDialog1.FileName);*

*pictureBox1.Image = (Image)pict;*

*(Program.general.nauRab2BS.Current as DataRowView)["fotoF"] = openFileDialog1.FileName.Substring(openFileDialog1.FileName.LastIndexOf('.'));*

*}*

*}*

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

*{*

*pictureBox1.Image = null;*

*}*

*}*

*}*

*Исходный код формы VneRab:*

*using System;*

*using System.Collections.Generic;*

*using System.ComponentModel;*

*using System.Data;*

*using System.Drawing;*

*using System.Linq;*

*using System.Text;*

*using System.Windows.Forms;*

*namespace diplom*

*{*

*public partial class VneRab : Form*

*{*

*public VneRab()*

*{*

*InitializeComponent();*

*}*

*private void VneRab\_Load(object sender, EventArgs e)*

*{*

*textBox1.DataBindings.Add(new Binding("Text", Program.general.vneRab3BS, "naz", true));*

*textBox2.DataBindings.Add(new Binding("Text", Program.general.vneRab3BS, "forma", true));*

*textBox3.DataBindings.Add(new Binding("Text", Program.general.vneRab3BS, "rez", true));*

*pictureBox1.DataBindings.Add(new Binding("Image", Program.general.vneRab3BS, "foto", true));*

*dateTimePicker1.DataBindings.Add(new Binding("Value", Program.general.vneRab3BS, "dateOt", true));*

*dateTimePicker2.DataBindings.Add(new Binding("Value", Program.general.vneRab3BS, "dateDo", true));*

*if ((Program.general.vneRab3BS.Current as DataRowView)["dateDo"].ToString() == "01.01.2000 0:00:00"*

*||*

*(Program.general.vneRab3BS.Current as DataRowView)["dateDo"].ToString() == ""*

*)*

*{*

*dateTimePicker2.Enabled = false;*

*checkBox1.Checked = true;*

*}*

*else*

*{*

*dateTimePicker2.Enabled = true;*

*checkBox1.Checked = false;*

*}*

*}*

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

*{*

*if (checkBox1.Checked)*

*{*

*dateTimePicker2.Value = Convert.ToDateTime("01.01.2000 0:00:00");*

*}*

*Program.general.Validate();*

*Program.general.vneRab3BS.EndEdit();*

*Program.general.vneRab3TA.Update(Program.general.portfolioDS.vneRab3);*

*Close();*

*Dispose();*

*}*

*private void button4\_Click(object sender, EventArgs e)*

*{*

*Program.general.vneRab3BS.CancelEdit();*

*Close();*

*Dispose();*

*Program.general.vneRab3BS.Position = Program.general.pos;*

*}*

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

*{*

*if (openFileDialog1.ShowDialog() == DialogResult.OK)*

*{*

*Bitmap pict = new Bitmap(openFileDialog1.FileName);*

*pictureBox1.Image = (Image)pict;*

*(Program.general.vneRab3BS.Current as DataRowView)["fotoF"] = openFileDialog1.FileName.Substring(openFileDialog1.FileName.LastIndexOf('.'));*

*}*

*}*

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

*{*

*pictureBox1.Image = null;*

*}*

*private void checkBox1\_CheckedChanged(object sender, EventArgs e)*

*{*

*if (checkBox1.Checked)*

*{*

*dateTimePicker2.Enabled = false;*

*dateTimePicker2.Value = Convert.ToDateTime("01.01.2000 0:00:00");*

*}*

*if (!checkBox1.Checked)*

*{*

*dateTimePicker2.Enabled = true;*

*dateTimePicker2.Value = DateTime.Now;*

*}*

*}*

*}*

*}*

*Исходный код формы General:*

*using System;*

*using System.Collections.Generic;*

*using System.ComponentModel;*

*using System.Data;*

*using System.Drawing;*

*using System.Linq;*

*using System.Text;*

*using System.Windows.Forms;*

*using Word = Microsoft.Office.Interop.Word;*

*using System.Reflection;*

*using System.Runtime.InteropServices;*

*using System.IO;*

*namespace diplom*

*{*

*public partial class General : Form*

*{*

*public int pos = 0;*

*Word.Application w;*

*public General()*

*{*

*InitializeComponent();*

*}*

*class PrPrakt*

*{*

*public string vid, dolj, dateOt, dateDo;*

*public PrPrakt(string vid = "x", string dolj = "x", string dateOt = "x", string dateDo = "x")*

*{*

*this.vid = vid;*

*this.dolj = dolj;*

*this.dateOt = dateOt;*

*this.dateDo = dateDo;*

*}*

*}*

*public void addPraktStud()*

*{*

*bool studHavepredm = false;*

*foreach (DataRowView predmGrup in prPraktBS)*

*{*

*studHavepredm = false;*

*if(studBS.Count != 0)*

*if (praktBS.Count == 0)*

*{*

*praktBS.AddNew();*

*((DataRowView)praktBS.Current)["vid"] = ((DataRowView)prPraktBS.Current)["vid"];*

*((DataRowView)praktBS.Current)["dolj"] = ((DataRowView)prPraktBS.Current)["dolj"];*

*((DataRowView)praktBS.Current)["dateOt"] = ((DataRowView)prPraktBS.Current)["dateOt"];*

*((DataRowView)praktBS.Current)["dateDo"] = ((DataRowView)prPraktBS.Current)["dateDo"];*

*praktBS.EndEdit();*

*}*

*foreach (DataRowView predmStud in praktBS)*

*{*

*if (predmGrup["vid"].ToString() == predmStud["vid"].ToString() && predmStud["idStud"].ToString() == (studBS.Current as DataRowView)["id"].ToString())*

*{*

*studHavepredm = true;*

*break;*

*}*

*}*

*if (studBS.Count != 0)*

*if (!studHavepredm)*

*{*

*praktBS.AddNew();*

*((DataRowView)praktBS.Current)["vid"] = predmGrup["vid"];*

*((DataRowView)praktBS.Current)["dolj"] = predmGrup["dolj"];*

*((DataRowView)praktBS.Current)["dateOt"] = predmGrup["dateOt"];*

*((DataRowView)praktBS.Current)["dateDo"] = predmGrup["dateDo"];*

*praktBS.EndEdit();*

*}*

*}*

*if(praktBS.Count !=0)*

*praktTA.Update(portfolioDS.prakt);*

*}*

*public void addDisciplPrakt()*

*{*

*List<PrPrakt> pr = new List<PrPrakt>();*

*foreach (DataRowView row in prPraktBS)*

*{*

*pr.Add(new PrPrakt(row["vid"].ToString(), row["dolj"].ToString(), row["dateOt"].ToString(), row["dateDo"].ToString()));*

*}*

*if (studBS.Count != 0)*

*if (praktBS.Count == 0)*

*{*

*praktBS.AddNew();*

*((DataRowView)praktBS.Current)["vid"] = ((DataRowView)prPraktBS.Current)["vid"];*

*((DataRowView)praktBS.Current)["dolj"] = ((DataRowView)prPraktBS.Current)["dolj"];*

*((DataRowView)praktBS.Current)["dateOt"] = ((DataRowView)prPraktBS.Current)["dateOt"];*

*((DataRowView)praktBS.Current)["dateDo"] = ((DataRowView)prPraktBS.Current)["dateDo"];*

*praktBS.EndEdit();*

*}*

*bool have = true;*

*foreach (DataRowView stud in studBS)*

*{*

*foreach (PrPrakt predm in pr)*

*{*

*foreach (DataRowView rez in praktAllBS)*

*{*

*if (rez["idStud"].ToString() == stud["id"].ToString())*

*{*

*if (rez["vid"].ToString() == predm.vid)*

*{*

*have = true;*

*break;*

*}*

*have = false;*

*}*

*}*

*if (!have)*

*{*

*praktAllBS.AddNew();*

*((DataRowView)praktAllBS.Current)["vid"] = predm.vid;*

*((DataRowView)praktAllBS.Current)["dolj"] = predm.dolj;*

*if (predm.dateOt != "")*

*((DataRowView)praktAllBS.Current)["dateOt"] = Convert.ToDateTime(predm.dateOt);*

*if (predm.dateOt != "")*

*((DataRowView)praktAllBS.Current)["dateDo"] = Convert.ToDateTime(predm.dateDo);*

*((DataRowView)praktAllBS.Current)["idStud"] = stud["id"].ToString();*

*praktAllBS.EndEdit();*

*}*

*}*

*}*

*if (praktAllBS.Count != 0)*

*praktTA.Update(portfolioDS.prakt);*

*}*

*public void delDisciplPrakt()*

*{*

*if (studBS.Count != 0)*

*if (praktBS.Count == 0)*

*{*

*praktBS.AddNew();*

*((DataRowView)praktBS.Current)["vid"] = ((DataRowView)prPraktBS.Current)["vid"];*

*((DataRowView)praktBS.Current)["dolj"] = ((DataRowView)prPraktBS.Current)["dolj"];*

*((DataRowView)praktBS.Current)["dateOt"] = ((DataRowView)prPraktBS.Current)["dateOt"];*

*((DataRowView)praktBS.Current)["dateDo"] = ((DataRowView)prPraktBS.Current)["dateDo"];*

*praktBS.EndEdit();*

*}*

*List<PrPrakt> pr = new List<PrPrakt>();*

*foreach (DataRowView row in prPraktBS)*

*{*

*pr.Add(new PrPrakt(row["vid"].ToString(), row["dolj"].ToString(), row["dateOt"].ToString(), row["dateDo"].ToString()));*

*}*

*string id = "";*

*bool exist = false;*

*foreach (DataRowView stud in studBS)*

*{*

*foreach (DataRowView rez in praktAllBS)*

*{*

*foreach (PrPrakt predm in pr)*

*{*

*if (rez["idStud"].ToString() == stud["id"].ToString())*

*{*

*if (rez["vid"].ToString() == predm.vid)*

*{*

*exist = true;*

*break;*

*}*

*else*

*{*

*exist = false;*

*}*

*}*

*}*

*if (rez["idStud"].ToString() == stud["id"].ToString())*

*if (!exist)*

*{*

*rez.Delete();*

*exist = true;*

*}*

*}*

*}*

*if (praktAllBS.Count != 0)*

*praktTA.Update(portfolioDS.prakt);*

*}*

*class PrKurs*

*{*

*public string disc, sem;*

*public PrKurs(string disc = "x", string sem = "x")*

*{*

*this.disc = disc;*

*this.sem = sem;*

*}*

*}*

*public void addKursStud()*

*{*

*bool studHavepredm = false;*

*foreach (DataRowView predmGrup in prKursBS)*

*{*

*studHavepredm = false;*

*if (studBS.Count != 0)*

*if (kursBS.Count == 0)*

*{*

*kursBS.AddNew();*

*((DataRowView)kursBS.Current)["disc"] = ((DataRowView)prKursBS.Current)["disc"];*

*((DataRowView)kursBS.Current)["sem"] = ((DataRowView)prKursBS.Current)["sem"];*

*kursBS.EndEdit();*

*}*

*foreach (DataRowView predmStud in kursBS)*

*{*

*if (predmGrup["disc"].ToString() == predmStud["disc"].ToString() && predmStud["idStud"].ToString() == (studBS.Current as DataRowView)["id"].ToString())*

*{*

*studHavepredm = true;*

*break;*

*}*

*}*

*if (kursBS.Count != 0)*

*if (!studHavepredm)*

*{*

*kursBS.AddNew();*

*((DataRowView)kursBS.Current)["disc"] = predmGrup["disc"];*

*((DataRowView)kursBS.Current)["sem"] = predmGrup["sem"];*

*kursBS.EndEdit();*

*}*

*}*

*if (kursBS.Count != 0)*

*kursTA.Update(portfolioDS.kurs);*

*}*

*public void addDisciplKurs()*

*{*

*if (studBS.Count != 0)*

*if (kursBS.Count == 0)*

*{*

*kursBS.AddNew();*

*((DataRowView)kursBS.Current)["disc"] = ((DataRowView)prKursBS.Current)["disc"];*

*((DataRowView)kursBS.Current)["sem"] = ((DataRowView)prKursBS.Current)["sem"];*

*kursBS.EndEdit();*

*}*

*List<PrKurs> pr = new List<PrKurs>();*

*foreach (DataRowView row in prKursBS)*

*{*

*pr.Add(new PrKurs(row["disc"].ToString(), row["sem"].ToString()));*

*}*

*bool have = true;*

*foreach (DataRowView stud in studBS)*

*{*

*foreach (PrKurs predm in pr)*

*{*

*foreach (DataRowView rez in kursAllBS)*

*{*

*if (rez["idStud"].ToString() == stud["id"].ToString())*

*{*

*if (rez["disc"].ToString() == predm.disc)*

*{*

*have = true;*

*break;*

*}*

*have = false;*

*}*

*}*

*if (!have)*

*{*

*kursAllBS.AddNew();*

*((DataRowView)kursAllBS.Current)["disc"] = predm.disc;*

*((DataRowView)kursAllBS.Current)["sem"] = predm.sem;*

*((DataRowView)kursAllBS.Current)["idStud"] = stud["id"].ToString();*

*kursAllBS.EndEdit();*

*}*

*}*

*}*

*if (kursAllBS.Count != 0)*

*kursTA.Update(portfolioDS.kurs);*

*}*

*public void delDisciplKurs()*

*{*

*if (studBS.Count != 0)*

*if (kursBS.Count == 0)*

*{*

*kursBS.AddNew();*

*((DataRowView)kursBS.Current)["disc"] = ((DataRowView)prKursBS.Current)["disc"];*

*((DataRowView)kursBS.Current)["sem"] = ((DataRowView)prKursBS.Current)["sem"];*

*kursBS.EndEdit();*

*}*

*List<PrKurs> pr = new List<PrKurs>();*

*foreach (DataRowView row in prKursBS)*

*{*

*pr.Add(new PrKurs(row["disc"].ToString(), row["sem"].ToString()));*

*}*

*bool exist = false;*

*foreach (DataRowView stud in studBS)*

*{*

*foreach (DataRowView rez in kursAllBS)*

*{*

*foreach (PrKurs predm in pr)*

*{*

*if (rez["idStud"].ToString() == stud["id"].ToString())*

*{*

*if (rez["disc"].ToString() == predm.disc)*

*{*

*exist = true;*

*break;*

*}*

*else*

*{*

*exist = false;*

*}*

*}*

*}*

*if (rez["idStud"].ToString() == stud["id"].ToString())*

*if (!exist)*

*{*

*rez.Delete();*

*exist = true;*

*}*

*}*

*}*

*if (kursAllBS.Count != 0)*

*kursTA.Update(portfolioDS.kurs);*

*}*

*class PrDisc*

*{*

*public string disc, nagr,forma,sem;*

*public PrDisc(string disc = "x", string nagr = "x", string forma = "x", string sem = "x")*

*{*

*this.disc = disc;*

*this.nagr = nagr;*

*this.forma = forma;*

*this.sem = sem;*

*}*

*}*

*public void addDisciplStud()*

*{*

*bool studHavepredm = false;*

*foreach (DataRowView predmGrup in prDiscBS)*

*{*

*studHavepredm = false;*

*if (studBS.Count != 0)*

*if (discBS.Count == 0)*

*{*

*discBS.AddNew();*

*((DataRowView)discBS.Current)["disc"] = ((DataRowView)prDiscBS.Current)["disc"];*

*((DataRowView)discBS.Current)["nagr"] = ((DataRowView)prDiscBS.Current)["nagr"];*

*((DataRowView)discBS.Current)["forma"] = ((DataRowView)prDiscBS.Current)["forma"];*

*((DataRowView)discBS.Current)["sem"] = ((DataRowView)prDiscBS.Current)["sem"];*

*discBS.EndEdit();*

*}*

*foreach (DataRowView predmStud in discBS)*

*{*

*if (predmGrup["disc"].ToString() == predmStud["disc"].ToString() &&*

*predmGrup["sem"].ToString() == predmStud["sem"].ToString() &&*

*predmStud["idStud"].ToString() == (studBS.Current as DataRowView)["id"].ToString())*

*{*

*studHavepredm = true;*

*break;*

*}*

*}*

*if (studBS.Count != 0)*

*if (!studHavepredm)*

*{*

*discBS.AddNew();*

*((DataRowView)discBS.Current)["disc"] = predmGrup["disc"];*

*((DataRowView)discBS.Current)["nagr"] = predmGrup["nagr"];*

*((DataRowView)discBS.Current)["forma"] = predmGrup["forma"];*

*((DataRowView)discBS.Current)["sem"] = predmGrup["sem"];*

*discBS.EndEdit();*

*}*

*}*

*if (discBS.Count != 0)*

*discTA.Update(portfolioDS.disc);*

*}*

*public void addDisciplDisc()*

*{*

*if (studBS.Count != 0)*

*if (discBS.Count == 0)*

*{*

*discBS.AddNew();*

*((DataRowView)discBS.Current)["disc"] = ((DataRowView)prDiscBS.Current)["disc"];*

*((DataRowView)discBS.Current)["nagr"] = ((DataRowView)prDiscBS.Current)["nagr"];*

*((DataRowView)discBS.Current)["forma"] = ((DataRowView)prDiscBS.Current)["forma"];*

*((DataRowView)discBS.Current)["sem"] = ((DataRowView)prDiscBS.Current)["sem"];*

*discBS.EndEdit();*

*}*

*List<PrDisc> pr = new List<PrDisc>();*

*foreach (DataRowView row in prDiscBS)*

*{*

*pr.Add(new PrDisc(row["disc"].ToString(), row["nagr"].ToString(), row["forma"].ToString(), row["sem"].ToString()));*

*}*

*bool have = true;*

*foreach (DataRowView stud in studBS)*

*{*

*foreach (PrDisc predm in pr)*

*{*

*foreach (DataRowView rez in discAllBS)*

*{*

*if (rez["idStud"].ToString() == stud["id"].ToString())*

*{*

*if (rez["disc"].ToString() == predm.disc && rez["sem"].ToString() == predm.sem)*

*{*

*have = true;*

*break;*

*}*

*have = false;*

*}*

*}*

*if (!have)*

*{*

*discAllBS.AddNew();*

*((DataRowView)discAllBS.Current)["disc"] = predm.disc;*

*((DataRowView)discAllBS.Current)["nagr"] = predm.nagr;*

*((DataRowView)discAllBS.Current)["sem"] = predm.sem;*

*((DataRowView)discAllBS.Current)["forma"] = predm.forma;*

*((DataRowView)discAllBS.Current)["idStud"] = stud["id"].ToString();*

*discAllBS.EndEdit();*

*}*

*}*

*}*

*if (discAllBS.Count != 0)*

*discTA.Update(portfolioDS.disc);*

*}*

*public void delDisciplDisc()*

*{*

*if (studBS.Count != 0)*

*if (discBS.Count == 0)*

*{*

*discBS.AddNew();*

*((DataRowView)discBS.Current)["disc"] = ((DataRowView)prDiscBS.Current)["disc"];*

*((DataRowView)discBS.Current)["nagr"] = ((DataRowView)prDiscBS.Current)["nagr"];*

*((DataRowView)discBS.Current)["forma"] = ((DataRowView)prDiscBS.Current)["forma"];*

*((DataRowView)discBS.Current)["sem"] = ((DataRowView)prDiscBS.Current)["sem"];*

*discBS.EndEdit();*

*}*

*List<PrDisc> pr = new List<PrDisc>();*

*foreach (DataRowView row in prDiscBS)*

*{*

*pr.Add(new PrDisc(row["disc"].ToString(), row["nagr"].ToString(), row["forma"].ToString(), row["sem"].ToString()));*

*}*

*string id = "";*

*bool exist = false;*

*foreach (DataRowView stud in studBS)*

*{*

*foreach (DataRowView rez in discAllBS)*

*{*

*foreach (PrDisc predm in pr)*

*{*

*if (rez["idStud"].ToString() == stud["id"].ToString())*

*{*

*if (rez["disc"].ToString() == predm.disc && rez["sem"].ToString() == predm.sem)*

*{*

*exist = true;*

*break;*

*}*

*else*

*{*

*exist = false;*

*}*

*}*

*}*

*if (rez["idStud"].ToString() == stud["id"].ToString())*

*if (!exist)*

*{*

*rez.Delete();*

*exist = true;*

*}*

*}*

*}*

*if (discAllBS.Count != 0)*

*discTA.Update(portfolioDS.disc);*

*}*

*private void show(string nameFolder,DataRowView curGroup,DataRowView curStud)*

*{*

*try*

*{*

*w = Marshal.GetActiveObject("Word.Application") as Word.Application;*

*}*

*catch (COMException err)*

*{*

*w = new Word.Application();*

*}*

*string s = "";*

*int k = 0;*

*int i = 0;*

*progressBar1.Value = 0;*

*progressBar1.Maximum = prDiscBS.Count + 1;*

*progressBar1.Visible = true;*

*w.Documents.Add(Application.StartupPath + "\\Templates\\anketa.dot");*

*w.ActiveDocument.Bookmarks["fam"].Range.Text = curStud["fam"].ToString();*

*w.ActiveDocument.Bookmarks["name"].Range.Text = curStud["name"].ToString();*

*w.ActiveDocument.Bookmarks["otch"].Range.Text = curStud["otch"].ToString();*

*w.ActiveDocument.Bookmarks["spec"].Range.Text = (specBS.Current as DataRowView)["naz"].ToString();*

*w.ActiveDocument.Bookmarks["nomZachisl"].Range.Text = curStud["nomZachisl"].ToString();*

*w.ActiveDocument.Bookmarks["formO"].Range.Text = curGroup["formO"].ToString();*

*w.ActiveDocument.Bookmarks["group"].Range.Text = curGroup["naz"].ToString();*

*w.ActiveDocument.Bookmarks["tipJaz"].Range.Text = curStud["urVlad"].ToString();*

*w.ActiveDocument.Bookmarks["tel"].Range.Text = curStud["tel"].ToString();*

*w.ActiveDocument.Bookmarks["email"].Range.Text = curStud["email"].ToString();*

*w.ActiveDocument.Bookmarks["fioLast"].Range.Text = curStud["fam"].ToString() + " " + curStud["name"].ToString() + " " + curStud["otch"].ToString();*

*if (curStud["dateR"].ToString() != "")*

*w.ActiveDocument.Bookmarks["dateR"].Range.Text = Convert.ToDateTime(curStud["dateR"]).ToString("dd/MM/yyyy");*

*if (curStud["dateZachisl"].ToString() != "")*

*w.ActiveDocument.Bookmarks["dateZachisl"].Range.Text = Convert.ToDateTime(curStud["dateZachisl"]).ToString("dd/MM/yyyy");*

*Byte[] blob = null;*

*MemoryStream memStream = null;*

*Image img = null;*

*if (curStud["foto"].ToString() != "")*

*{*

*blob = (byte[])curStud["foto"];*

*memStream = new MemoryStream(blob);*

*memStream.Write(blob, 0, blob.Length);*

*memStream.Position = 0;*

*img = Image.FromStream(memStream);*

*Clipboard.SetImage(img);*

*w.ActiveDocument.Bookmarks["foto"].Range.Paste();*

*w.ActiveDocument.InlineShapes[1].Width = 90;*

*w.ActiveDocument.InlineShapes[1].Height = 120;*

*}*

*foreach (DataRowView row in jazBS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString())*

*{*

*s += row["nameJaz"].ToString() + ", ";*

*}*

*}*

*if (s != "")*

*{*

*s = s.Remove(s.Length - 2);*

*}*

*w.ActiveDocument.Bookmarks["nameJaz"].Range.Text = s;*

*/////////////////////////////////////////*

*k = 2;*

*foreach (DataRowView row in prikazPerevodBS)*

*{*

*if (row["idGroup"].ToString() == curGroup["id"].ToString())*

*{*

*w.ActiveDocument.Tables[3].Rows.Add().SetHeight(0, 0);*

*w.ActiveDocument.Tables[3].Rows[k].Cells[1].Range.Text = row["kurs"].ToString();*

*if (row["date"].ToString() != "")*

*w.ActiveDocument.Tables[3].Rows[k].Cells[2].Range.Text = row["nomer"].ToString() + " от "+ Convert.ToDateTime(row["date"]).ToString("dd.MM.yyyy");*

*w.ActiveDocument.Tables[3].Rows[k].Cells[3].Range.Text = row["soderj"].ToString();*

*k++;*

*}*

*}*

*/////////////////////////////////////////*

*k = 2;*

*foreach (DataRowView row in prikazStudBS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString() && row["tip"].ToString() == "Перерыв, в академическом отпуске")*

*{*

*w.ActiveDocument.Tables[4].Rows.Add().SetHeight(0, 0);*

*w.ActiveDocument.Tables[4].Rows[k].Cells[1].Range.Text = row["kurs"].ToString();*

*if (row["date"].ToString() != "")*

*w.ActiveDocument.Tables[4].Rows[k].Cells[2].Range.Text = row["nomer"].ToString() + " от " + Convert.ToDateTime(row["date"]).ToString("dd.MM.yyyy");*

*w.ActiveDocument.Tables[4].Rows[k].Cells[3].Range.Text = row["soderj"].ToString();*

*k++;*

*}*

*}*

*if (w.ActiveDocument.Tables[4].Rows.Count==1)*

*{*

*w.ActiveDocument.Tables[4].Rows.Add().SetHeight(0, 0);*

*w.ActiveDocument.Tables[4].Rows[k].Cells[1].Range.Text = "-";*

*w.ActiveDocument.Tables[4].Rows[k].Cells[2].Range.Text = "-";*

*w.ActiveDocument.Tables[4].Rows[k].Cells[3].Range.Text = "-";*

*}*

*/////////////////////////////////////////*

*k = 2;*

*foreach (DataRowView row in prikazStudBS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString() && row["tip"].ToString() == "Поощрения")*

*{*

*w.ActiveDocument.Tables[5].Rows.Add().SetHeight(0, 0);*

*w.ActiveDocument.Tables[5].Rows[k].Cells[1].Range.Text = row["kurs"].ToString();*

*if (row["date"].ToString() != "")*

*w.ActiveDocument.Tables[5].Rows[k].Cells[2].Range.Text = row["nomer"].ToString() + " от " + Convert.ToDateTime(row["date"]).ToString("dd.MM.yyyy");*

*w.ActiveDocument.Tables[5].Rows[k].Cells[3].Range.Text = row["soderj"].ToString();*

*k++;*

*}*

*}*

*if (w.ActiveDocument.Tables[5].Rows.Count == 1)*

*{*

*w.ActiveDocument.Tables[5].Rows.Add().SetHeight(0, 0);*

*w.ActiveDocument.Tables[5].Rows[k].Cells[1].Range.Text = "-";*

*w.ActiveDocument.Tables[5].Rows[k].Cells[2].Range.Text = "-";*

*w.ActiveDocument.Tables[5].Rows[k].Cells[3].Range.Text = "-";*

*}*

*/////////////////////////////////////////*

*k = 2;*

*foreach (DataRowView row in prikazStudBS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString() && row["tip"].ToString() == "Взыскания")*

*{*

*w.ActiveDocument.Tables[6].Rows.Add().SetHeight(0, 0);*

*w.ActiveDocument.Tables[6].Rows[k].Cells[1].Range.Text = row["kurs"].ToString();*

*if (row["date"].ToString() != "")*

*w.ActiveDocument.Tables[6].Rows[k].Cells[2].Range.Text = row["nomer"].ToString() + " от " + Convert.ToDateTime(row["date"]).ToString("dd.MM.yyyy");*

*w.ActiveDocument.Tables[6].Rows[k].Cells[3].Range.Text = row["soderj"].ToString();*

*k++;*

*}*

*}*

*if (w.ActiveDocument.Tables[6].Rows.Count == 1)*

*{*

*w.ActiveDocument.Tables[6].Rows.Add().SetHeight(0, 0);*

*w.ActiveDocument.Tables[6].Rows[k].Cells[1].Range.Text = "-";*

*w.ActiveDocument.Tables[6].Rows[k].Cells[2].Range.Text = "-";*

*w.ActiveDocument.Tables[6].Rows[k].Cells[3].Range.Text = "-";*

*}*

*/////////////////////////////////////////*

*k = 2;*

*foreach (DataRowView row in prevObrBS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString())*

*{*

*w.ActiveDocument.Tables[7].Rows.Add().SetHeight(0, 0);*

*if (row["dateOt"].ToString() != "" && row["dateDo"].ToString() != "")*

*w.ActiveDocument.Tables[7].Rows[k].Cells[1].Range.Text =*

*Convert.ToDateTime(row["dateOt"]).ToString("dd.MM.yyyy")*

*+ " - " +*

*Convert.ToDateTime(row["dateDo"]).ToString("dd.MM.yyyy");*

*w.ActiveDocument.Tables[7].Rows[k].Cells[2].Range.Text = row["ucher"].ToString();*

*w.ActiveDocument.Tables[7].Rows[k].Cells[3].Range.Text = row["kval"].ToString();*

*k++;*

*}*

*}*

*/////////////////////////////////////////*

*k = 2;*

*foreach (DataRowView row in discAllBS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString() && row["sem"].ToString() == "1")*

*{*

*w.ActiveDocument.Tables[8].Rows.Add().SetHeight(0, 0);*

*w.ActiveDocument.Tables[8].Rows[k].Cells[1].Range.Text = (k - 1).ToString();*

*w.ActiveDocument.Tables[8].Rows[k].Cells[2].Range.Text = row["disc"].ToString();*

*w.ActiveDocument.Tables[8].Rows[k].Cells[3].Range.Text = row["nagr"].ToString();*

*w.ActiveDocument.Tables[8].Rows[k].Cells[4].Range.Text = row["forma"].ToString();*

*w.ActiveDocument.Tables[8].Rows[k].Cells[5].Range.Text = row["itog"].ToString();*

*k++;*

*progressBar1.Value++;*

*}*

*}*

*w.ActiveDocument.Tables[8].Rows[k].Delete();*

*/////////////////////////////////////////*

*k = 2;*

*foreach (DataRowView row in discAllBS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString() && row["sem"].ToString() == "2")*

*{*

*w.ActiveDocument.Tables[9].Rows.Add().SetHeight(0, 0);*

*w.ActiveDocument.Tables[9].Rows[k].Cells[1].Range.Text = (k - 1).ToString();*

*w.ActiveDocument.Tables[9].Rows[k].Cells[2].Range.Text = row["disc"].ToString();*

*w.ActiveDocument.Tables[9].Rows[k].Cells[3].Range.Text = row["nagr"].ToString();*

*w.ActiveDocument.Tables[9].Rows[k].Cells[4].Range.Text = row["forma"].ToString();*

*w.ActiveDocument.Tables[9].Rows[k].Cells[5].Range.Text = row["itog"].ToString();*

*k++;*

*progressBar1.Value++;*

*}*

*}*

*w.ActiveDocument.Tables[9].Rows[k].Delete();*

*/////////////////////////////////////////*

*k = 2;*

*foreach (DataRowView row in discAllBS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString() && row["sem"].ToString() == "3")*

*{*

*w.ActiveDocument.Tables[10].Rows.Add().SetHeight(0, 0);*

*w.ActiveDocument.Tables[10].Rows[k].Cells[1].Range.Text = (k - 1).ToString();*

*w.ActiveDocument.Tables[10].Rows[k].Cells[2].Range.Text = row["disc"].ToString();*

*w.ActiveDocument.Tables[10].Rows[k].Cells[3].Range.Text = row["nagr"].ToString();*

*w.ActiveDocument.Tables[10].Rows[k].Cells[4].Range.Text = row["forma"].ToString();*

*w.ActiveDocument.Tables[10].Rows[k].Cells[5].Range.Text = row["itog"].ToString();*

*k++;*

*progressBar1.Value++;*

*}*

*}*

*w.ActiveDocument.Tables[10].Rows[k].Delete();*

*/////////////////////////////////////////*

*k = 2;*

*foreach (DataRowView row in discAllBS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString() && row["sem"].ToString() == "4")*

*{*

*w.ActiveDocument.Tables[11].Rows.Add().SetHeight(0, 0);*

*w.ActiveDocument.Tables[11].Rows[k].Cells[1].Range.Text = (k - 1).ToString();*

*w.ActiveDocument.Tables[11].Rows[k].Cells[2].Range.Text = row["disc"].ToString();*

*w.ActiveDocument.Tables[11].Rows[k].Cells[3].Range.Text = row["nagr"].ToString();*

*w.ActiveDocument.Tables[11].Rows[k].Cells[4].Range.Text = row["forma"].ToString();*

*w.ActiveDocument.Tables[11].Rows[k].Cells[5].Range.Text = row["itog"].ToString();*

*k++;*

*progressBar1.Value++;*

*}*

*}*

*w.ActiveDocument.Tables[11].Rows[k].Delete();*

*/////////////////////////////////////////*

*k = 2;*

*foreach (DataRowView row in discAllBS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString() && row["sem"].ToString() == "5")*

*{*

*w.ActiveDocument.Tables[12].Rows.Add().SetHeight(0, 0);*

*w.ActiveDocument.Tables[12].Rows[k].Cells[1].Range.Text = (k - 1).ToString();*

*w.ActiveDocument.Tables[12].Rows[k].Cells[2].Range.Text = row["disc"].ToString();*

*w.ActiveDocument.Tables[12].Rows[k].Cells[3].Range.Text = row["nagr"].ToString();*

*w.ActiveDocument.Tables[12].Rows[k].Cells[4].Range.Text = row["forma"].ToString();*

*w.ActiveDocument.Tables[12].Rows[k].Cells[5].Range.Text = row["itog"].ToString();*

*k++;*

*progressBar1.Value++;*

*}*

*}*

*w.ActiveDocument.Tables[12].Rows[k].Delete();*

*/////////////////////////////////////////*

*k = 2;*

*foreach (DataRowView row in discAllBS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString() && row["sem"].ToString() == "6")*

*{*

*w.ActiveDocument.Tables[13].Rows.Add().SetHeight(0, 0);*

*w.ActiveDocument.Tables[13].Rows[k].Cells[1].Range.Text = (k - 1).ToString();*

*w.ActiveDocument.Tables[13].Rows[k].Cells[2].Range.Text = row["disc"].ToString();*

*w.ActiveDocument.Tables[13].Rows[k].Cells[3].Range.Text = row["nagr"].ToString();*

*w.ActiveDocument.Tables[13].Rows[k].Cells[4].Range.Text = row["forma"].ToString();*

*w.ActiveDocument.Tables[13].Rows[k].Cells[5].Range.Text = row["itog"].ToString();*

*k++;*

*progressBar1.Value++;*

*}*

*}*

*w.ActiveDocument.Tables[13].Rows[k].Delete();*

*/////////////////////////////////////////*

*k = 2;*

*foreach (DataRowView row in discAllBS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString() && row["sem"].ToString() == "7")*

*{*

*w.ActiveDocument.Tables[14].Rows.Add().SetHeight(0, 0);*

*w.ActiveDocument.Tables[14].Rows[k].Cells[1].Range.Text = (k - 1).ToString();*

*w.ActiveDocument.Tables[14].Rows[k].Cells[2].Range.Text = row["disc"].ToString();*

*w.ActiveDocument.Tables[14].Rows[k].Cells[3].Range.Text = row["nagr"].ToString();*

*w.ActiveDocument.Tables[14].Rows[k].Cells[4].Range.Text = row["forma"].ToString();*

*w.ActiveDocument.Tables[14].Rows[k].Cells[5].Range.Text = row["itog"].ToString();*

*k++;*

*progressBar1.Value++;*

*}*

*}*

*w.ActiveDocument.Tables[14].Rows[k].Delete();*

*/////////////////////////////////////////*

*k = 2;*

*foreach (DataRowView row in discAllBS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString() && row["sem"].ToString() == "8")*

*{*

*w.ActiveDocument.Tables[15].Rows.Add().SetHeight(0, 0);*

*w.ActiveDocument.Tables[15].Rows[k].Cells[1].Range.Text = (k - 1).ToString();*

*w.ActiveDocument.Tables[15].Rows[k].Cells[2].Range.Text = row["disc"].ToString();*

*w.ActiveDocument.Tables[15].Rows[k].Cells[3].Range.Text = row["nagr"].ToString();*

*w.ActiveDocument.Tables[15].Rows[k].Cells[4].Range.Text = row["forma"].ToString();*

*w.ActiveDocument.Tables[15].Rows[k].Cells[5].Range.Text = row["itog"].ToString();*

*k++;*

*progressBar1.Value++;*

*}*

*}*

*w.ActiveDocument.Tables[15].Rows[k].Delete();*

*/////////////////////////////////////////*

*k = 2;*

*foreach (DataRowView row in kursBS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString())*

*{*

*w.ActiveDocument.Tables[16].Rows.Add();*

*w.ActiveDocument.Tables[16].Rows[k].Cells[1].Range.Text = (k - 1).ToString();*

*w.ActiveDocument.Tables[16].Rows[k].Cells[2].Range.Text = row["disc"].ToString();*

*w.ActiveDocument.Tables[16].Rows[k].Cells[3].Range.Text = row["sem"].ToString();*

*w.ActiveDocument.Tables[16].Rows[k].Cells[3].Range.Text = row["tema"].ToString();*

*w.ActiveDocument.Tables[16].Rows[k].Cells[3].Range.Text = row["ocenka"].ToString();*

*k++;*

*}*

*}*

*w.ActiveDocument.Tables[16].Rows[k].Delete();*

*/////////////////////////////////////////*

*k = 2;*

*foreach (DataRowView row in praktBS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString())*

*{*

*w.ActiveDocument.Tables[17].Rows.Add();*

*w.ActiveDocument.Tables[17].Rows[k].Cells[1].Range.Text = (k - 1).ToString();*

*w.ActiveDocument.Tables[17].Rows[k].Cells[2].Range.Text = row["vid"].ToString();*

*w.ActiveDocument.Tables[17].Rows[k].Cells[3].Range.Text = row["baza"].ToString();*

*w.ActiveDocument.Tables[17].Rows[k].Cells[4].Range.Text = row["dolj"].ToString();*

*if (row["dateOt"].ToString() != "" && row["dateDo"].ToString() != "")*

*w.ActiveDocument.Tables[17].Rows[k].Cells[5].Range.Text =*

*Convert.ToDateTime(row["dateOt"]).ToString("dd.MM.yyyy")*

*+ " - " +*

*Convert.ToDateTime(row["dateDo"]).ToString("dd.MM.yyyy");*

*w.ActiveDocument.Tables[17].Rows[k].Cells[6].Range.Text = row["ocenka"].ToString();*

*k++;*

*}*

*}*

*w.ActiveDocument.Tables[17].Rows[k].Delete();*

*/////////////////////////////////////////*

*k = 2;*

*foreach (DataRowView row in kvalRabBS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString())*

*{*

*w.ActiveDocument.Tables[18].Rows.Add();*

*w.ActiveDocument.Tables[18].Rows[k].Cells[1].Range.Text = (k - 1).ToString();*

*w.ActiveDocument.Tables[18].Rows[k].Cells[2].Range.Text = row["tema"].ToString();*

*w.ActiveDocument.Tables[18].Rows[k].Cells[3].Range.Text = row["ocenka"].ToString();*

*k++;*

*}*

*}*

*w.ActiveDocument.Tables[18].Rows[k].Delete();*

*/////////////////////////////////////////*

*k = 2;*

*foreach (DataRowView row in nauRab2BS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString())*

*{*

*w.ActiveDocument.Tables[19].Rows.Add().SetHeight(0, 0);*

*w.ActiveDocument.Tables[19].Rows[k].Cells[1].Range.Text = (k - 1).ToString();*

*w.ActiveDocument.Tables[19].Rows[k].Cells[2].Range.Text = row["tip"].ToString();*

*w.ActiveDocument.Tables[19].Rows[k].Cells[3].Range.Text = row["uroven"].ToString();*

*w.ActiveDocument.Tables[19].Rows[k].Cells[4].Range.Text = row["tema"].ToString();*

*w.ActiveDocument.Tables[19].Rows[k].Cells[6].Range.Text = row["forma"].ToString();*

*w.ActiveDocument.Tables[19].Rows[k].Cells[7].Range.Text = row["rez"].ToString();*

*if (row["dateDo"].ToString() == "01.01.2000 0:00:00" || row["dateDo"].ToString() == "")*

*{*

*if (row["dateOt"].ToString() != "")*

*{*

*w.ActiveDocument.Tables[19].Rows[k].Cells[5].Range.Text =*

*Convert.ToDateTime(row["dateOt"]).ToString("dd.MM.yyyy")*

*+ " " +*

*row["mestoProv"].ToString();*

*}*

*}*

*else if (row["dateDo"].ToString() != "01.01.2000 0:00:00" && row["dateDo"].ToString() != "")*

*{*

*if (row["dateOt"].ToString() != "")*

*{*

*w.ActiveDocument.Tables[19].Rows[k].Cells[5].Range.Text =*

*Convert.ToDateTime(row["dateOt"]).ToString("dd.MM.yyyy")*

*+ " - " +*

*Convert.ToDateTime(row["dateDo"]).ToString("dd.MM.yyyy")*

*+ " " +*

*row["mestoProv"].ToString();*

*}*

*}*

*k++;*

*}*

*}*

*/////////////////////////////////////////*

*k = 2;*

*foreach (DataRowView row in vneRab3BS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString())*

*{*

*w.ActiveDocument.Tables[20].Rows.Add().SetHeight(0, 0);*

*w.ActiveDocument.Tables[20].Rows[k].Cells[1].Range.Text = (k - 1).ToString();*

*w.ActiveDocument.Tables[20].Rows[k].Cells[2].Range.Text = row["naz"].ToString();*

*w.ActiveDocument.Tables[20].Rows[k].Cells[3].Range.Text = row["forma"].ToString();*

*w.ActiveDocument.Tables[20].Rows[k].Cells[5].Range.Text = row["rez"].ToString();*

*if (row["dateDo"].ToString() == "01.01.2000 0:00:00" || row["dateDo"].ToString() == "")*

*{*

*if (row["dateOt"].ToString() != "")*

*{*

*w.ActiveDocument.Tables[20].Rows[k].Cells[4].Range.Text =*

*Convert.ToDateTime(row["dateOt"]).ToString("dd.MM.yyyy");*

*}*

*}*

*else if (row["dateDo"].ToString() != "01.01.2000 0:00:00" && row["dateDo"].ToString() != "")*

*{*

*if (row["dateOt"].ToString() != "")*

*{*

*w.ActiveDocument.Tables[20].Rows[k].Cells[4].Range.Text =*

*Convert.ToDateTime(row["dateOt"]).ToString("dd.MM.yyyy")*

*+ " - " +*

*Convert.ToDateTime(row["dateDo"]).ToString("dd.MM.yyyy");*

*}*

*}*

*k++;*

*}*

*}*

*/////////////////////////////////////////*

*k = 2;*

*foreach (DataRowView row in dopRab4BS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString())*

*{*

*w.ActiveDocument.Tables[21].Rows.Add().SetHeight(0, 0);*

*w.ActiveDocument.Tables[21].Rows[k].Cells[1].Range.Text = (k - 1).ToString();*

*w.ActiveDocument.Tables[21].Rows[k].Cells[2].Range.Text = row["uchir"].ToString();*

*w.ActiveDocument.Tables[21].Rows[k].Cells[3].Range.Text = row["napr"].ToString();*

*w.ActiveDocument.Tables[21].Rows[k].Cells[4].Range.Text = row["oby"].ToString();*

*w.ActiveDocument.Tables[21].Rows[k].Cells[5].Range.Text = row["rez"].ToString();*

*k++;*

*}*

*}*

*/////////////////////////////////////////*

*string FIO ="ст."+curStud["fam"].ToString()+curStud["name"].ToString()+ curStud["otch"].ToString();*

*if (Directory.Exists(nameFolder + "\\" + FIO))*

*{*

*Directory.Delete(nameFolder + "\\" + FIO,true);*

*}*

*Directory.CreateDirectory(nameFolder + "\\" + FIO);*

*w.ActiveDocument.SaveAs(nameFolder + "\\" + FIO + "\\" + FIO + ".doc");*

*string folderSavePriloj;*

*folderSavePriloj = nameFolder + "\\" + FIO + "\\Приложения\\Научно-исследовательская деятельность\\";*

*Directory.CreateDirectory(folderSavePriloj);*

*folderSavePriloj = nameFolder + "\\" + FIO + "\\Приложения\\Внеаудиторная деятельность\\";*

*Directory.CreateDirectory(folderSavePriloj);*

*folderSavePriloj = nameFolder + "\\" + FIO + "\\Приложения\\Дополнительное образование, самообразование\\";*

*Directory.CreateDirectory(folderSavePriloj);*

*progressBar1.Value++;*

*progressBar1.Visible = false;*

*w.ActiveDocument.Close();*

*w = null;*

*foreach (DataRowView row in nauRab2BS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString())*

*{*

*if (row["foto"].ToString() != "")*

*{*

*blob = (byte[])row["foto"];*

*memStream = new MemoryStream(blob);*

*memStream.Write(blob, 0, blob.Length);*

*memStream.Position = 0;*

*img = Image.FromStream(memStream);*

*img.Save(folderSavePriloj + row["rez"].ToString() + row["fotoF"].ToString(), System.Drawing.Imaging.ImageFormat.Jpeg);*

*}*

*}*

*}*

*foreach (DataRowView row in vneRab3BS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString())*

*{*

*if (row["foto"].ToString() != "")*

*{*

*blob = (byte[])row["foto"];*

*memStream = new MemoryStream(blob);*

*memStream.Write(blob, 0, blob.Length);*

*memStream.Position = 0;*

*img = Image.FromStream(memStream);*

*img.Save(folderSavePriloj + row["rez"].ToString() + row["fotoF"].ToString(), System.Drawing.Imaging.ImageFormat.Jpeg);*

*}*

*}*

*}*

*foreach (DataRowView row in dopRab4BS)*

*{*

*if (row["idStud"].ToString() == curStud["id"].ToString())*

*{*

*if (row["foto"].ToString() != "")*

*{*

*blob = (byte[])row["foto"];*

*memStream = new MemoryStream(blob);*

*memStream.Write(blob, 0, blob.Length);*

*memStream.Position = 0;*

*img = Image.FromStream(memStream);*

*img.Save(folderSavePriloj + row["rez"].ToString() + row["fotoF"].ToString(), System.Drawing.Imaging.ImageFormat.Jpeg);*

*}*

*}*

*}*

*}*

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

*{*

*try*

*{*

*tableAM.Connection.ConnectionString = "Data Source=" + Program.server + "," + Program.port + ";Initial Catalog=portfolio;Persist Security Info=True;User ID=" + Program.user + ";password=" + Program.password;*

*this.specTA.Fill(this.portfolioDS.spec);*

*this.groupTA.Fill(this.portfolioDS.group);*

*this.studTA.Fill(this.portfolioDS.stud);*

*this.prevObrTA.Fill(this.portfolioDS.prevObr);*

*this.jazTA.Fill(this.portfolioDS.jaz);*

*this.kvalRabTA.Fill(this.portfolioDS.kvalRab);*

*this.dopRab4TA.Fill(this.portfolioDS.dopRab4);*

*this.nauRab2TA.Fill(this.portfolioDS.nauRab2);*

*this.vneRab3TA.Fill(this.portfolioDS.vneRab3);*

*this.prikazPerevodTA.Fill(this.portfolioDS.prikazPerevod);*

*this.prikazStudTA.Fill(this.portfolioDS.prikazStud);*

*this.praktTA.Fill(this.portfolioDS.prakt);*

*this.prPraktTA.Fill(this.portfolioDS.prPrakt);*

*this.kursTA.Fill(this.portfolioDS.kurs);*

*this.prKursTA.Fill(this.portfolioDS.prKurs);*

*this.discTA.Fill(this.portfolioDS.disc);*

*this.prDiscTA.Fill(this.portfolioDS.prDisc);*

*Properties.Settings.Default.Save();*

*}*

*catch (Exception ex)*

*{*

*MessageBox.Show("Неверные идентификационные данные", "Ошибка входа", MessageBoxButtons.OK, MessageBoxIcon.Error);*

*Close();*

*}*

*label16.Text = "Количество студентов: "+studBS.Count.ToString();*

*discBS.Filter = "sem = '1'";*

*}*

*private void general\_KeyUp(object sender, KeyEventArgs e)*

*{*

*if (e.KeyCode == Keys.Return)*

*{*

*dopRab4BS\_CurrentChanged(sender, e);*

*specBS\_CurrentChanged(sender, e);*

*groupBS\_CurrentChanged(sender, e);*

*studBS\_CurrentChanged(sender, e);*

*jazBS\_CurrentChanged(sender, e);*

*prevObrBS\_CurrentChanged(sender, e);*

*nauRab2BS\_CurrentChanged(sender, e);*

*vneRab3BS\_CurrentChanged(sender, e);*

*prikazStudBS\_CurrentChanged(sender, e);*

*kvalRabBS\_CurrentChanged(sender, e);*

*discBS\_CurrentChanged(sender, e);*

*}*

*}*

*#region bt*

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

*{*

*specBS.AddNew();*

*button1.Enabled = false;*

*}*

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

*{*

*if (specBS.Count == 0)*

*{*

*MessageBox.Show("Список специальностей пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*if (MessageBox.Show(*

*(specBS.Current as DataRowView)["naz"].ToString(),*

*"Вы точно хотите удалить?", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)*

*{*

*specBS.RemoveCurrent();*

*specTA.Update(portfolioDS.spec);*

*}*

*}*

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

*{*

*if (specBS.Count==0)*

*{*

*MessageBox.Show("Список специальностей пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*groupBS.AddNew();*

*button3.Enabled = false;*

*}*

*private void button4\_Click(object sender, EventArgs e)*

*{*

*if (groupBS.Count == 0)*

*{*

*MessageBox.Show("Список групп пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*if (MessageBox.Show(*

*(groupBS.Current as DataRowView)["naz"].ToString(),*

*"Вы точно хотите удалить?", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)*

*{*

*groupBS.RemoveCurrent();*

*groupTA.Update(portfolioDS.group);*

*}*

*}*

*private void button5\_Click(object sender, EventArgs e)*

*{*

*if (groupBS.Count == 0)*

*{*

*MessageBox.Show("Список групп пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*Perevod perevod = new Perevod();*

*perevod.ShowDialog();*

*}*

*private void button6\_Click(object sender, EventArgs e)*

*{*

*if (groupBS.Count == 0)*

*{*

*MessageBox.Show("Список групп пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*Predm predm = new Predm();*

*predm.ShowDialog();*

*}*

*private void button7\_Click(object sender, EventArgs e)*

*{*

*if (groupBS.Count == 0)*

*{*

*MessageBox.Show("Список групп пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*pos = studBS.Position;*

*studBS.AddNew();*

*Profile profile = new Profile();*

*profile.ShowDialog();*

*}*

*private void button8\_Click(object sender, EventArgs e)*

*{*

*if (studBS.Count == 0)*

*{*

*MessageBox.Show("Список студентов пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*if (MessageBox.Show(*

*(studBS.Current as DataRowView)["fam"].ToString() + " " +*

*(studBS.Current as DataRowView)["name"].ToString() + " " +*

*(studBS.Current as DataRowView)["otch"].ToString(),*

*"Вы точно хотите удалить?", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)*

*{*

*studBS.RemoveCurrent();*

*studTA.Update(portfolioDS.stud);*

*}*

*}*

*private void button9\_Click(object sender, EventArgs e)*

*{*

*if (kvalRabBS.Count == 0)*

*{*

*MessageBox.Show("Список работ пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*if (MessageBox.Show(*

*(kvalRabBS.Current as DataRowView)["tema"].ToString(),*

*"Вы точно хотите удалить?", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)*

*{*

*kvalRabBS.RemoveCurrent();*

*kvalRabTA.Update(portfolioDS.kvalRab);*

*}*

*}*

*private void button10\_Click(object sender, EventArgs e)*

*{*

*if (studBS.Count == 0)*

*{*

*MessageBox.Show("Список студентов пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*kvalRabBS.AddNew();*

*button10.Enabled = false;*

*}*

*private void button11\_Click(object sender, EventArgs e)*

*{*

*if (studBS.Count == 0)*

*{*

*MessageBox.Show("Список студентов пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*jazBS.AddNew();*

*button11.Enabled = false;*

*}*

*private void button12\_Click(object sender, EventArgs e)*

*{*

*if (jazBS.Count == 0)*

*{*

*MessageBox.Show("Список языков пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*if (MessageBox.Show(*

*(jazBS.Current as DataRowView)["nameJaz"].ToString(),*

*"Вы точно хотите удалить?", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)*

*{*

*jazBS.RemoveCurrent();*

*jazTA.Update(portfolioDS.jaz);*

*}*

*}*

*private void button13\_Click(object sender, EventArgs e)*

*{*

*if (studBS.Count == 0)*

*{*

*MessageBox.Show("Список студентов пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*prevObrBS.AddNew();*

*button13.Enabled = false;*

*}*

*private void button14\_Click(object sender, EventArgs e)*

*{*

*if (prevObrBS.Count == 0)*

*{*

*MessageBox.Show("Список учереждений пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*if (MessageBox.Show(*

*(prevObrBS.Current as DataRowView)["ucher"].ToString(),*

*"Вы точно хотите удалить?", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)*

*{*

*prevObrBS.RemoveCurrent();*

*prevObrTA.Update(portfolioDS.prevObr);*

*}*

*}*

*private void button15\_Click(object sender, EventArgs e)*

*{*

*if (nauRab2BS.Count == 0)*

*{*

*MessageBox.Show("Список достижений пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*if (MessageBox.Show(*

*(nauRab2BS.Current as DataRowView)["tip"].ToString(),*

*"Вы точно хотите удалить?", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)*

*{*

*nauRab2BS.RemoveCurrent();*

*nauRab2TA.Update(portfolioDS.nauRab2);*

*}*

*}*

*private void button16\_Click(object sender, EventArgs e)*

*{*

*if (studBS.Count == 0)*

*{*

*MessageBox.Show("Список студентов пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*pos = nauRab2BS.Position;*

*nauRab2BS.AddNew();*

*NauRab nauRab = new NauRab();*

*nauRab.ShowDialog();*

*}*

*private void button17\_Click(object sender, EventArgs e)*

*{*

*if (vneRab3BS.Count == 0)*

*{*

*MessageBox.Show("Список достижений пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*if (MessageBox.Show(*

*(vneRab3BS.Current as DataRowView)["naz"].ToString(),*

*"Вы точно хотите удалить?", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)*

*{*

*vneRab3BS.RemoveCurrent();*

*vneRab3TA.Update(portfolioDS.vneRab3);*

*}*

*}*

*private void button18\_Click(object sender, EventArgs e)*

*{*

*if (studBS.Count == 0)*

*{*

*MessageBox.Show("Список студентов пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*pos = vneRab3BS.Position;*

*vneRab3BS.AddNew();*

*VneRab vneRab = new VneRab();*

*vneRab.ShowDialog();*

*}*

*private void button19\_Click(object sender, EventArgs e)*

*{*

*if (dopRab4BS.Count == 0)*

*{*

*MessageBox.Show("Список достижений пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*if (MessageBox.Show(*

*(dopRab4BS.Current as DataRowView)["uchir"].ToString(),*

*"Вы точно хотите удалить?", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)*

*{*

*dopRab4BS.RemoveCurrent();*

*dopRab4TA.Update(portfolioDS.dopRab4);*

*}*

*}*

*private void button20\_Click(object sender, EventArgs e)*

*{*

*if (studBS.Count == 0)*

*{*

*MessageBox.Show("Список студентов пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*pos = dopRab4BS.Position;*

*dopRab4BS.AddNew();*

*DopRab dopRab = new DopRab();*

*dopRab.ShowDialog();*

*}*

*private void button21\_Click(object sender, EventArgs e)*

*{*

*if (groupBS.Count == 0)*

*{*

*MessageBox.Show("Список групп пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*if (folderBrowserDialog1.ShowDialog() != DialogResult.OK)return;*

*string nameFolder = folderBrowserDialog1.SelectedPath;*

*show(nameFolder, (groupBS.Current as DataRowView), (studBS.Current as DataRowView));*

*}*

*private void button22\_Click(object sender, EventArgs e)*

*{*

*if (groupBS.Count == 0)*

*{*

*MessageBox.Show("Список групп пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*if (folderBrowserDialog1.ShowDialog() != DialogResult.OK)return;*

*string nameFolder = folderBrowserDialog1.SelectedPath + "\\гр." + (groupBS.Current as DataRowView)["naz"].ToString();*

*Directory.CreateDirectory(nameFolder);*

*foreach (DataRowView row in studBS)*

*{*

*if (row["idGroup"].ToString() == (groupBS.Current as DataRowView)["id"].ToString())*

*{*

*show(nameFolder, (groupBS.Current as DataRowView), row);*

*}*

*}*

*}*

*private void button23\_Click(object sender, EventArgs e)*

*{*

*if (prikazStudBS.Count == 0)*

*{*

*MessageBox.Show("Список приказов пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*if (MessageBox.Show(*

*(prikazStudBS.Current as DataRowView)["nomer"].ToString(),*

*"Вы точно хотите удалить?", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)*

*{*

*prikazStudBS.RemoveCurrent();*

*prikazStudTA.Update(portfolioDS.prikazStud);*

*}*

*}*

*private void button24\_Click(object sender, EventArgs e)*

*{*

*if (studBS.Count == 0)*

*{*

*MessageBox.Show("Список студентов Пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*prikazStudBS.AddNew();*

*button24.Enabled = false;*

*}*

*#endregion*

*private void specBS\_CurrentChanged(object sender, EventArgs e)*

*{*

*if (portfolioDS.spec.GetChanges() != null)*

*{*

*Validate();*

*specBS.EndEdit();*

*specTA.Update(portfolioDS.spec);*

*}*

*button1.Enabled = true;*

*}*

*private void groupBS\_CurrentChanged(object sender, EventArgs e)*

*{*

*if (portfolioDS.group.GetChanges() != null)*

*{*

*Validate();*

*groupBS.EndEdit();*

*groupTA.Update(portfolioDS.group);*

*}*

*button3.Enabled = true;*

*}*

*private void studBS\_CurrentChanged(object sender, EventArgs e)*

*{*

*label16.Text = "Количество студентов: " + studBS.Count.ToString();*

*if (portfolioDS.stud.GetChanges() != null)*

*{*

*Validate();*

*studBS.EndEdit();*

*studTA.Update(portfolioDS.stud);*

*}*

*button7.Enabled = true;*

*}*

*private void prevObrBS\_CurrentChanged(object sender, EventArgs e)*

*{*

*if (portfolioDS.prevObr.GetChanges() != null)*

*{*

*Validate();*

*prevObrBS.EndEdit();*

*prevObrTA.Update(portfolioDS.prevObr);*

*}*

*button13.Enabled = true;*

*}*

*private void nauRab2BS\_CurrentChanged(object sender, EventArgs e)*

*{*

*if (portfolioDS.nauRab2.GetChanges() != null)*

*{*

*Validate();*

*nauRab2BS.EndEdit();*

*nauRab2TA.Update(portfolioDS.nauRab2);*

*}*

*button16.Enabled = true;*

*}*

*private void vneRab3BS\_CurrentChanged(object sender, EventArgs e)*

*{*

*if (portfolioDS.vneRab3.GetChanges() != null)*

*{*

*Validate();*

*vneRab3BS.EndEdit();*

*vneRab3TA.Update(portfolioDS.vneRab3);*

*}*

*button18.Enabled = true;*

*}*

*private void dopRab4BS\_CurrentChanged(object sender, EventArgs e)*

*{*

*if (portfolioDS.dopRab4.GetChanges() != null)*

*{*

*Validate();*

*dopRab4BS.EndEdit();*

*dopRab4TA.Update(portfolioDS.dopRab4);*

*}*

*button20.Enabled = true;*

*}*

*private void prikazStudBS\_CurrentChanged(object sender, EventArgs e)*

*{*

*if (portfolioDS.prikazStud.GetChanges() != null)*

*{*

*Validate();*

*prikazStudBS.EndEdit();*

*prikazStudTA.Update(portfolioDS.prikazStud);*

*}*

*button24.Enabled = true;*

*}*

*private void kvalRabBS\_CurrentChanged(object sender, EventArgs e)*

*{*

*if (portfolioDS.kvalRab.GetChanges() != null)*

*{*

*Validate();*

*kvalRabBS.EndEdit();*

*kvalRabTA.Update(portfolioDS.kvalRab);*

*}*

*button10.Enabled = true;*

*}*

*public void prPraktBS\_CurrentChanged(object sender, EventArgs e)*

*{*

*if (portfolioDS.prPrakt.GetChanges() != null)*

*{*

*Validate();*

*prPraktBS.EndEdit();*

*prPraktTA.Update(portfolioDS.prPrakt);*

*}*

*}*

*private void jazBS\_CurrentChanged(object sender, EventArgs e)*

*{*

*if (portfolioDS.jaz.GetChanges() != null)*

*{*

*Validate();*

*jazBS.EndEdit();*

*jazTA.Update(portfolioDS.jaz);*

*}*

*button11.Enabled = true;*

*}*

*public void prKursBS\_CurrentChanged(object sender, EventArgs e)*

*{*

*if (portfolioDS.prKurs.GetChanges() != null)*

*{*

*Validate();*

*prKursBS.EndEdit();*

*prKursTA.Update(portfolioDS.prKurs);*

*}*

*}*

*public void prDiscBS\_CurrentChanged(object sender, EventArgs e)*

*{*

*if (portfolioDS.prDisc.GetChanges() != null)*

*{*

*Validate();*

*prDiscBS.EndEdit();*

*prDiscTA.Update(portfolioDS.prDisc);*

*}*

*}*

*private void discBS\_CurrentChanged(object sender, EventArgs e)*

*{*

*if (portfolioDS.disc.GetChanges() != null)*

*{*

*Validate();*

*discBS.EndEdit();*

*discTA.Update(portfolioDS.disc);*

*}*

*}*

*private void radioButton1\_CheckedChanged(object sender, EventArgs e)*

*{*

*if (radioButton1.Checked)*

*{*

*Program.general.discBS.RemoveFilter();*

*Program.general.discBS.Filter = "sem = '1'";*

*}*

*}*

*private void radioButton2\_CheckedChanged(object sender, EventArgs e)*

*{*

*if (radioButton2.Checked)*

*{*

*Program.general.discBS.RemoveFilter();*

*Program.general.discBS.Filter = "sem = '2'";*

*}*

*}*

*private void radioButton3\_CheckedChanged(object sender, EventArgs e)*

*{*

*if (radioButton3.Checked)*

*{*

*Program.general.discBS.RemoveFilter();*

*Program.general.discBS.Filter = "sem = '3'";*

*}*

*}*

*private void radioButton4\_CheckedChanged(object sender, EventArgs e)*

*{*

*if (radioButton4.Checked)*

*{*

*Program.general.discBS.RemoveFilter();*

*Program.general.discBS.Filter = "sem = '4'";*

*}*

*}*

*private void radioButton5\_CheckedChanged(object sender, EventArgs e)*

*{*

*if (radioButton5.Checked)*

*{*

*Program.general.discBS.RemoveFilter();*

*Program.general.discBS.Filter = "sem = '5'";*

*}*

*}*

*private void radioButton6\_CheckedChanged(object sender, EventArgs e)*

*{*

*if (radioButton6.Checked)*

*{*

*Program.general.discBS.RemoveFilter();*

*Program.general.discBS.Filter = "sem = '6'";*

*}*

*}*

*private void radioButton7\_CheckedChanged(object sender, EventArgs e)*

*{*

*if (radioButton7.Checked)*

*{*

*Program.general.discBS.RemoveFilter();*

*Program.general.discBS.Filter = "sem = '7'";*

*}*

*}*

*private void radioButton8\_CheckedChanged(object sender, EventArgs e)*

*{*

*if (radioButton8.Checked)*

*{*

*Program.general.discBS.RemoveFilter();*

*Program.general.discBS.Filter = "sem = '8'";*

*}*

*}*

*private void dataGridView3\_DoubleClick(object sender, EventArgs e)*

*{*

*pos = studBS.Position;*

*Profile profile = new Profile();*

*profile.ShowDialog();*

*}*

*private void dataGridView8\_DoubleClick(object sender, EventArgs e)*

*{*

*pos = nauRab2BS.Position;*

*NauRab nauRab = new NauRab();*

*nauRab.ShowDialog();*

*}*

*private void dataGridView9\_DoubleClick(object sender, EventArgs e)*

*{*

*pos = vneRab3BS.Position;*

*VneRab vneRab = new VneRab();*

*vneRab.ShowDialog();*

*}*

*private void dataGridView10\_DoubleClick(object sender, EventArgs e)*

*{*

*pos = dopRab4BS.Position;*

*DopRab dopRab4 = new DopRab();*

*dopRab4.ShowDialog();*

*}*

*}*

*}*

*Исходный код формы Login:*

*using System;*

*using System.Collections.Generic;*

*using System.ComponentModel;*

*using System.Data;*

*using System.Drawing;*

*using System.Linq;*

*using System.Text;*

*using System.Windows.Forms;*

*namespace diplom*

*{*

*public partial class Login : Form*

*{*

*public Login()*

*{*

*InitializeComponent();*

*}*

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

*{*

*Program.user = textBox1.Text;*

*Program.password = textBox2.Text;*

*Program.server = textBox3.Text;*

*Program.port = textBox4.Text;*

*Close();*

*}*

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

*{*

*Close();*

*}*

*}*

*}*

*Исходный код формы Perevod:*

*using System;*

*using System.Collections.Generic;*

*using System.ComponentModel;*

*using System.Data;*

*using System.Drawing;*

*using System.Linq;*

*using System.Text;*

*using System.Windows.Forms;*

*namespace diplom*

*{*

*public partial class Perevod : Form*

*{*

*public Perevod()*

*{*

*InitializeComponent();*

*}*

*private void Perevod\_Load(object sender, EventArgs e)*

*{*

*this.prikazPerevodTA.Fill(this.portfolioDataSet.prikazPerevod);*

*Perevod.ActiveForm.Text += (Program.general.groupBS.Current as DataRowView)["naz"].ToString();*

*dataGridView1.DataSource = Program.general.prikazPerevodBS;*

*}*

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

*{*

*Program.general.prikazPerevodBS.CancelEdit();*

*Close();*

*Dispose();*

*}*

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

*{*

*Program.general.Validate();*

*Program.general.prikazPerevodBS.EndEdit();*

*Program.general.prikazPerevodTA.Update(Program.general.portfolioDS.prikazPerevod);*

*Close();*

*Dispose();*

*}*

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

*{*

*Program.general.prikazPerevodBS.AddNew();*

*button3.Enabled = false;*

*}*

*private void button4\_Click(object sender, EventArgs e)*

*{*

*if (MessageBox.Show(*

*(Program.general.prikazPerevodBS.Current as DataRowView)["nomer"].ToString(),*

*"Вы точно хотите удалить?", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)*

*{*

*Program.general.prikazPerevodBS.RemoveCurrent();*

*Program.general.prikazPerevodTA.Update(Program.general.portfolioDS.prikazPerevod);*

*}*

*}*

*private void Perevod\_KeyUp(object sender, KeyEventArgs e)*

*{*

*prikazPerevodBS\_CurrentChanged(sender, e);*

*}*

*private void prikazPerevodBS\_CurrentChanged(object sender, EventArgs e)*

*{*

*if (Program.general.portfolioDS.prikazPerevod.GetChanges() != null)*

*{*

*Validate();*

*Program.general.prikazPerevodBS.EndEdit();*

*Program.general.prikazPerevodTA.Update(Program.general.portfolioDS.prikazPerevod);*

*}*

*button3.Enabled = true;*

*}*

*}*

*}*

*Исходный код формы Predm:*

*using System;*

*using System.Collections.Generic;*

*using System.ComponentModel;*

*using System.Data;*

*using System.Drawing;*

*using System.Linq;*

*using System.Text;*

*using System.Windows.Forms;*

*namespace diplom*

*{*

*public partial class Predm : Form*

*{*

*public Predm()*

*{*

*InitializeComponent();*

*}*

*private void Predm\_Load(object sender, EventArgs e)*

*{*

*dataGridView1.DataSource = Program.general.prKursBS;*

*dataGridView2.DataSource = Program.general.prPraktBS;*

*dataGridView3.DataSource = Program.general.prDiscBS;*

*Program.general.prDiscBS.Filter = "sem = '1'";*

*}*

*private void Predm\_KeyUp(object sender, KeyEventArgs e)*

*{*

*prPraktBS\_CurrentChanged(sender, e);*

*prKursBS\_CurrentChanged(sender, e);*

*if (e.KeyCode == Keys.Enter) prDiscBS\_CurrentChanged(sender, e);*

*}*

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

*{*

*Program.general.discBS.RemoveFilter();*

*Program.general.prDiscBS.RemoveFilter();*

*Program.general.Validate();*

*Program.general.prPraktBS.EndEdit();*

*Program.general.prKursBS.EndEdit();*

*Program.general.prDiscBS.EndEdit();*

*if (Program.general.prKursBS.Count != 0)*

*{*

*Program.general.addDisciplKurs();*

*Program.general.delDisciplKurs();*

*}*

*if (Program.general.prPraktBS.Count != 0)*

*{*

*Program.general.addDisciplPrakt();*

*Program.general.delDisciplPrakt();*

*}*

*if (Program.general.prDiscBS.Count != 0)*

*{*

*Program.general.addDisciplDisc();*

*Program.general.delDisciplDisc();*

*}*

*Program.general.prPraktTA.Update(Program.general.portfolioDS.prPrakt);*

*Program.general.prKursTA.Update(Program.general.portfolioDS.prKurs);*

*Program.general.prDiscTA.Update(Program.general.portfolioDS.prDisc);*

*Program.general.praktTA.Update(Program.general.portfolioDS.prakt);*

*Program.general.kursTA.Update(Program.general.portfolioDS.kurs);*

*Program.general.discTA.Update(Program.general.portfolioDS.disc);*

*Program.general.radioButton1.Checked = true;*

*Program.general.discBS.Filter = "sem = '1'";*

*Close();*

*Dispose();*

*}*

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

*{*

*Program.general.prPraktBS.CancelEdit();*

*Program.general.prKursBS.CancelEdit();*

*Close();*

*Dispose();*

*}*

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

*{*

*Program.general.prKursBS.AddNew();*

*button3.Enabled = false;*

*button4.Enabled = false;*

*}*

*private void button4\_Click(object sender, EventArgs e)*

*{*

*if (Program.general.prKursBS.Count == 0)*

*{*

*MessageBox.Show("Список дисциплин пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*if (MessageBox.Show(*

*(Program.general.prKursBS.Current as DataRowView)["disc"].ToString(),*

*"Вы точно хотите удалить?", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)*

*{*

*Program.general.prKursBS.RemoveCurrent();*

*Program.general.prKursTA.Update(Program.general.portfolioDS.prKurs);*

*}*

*}*

*private void button5\_Click(object sender, EventArgs e)*

*{*

*Program.general.prPraktBS.AddNew();*

*button5.Enabled = false;*

*button6.Enabled = false;*

*}*

*private void button6\_Click(object sender, EventArgs e)*

*{*

*if (Program.general.prPraktBS.Count == 0)*

*{*

*MessageBox.Show("Список дисциплин пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*if (MessageBox.Show(*

*(Program.general.prPraktBS.Current as DataRowView)["vid"].ToString(),*

*"Вы точно хотите удалить?", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)*

*{*

*Program.general.prPraktBS.RemoveCurrent();*

*Program.general.prPraktTA.Update(Program.general.portfolioDS.prPrakt);*

*}*

*}*

*private void button7\_Click(object sender, EventArgs e)*

*{*

*Program.general.prDiscBS.AddNew();*

*if (radioButton1.Checked)*

*{*

*((DataRowView)Program.general.prDiscBS.Current)["sem"] = "1";*

*}*

*if (radioButton2.Checked)*

*{*

*((DataRowView)Program.general.prDiscBS.Current)["sem"] = "2";*

*}*

*if (radioButton3.Checked)*

*{*

*((DataRowView)Program.general.prDiscBS.Current)["sem"] = "3";*

*}*

*if (radioButton4.Checked)*

*{*

*((DataRowView)Program.general.prDiscBS.Current)["sem"] = "4";*

*}*

*if (radioButton5.Checked)*

*{*

*((DataRowView)Program.general.prDiscBS.Current)["sem"] = "5";*

*}*

*if (radioButton6.Checked)*

*{*

*((DataRowView)Program.general.prDiscBS.Current)["sem"] = "6";*

*}*

*if (radioButton7.Checked)*

*{*

*((DataRowView)Program.general.prDiscBS.Current)["sem"] = "7";*

*}*

*if (radioButton8.Checked)*

*{*

*((DataRowView)Program.general.prDiscBS.Current)["sem"] = "8";*

*}*

*Validate();*

*Program.general.prDiscBS.EndEdit();*

*Program.general.prDiscTA.Update(Program.general.portfolioDS.prDisc);*

*dataGridView3.DataSource = Program.general.prDiscBS;*

*button7.Enabled = false;*

*button8.Enabled = false;*

*}*

*private void button8\_Click(object sender, EventArgs e)*

*{*

*if (dataGridView3.Rows.Count==0)*

*{*

*MessageBox.Show("Список дисциплин пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);*

*return;*

*}*

*Program.general.prDiscBS.CancelEdit();*

*if (MessageBox.Show(*

*(Program.general.prDiscBS.Current as DataRowView)["disc"].ToString(),*

*"Вы точно хотите удалить?", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)*

*{*

*Program.general.prDiscBS.RemoveCurrent();*

*Program.general.prDiscTA.Update(Program.general.portfolioDS.prDisc);*

*}*

*button7.Enabled = true;*

*button8.Enabled = true;*

*}*

*private void prPraktBS\_CurrentChanged(object sender, EventArgs e)*

*{*

*Program.general.prPraktBS\_CurrentChanged(sender, e);*

*button5.Enabled = true;*

*button6.Enabled = true;*

*}*

*private void prKursBS\_CurrentChanged(object sender, EventArgs e)*

*{*

*Program.general.prKursBS\_CurrentChanged(sender, e);*

*button3.Enabled = true;*

*button4.Enabled = true;*

*}*

*private void prDiscBS\_CurrentChanged(object sender, EventArgs e)*

*{*

*Program.general.prDiscBS\_CurrentChanged(sender, e);*

*button7.Enabled = true;*

*button8.Enabled = true;*

*}*

*#region rb*

*private void radioButton1\_CheckedChanged(object sender, EventArgs e)*

*{*

*if (radioButton1.Checked)*

*{*

*Program.general.prDiscBS.Filter = "sem = '1'";*

*//Program.general.prDiscTA.Update(Program.general.portfolioDS.prDisc);*

*}*

*}*

*private void radioButton2\_CheckedChanged(object sender, EventArgs e)*

*{*

*if (radioButton2.Checked)*

*{*

*Program.general.prDiscBS.Filter = "sem = '2'";*

*//Program.general.prDiscTA.Update(Program.general.portfolioDS.prDisc);*

*}*

*}*

*private void radioButton3\_CheckedChanged(object sender, EventArgs e)*

*{*

*if (radioButton3.Checked)*

*{*

*Program.general.prDiscBS.Filter = "sem = '3'";*

*//Program.general.prDiscTA.Update(Program.general.portfolioDS.prDisc);*

*}*

*}*

*private void radioButton4\_CheckedChanged(object sender, EventArgs e)*

*{*

*if (radioButton4.Checked)*

*{*

*Program.general.prDiscBS.Filter = "sem = '4'";*

*//Program.general.prDiscTA.Update(Program.general.portfolioDS.prDisc);*

*}*

*}*

*private void radioButton5\_CheckedChanged(object sender, EventArgs e)*

*{*

*if (radioButton5.Checked)*

*{*

*Program.general.prDiscBS.Filter = "sem = '5'";*

*//Program.general.prDiscTA.Update(Program.general.portfolioDS.prDisc);*

*}*

*}*

*private void radioButton6\_CheckedChanged(object sender, EventArgs e)*

*{*

*if (radioButton6.Checked)*

*{*

*Program.general.prDiscBS.Filter = "sem = '6'";*

*//Program.general.prDiscTA.Update(Program.general.portfolioDS.prDisc);*

*}*

*}*

*private void radioButton7\_CheckedChanged(object sender, EventArgs e)*

*{*

*if (radioButton7.Checked)*

*{*

*Program.general.prDiscBS.Filter = "sem = '7'";*

*//Program.general.prDiscTA.Update(Program.general.portfolioDS.prDisc);*

*}*

*}*

*private void radioButton8\_CheckedChanged(object sender, EventArgs e)*

*{*

*if (radioButton8.Checked)*

*{*

*Program.general.prDiscBS.Filter = "sem = '8'";*

*//Program.general.prDiscTA.Update(Program.general.portfolioDS.prDisc);*

*}*

*}*

*#endregion*

*}*

*}*

*Исходный код формы Predm:*

*using System;*

*using System.Collections.Generic;*

*using System.ComponentModel;*

*using System.Data;*

*using System.Drawing;*

*using System.Linq;*

*using System.Text;*

*using System.Windows.Forms;*

*namespace diplom*

*{*

*public partial class Profile : Form*

*{*

*public Profile()*

*{*

*InitializeComponent();*

*}*

*private void Profile\_Load(object sender, EventArgs e)*

*{*

*textBox1.DataBindings.Add(new Binding("Text", Program.general.studBS, "fam", true));*

*textBox2.DataBindings.Add(new Binding("Text", Program.general.studBS, "name", true));*

*textBox3.DataBindings.Add(new Binding("Text", Program.general.studBS, "otch", true));*

*textBox4.DataBindings.Add(new Binding("Text", Program.general.studBS, "tel", true));*

*textBox5.DataBindings.Add(new Binding("Text", Program.general.studBS, "email", true));*

*textBox6.DataBindings.Add(new Binding("Text", Program.general.studBS, "nomZachisl", true));*

*dateTimePicker1.DataBindings.Add(new Binding("Value", Program.general.studBS, "dateR", true));*

*dateTimePicker2.DataBindings.Add(new Binding("Value", Program.general.studBS, "dateZachisl", true));*

*comboBox1.DataBindings.Add(new Binding("Text", Program.general.studBS, "urVlad", true));*

*pictureBox1.DataBindings.Add(new Binding("Image", Program.general.studBS, "foto", true));*

*}*

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

*{*

*if (openFileDialog1.ShowDialog() == DialogResult.OK)*

*{*

*Bitmap pict = new Bitmap(openFileDialog1.FileName);*

*pictureBox1.Image = (Image)pict;*

*}*

*}*

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

*{*

*pictureBox1.Image = null;*

*}*

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

*{*

*Program.general.discBS.RemoveFilter();*

*Program.general.Validate();*

*Program.general.studBS.EndEdit();*

*Program.general.studTA.Update(Program.general.portfolioDS.stud);*

*Program.general.addPraktStud();*

*Program.general.praktTA.Update(Program.general.portfolioDS.prakt);*

*Program.general.addKursStud();*

*Program.general.kursTA.Update(Program.general.portfolioDS.kurs);*

*Program.general.addDisciplStud();*

*Program.general.discTA.Update(Program.general.portfolioDS.disc);*

*Program.general.discBS.Filter = "sem = '1'";*

*Program.general.radioButton1.Checked = true;*

*Close();*

*Dispose();*

*}*

*private void button4\_Click(object sender, EventArgs e)*

*{*

*Program.general.studBS.CancelEdit();*

*Close();*

*Dispose();*

*Program.general.studBS.Position = Program.general.pos;*

*}*

*}*

*}*

*Исходный код формы Zast:*

*using System;*

*using System.Collections.Generic;*

*using System.ComponentModel;*

*using System.Data;*

*using System.Drawing;*

*using System.Linq;*

*using System.Text;*

*using System.Windows.Forms;*

*namespace diplom*

*{*

*public partial class Zast : Form*

*{*

*public Zast()*

*{*

*InitializeComponent();*

*timer1.Start();*

*}*

*private void timer1\_Tick(object sender, EventArgs e)*

*{*

*for (int i = 0; i < 10; i++)*

*{*

*Opacity += 0.1;*

*}*

*}*

*}*

*}*