МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«Новосибирский национальный исследовательский государственный университет»

(Новосибирский государственный университет, НГУ)

Структурное подразделение Новосибирского государственного университета – Высший колледж информатики Университета (ВКИ НГУ)

КАФЕДРА ИНФОРМАТИКИ

**Разработка программного средства для хранения данных электромагнитных зондирований становлением поля в ближней зоне с использованием реляционной базы данных**

Квалификация программист

|  |  |
| --- | --- |
| Супервайзер  к.т.н., с.н.с. ИСИ СО РАН | Агеенков Евгений Валерьевич  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2024 г. |
| Студент 4 курса  гр. 007а2 | Андреев Данила Александрович  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_2024 г. |

Новосибирск

2024

**СОДЕРЖАНИЕ**

[ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, УСЛОВНЫХ ОБОЗНАЧЕНИЙ И ТЕРМИНОВ 4](#_Toc167257712)

[ВВЕДЕНИЕ 6](#_Toc167257713)

[1 ПОСТАНОВКА ЗАДАЧИ ВКР 8](#_Toc167257714)

[*1.1* *Бизнес-требования* 8](#_Toc167257715)

[*1.2* *Пользовательские требования* 9](#_Toc167257716)

[*1.3* *Системные требования* 9](#_Toc167257717)

[*1.4* *Требования к графическому пользовательскому интерфейсу* 10](#_Toc167257718)

[*1.5* *План-график выполнения ВКР* 11](#_Toc167257719)

[2 АНАЛИЗ ТРЕБОВАНИЙ И ОПРЕДЕЛЕНИЕ СПЕЦИФИКАЦИЙ 13](#_Toc167257720)

[*2.1 Описание предметной области задачи ВКР* 13](#_Toc167257721)

[*2.1.1 Информационные объекты предметной области, их характеристика и взаимосвязь между ними* 13](#_Toc167257722)

[*2.1.2 Информационные и функциональные потребности пользователей разрабатываемого ПМ* 14](#_Toc167257723)

[*2.1.3 Методы работы с информационными объектами предметной области* 14](#_Toc167257724)

[*2.1.4 Концептуальное обоснование разработки* 15](#_Toc167257725)

[*2.2 Классы и характеристики пользователей* 16](#_Toc167257726)

[*2.3 Функциональные требования* 17](#_Toc167257727)

[*2.3.1 Определение функциональных возможностей программного модуля* 17](#_Toc167257728)

[*2.3.2 Описание прецедентов* 18](#_Toc167257729)

[*2.4 Нефункциональные требования* 21](#_Toc167257730)

[3 ВЫБОР ПРОГРАММНЫХ СРЕД И СРЕДСТВ РАЗРАБОТКИ 22](#_Toc167257731)

[*3.1 Сравнительный анализ имеющихся возможностей по выбору средств разработки* 22](#_Toc167257732)

[*3.2 Характеристика выбранных программных сред и средств* 25](#_Toc167257733)

[4 АЛГОРИТМ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ 26](#_Toc167257734)

[*4.1 Этапы реализации ПМ* 26](#_Toc167257735)

[*4.2 Пользовательский интерфейс ПМ* 26](#_Toc167257736)

[*4.2.1 Взаимодействие пользователей с ПМ* 26](#_Toc167257737)

[*4.2.2 Проектирование пользовательских сценариев* 28](#_Toc167257738)

[*4.2.3 Определение операций пользователей* 29](#_Toc167257739)

[*4.2.4 Составление функциональных блоков* 31](#_Toc167257740)

[*4.2.5 Проектирование структуры экранов (ПМ) и схемы навигации* 32](#_Toc167257741)

[*4.2.6 Низкоуровневое проектирование* 33](#_Toc167257742)

[*4.3 Входные, выходные и промежуточные данные* 35](#_Toc167257743)

[*4.4 Разработка базы данных, реализуемой в рамках (ПМ)* 36](#_Toc167257744)

[*4.5 Алгоритмы передачи данных* 47](#_Toc167257745)

[*4.6 Архитектура программной системы* 47](#_Toc167257746)

[*4.7 Архитектура программного кода модуля* 48](#_Toc167257747)

[5 ТЕСТИРОВАНИЕ И ОПТИМИЗАЦИЯМ 49](#_Toc167257748)

[*5.1 План тестирования* 49](#_Toc167257749)

[*5.2 Результаты тестирования* 50](#_Toc167257750)

[*5.3* *Оптимизация ПМ* 53](#_Toc167257751)

[6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 54](#_Toc167257752)

[*6.1 Установка приложения* 54](#_Toc167257753)

[*6.2 Основные операции 55*](#_Toc167257754)

[*6.2.1 Первый запуск программы 55*](#_Toc167257755)

[*6.2.2 Создание записи в базе данных* 55](#_Toc167257759)

[*6.2.3 Изменение записи в базе данных* 56](#_Toc167257760)

[*6.2.4 Удаление записи в базе данных* 57](#_Toc167257761)

[ЗАКЛЮЧЕНИЕ 58](#_Toc167257762)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 59](#_Toc167257763)

[ПРИЛОЖЕНИЯ 59](#_Toc167257764)

[*Приложение А* 59](#_Toc167257765)

[*Приложение Б* 61](#_Toc167257766)

[*Приложение В* 61](#_Toc167257767)

[*Приложение Г* 65](#_Toc167257768)

[*Приложение Д* 65](#_Toc167257769)

[*Приложение Е* 65](#_Toc167257770)

# *ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, УСЛОВНЫХ ОБОЗНАЧЕНИЙ И ТЕРМИНОВ*

База Данных (БД) − совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.

Супервайзер − административная должность в различных отраслях бизнеса, государственных учреждениях, а также в научных и образовательных институтах, выполняющая контроль за работой персонала.

Геодезия − это наука о методах и технике производства измерений на земной поверхности, выполняемых с целью изучения фигуры Земли, изображения земной поверхности в виде планов, карт, профилей, а также решения различных прикладных задач. В геодезии применяются преимущественно линейные и угловые измерения.

Зондирование становлением поля в ближней зоне (метод ЗСБ) − метод с импульсным контролируемым источником, основанный на изучении поля переходных процессов, которое возбуждается в земле при изменении тока в источнике.

Геофизическая информационная система (ГИС) − система, которая используется для обработки и анализа геофизических данных.

Электродвижущая сила (ЭДС) − это скалярная физическая величина, которая является результатом явления электромагнитной индукции.

Лаборатория Геоэлектрики института нефтегазовой геологии и геофизики СО РАН (ИНГГ СО РАН) - научно-исследовательское подразделение Института нефтегазовой геологии и геофизики им. А.А. Трофимука Сибирского отделения Российской академии наук

Геофизика − наука о поиске и разведке месторождений полезных ископаемых физическим методом.

Электроразведка − одна из отраслей геофизики, основанная на изучении естественных и искусственно созданных в Земле электромагнитных полей.

# *ВВЕДЕНИЕ*

Исследования электромагнитных полей начались в 80-90-х годах 20-го века и были успешно выполнены исследования в СНИИГГиМС по разработке методики работы электромагнитных полей и методов. Занимались ЗСБ-МПП (Г.А.Исаев, Б.И.Рабинович, А.К.Захаркин, В.В.Филатов, Г.М.Тригубович, B.C.Могилатов), ВП и наземно-скважинной электроразведки (B.C.Моисеев, B.C.Могилатов). Наряду с теоретическими и методическими исследованиями осуществлены удачные аппаратурные разработки («Цикл-2»), в том числе, под руководством и при непосредственном участии автора («Цикл-4», «Цикл-Микро», «Цикл-5»). Исследование электромагнитных полей были использованы в регионах Европейской части России, Урала, Западной и Восточной Сибири, Дальнего Востока, Казахстана, Средней Азии для поисков углеводородов, рудных и кимберлитовых тел.

Накопленный научно-технический потенциал позволил сделать качественный шаг в развитии индукционной электроразведки и перейти к разработке метода ЗСБ и созданию аппаратурно-методического комплекса (электроразведочной системы) для его реализации.

Метод электромагнитных зондированиях становлением поля в ближней зоне(далее ЗСБ) относится к группе индуктивных методов электроразведки. Метод ЗСБ отличается тем, что в этом методе изучается неустановившееся электромагнитное поле (становление поля), и сведения о геоэлектрическом разрезе исследуемой площади получаются на основе изучения характера переходного процесса в различных пунктах наблюдения.

Сущность метода заключается в возбуждении с помощью незаземленного контура первичного импульсного магнитного поля, которое в свою очередь наводит в проводящих горных породах и рудах первичное неустановившееся поле вихревых электрических токов, и в приеме в паузах между импульсами на другой или тот же контур вторичного неустановившегося магнитного поля, возникающего под действием затухающих вихревых токов от проводящих объектов в земле.

Этот метод применяется при глубинном методе исследования. Подбирая размеры генераторных и приемных петель, можно исследовать геологические разрезы в довольно широком диапазоне глубин, становится возможным одновременное решение вопросов нескольких этапов исследований (например­­­ − исследование верхних рыхлых отложений и подстилающих их коренных пород фундамента).

Объектом исследований является аппаратурное и метрологическое обеспечение площадной ЗБ-электроразведки метода зондирований становлением в ближней зоне (ЗСБ).

Цельюданного дипломного проекта является разработка программного обеспечения, способного эффективно хранить и управлять реляционными данными, полученными методом ЗСБ.

Для решения цели, необходимо решить следующие задачи:

* Изучить предметную область метода измерений ЗБС.
* Изучение предметную область реляционной базы данных.
* Разработать реляционную базу данных.
* Создать интерфейс для работы с базой данных.

Для реализации проекта будут использованы следующие технологии:

* Visual Studio 2022 − среда разработки с использованием языка программирования C#
* Windows Presentation Foundation (WPF) − создание пользовательского интерфейса
* SQl Server 2019 − реляционная система управления базами данных

# 1 ПОСТАНОВКА ЗАДАЧИ ВКР

## *Бизнес-требования*

Одним из ключевых направлений работы ИНГГ СО РАН является изучение геологического строения Земли с использованием метода ЗБС. В ходе таких исследований появляется огромное количество данных, которые необходимо хранить, обрабатывать и анализировать.

В результате исследования геологического строения земли ИНГГ СО РАН использует метод ЗБС. Результаты исследований хранятся и обрабатываются.

Следующие требования для решения поставленной задачи:

* Обеспечение безопасного хранения − программное обеспечение должно использовать шифрование для защиты конфиденциальной информации при хранении.
* Наличие поддержки различных форматов входных данных − программное обеспечение должно предоставлять возможность преобразования различных форматов входных данных для их правильной обработки.
* Организация хранения в реляционной базе данных − программное обеспечение должно обеспечивать надежное хранение данных, полученных методом ЗБС, с быстрым и легким доступом к ним.
* Наличие системы резервного копирования − при сбой системы важно сохранить важную информацию.
* Наличие настроек для управления параметрами хранения

данных − программное обеспечение должно поддерживать настройки для управления параметрами хранения данных.

В результате, требуется программное обеспечение, которое обеспечивает безопасное хранение данных в реляционной базе, выполняя все указанные требования.

## *Пользовательские требования*

При разработке программного обеспечения следует учитывать требования к функционалу, который должен иметь пользователь при работе с программой, а также характеристики продукта, важные для пользователей. Основываясь на этом, мы выделили следующие списки требований и характеристик для программного обеспечения.

Требования к функциональным возможностям программы:

* Регистрация проектов, площадок, профилей и точек на профиле – пользователь должен быть способен вносить информацию о различных объектах.
* Возможность записи результатов измерений – пользователь должен иметь возможность сохранять данные, полученные при проведении измерений.
* Регистрация операторов и менеджеров – пользователь должен иметь доступ к функциям, позволяющим регистрировать операторов и управляющих.
* Создание отчетов − пользователь должен иметь функцию создания отчетов и экспорта данных.

Требования к техническим характеристикам программы:

* Надежность – ПО должно работать стабильно и надежно, чтобы пользователь чувствовал уверенность в сохранении своих данных.
* Высокая скорость работы – ПО должно обеспечить высокую скорость обработки информации.
* Простота использования − ПО должно быть удобно и интуитивно понятно для пользователя.

## *Системные требования*

Это программное обеспечение является составной частью большой системы, состоящей из трех подсистем. Каждая из них выполняет свою функцию для обеспечения пользователю эффективного и надежного способа хранения, обработки и анализа данных

1. Система хранения результатов ЗБС с использованием реляционных баз данных − эта система предоставляет интерфейс для работы с данными в реляционных базах данных. Эта модель базы данных упрощает работу с обработкой и анализом данных и обеспечивает их безопасное хранение.
2. Система хранения результатов ЗБС данных с использованием локальных файловых баз данных − эта система обеспечивает интерфейс для работы с данными в файловых базах данных на локальном компьютере. Эта модель баз данных обеспечивает эффективное хранение больших объемов данных и быстрый доступ к ним.
3. Конвертер между реляционными и файловыми базами данных − эта система позволяет конвертировать данные из одной базы в другую, что позволяет пользователю работать с данными в разных форматах баз данных.

Система с реляционными базами данных разделяется на две части:

1. База данных: обеспечивает эффективное хранение и быстрый доступ к большим объемам данных. База данных предоставляет пользователю доступ к данным через графический интерфейс программы.
2. Пользовательский интерфейс: предоставляет пользователям доступ к базе данных, позволяя им вводить, редактировать и просматривать данные, а также выполнять различные операции с ними.

## *Требования к графическому пользовательскому интерфейсу*

При создании пользовательского интерфейса этого программного обеспечения, следует разделить требования на две группы:

Требования к внешнему виду:

* Простой и понятный дизайн − пользовательский интерфейс должен быть прост в использовании и удобен, с учетом предпочтений и потребностей пользователей.
* Легкая навигация по функциям интерфейса − пользователь должен иметь быстрый доступ к основным функциям без необходимости изучать сложные инструкции.
* Интуитивное перемещение между формами − элементы интерфейса должны быть расположены так, чтобы пользователям было легко ориентироваться.
* Отображение состояния процесса − интерфейс должен показывать пользователю, что происходит в текущий момент.
* Простой вид результатов − результаты должны быть представлены в понятной и наглядной форме.
* Отображение данных − пользователи должны иметь возможность просматривать данные в виде графиков или таблиц.
* Поиск и фильтрация данных − должна быть возможность искать и фильтровать данные по различным параметрам.

## *План-график выполнения ВКР*

На основе предоставленных требований, можно выделить следующие задачи для выполнения выпускной квалификационной работы:

1. Изучение предметной области − этап предполагает изучение предметной области, включающее анализ требований. Этап разделен на две задачи: изучение метода ЗБС и изучение РБД.
2. Написание и доработка ВКР − этап предполагает написание ВКР.
3. Разработка РБД − этап предполагает разработку РБД.
4. Создание и настройка РБД − этап предполагает создание БД для предметной области.
5. Создание интерфейса − этап предполагает разработку интерфейса для работы с базой данных.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ГРАФИК ВЫПОЛНЕНИЯ | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Месяц | Октябрь | | | | Ноябрь | | | | Декабрь | | | | Январь | | | | Февраль | | | | Март | | | | Апрель | | | |
| Неделя | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| Изучение предметной области (Метод ЗБС) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Изучение предметной области (Реляционная БД) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Написание ВКР |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Разработка Реляционная БД |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Создание и настройка БД для программного обеспечения |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Доработка ВКР |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Создание интерфейса для работы с БД |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Таблица 1.5.1 - График выполнения ВКР.

Из всего вышеперечисленного можно сделать вывод, что на выполнение данной ВКР потребуется приблизительно 28 недель работы.

# *2 АНАЛИЗ ТРЕБОВАНИЙ И ОПРЕДЕЛЕНИЕ СПЕЦИФИКАЦИЙ*

## *2.1 Описание предметной области задачи ВКР*

Требуется создать программу для хранения данных, получаемых с помощью электромагнитного зондирования методом становления поля в ближней зоне (ЗСБ). ЗСБ является широко используемой и развитой технологией в мире для изучения геофизической среды на расстояниях от нескольких десятков до тысяч метров.

Процесс ЗСБ включает создание первичного импульсного магнитного поля с помощью разомкнутого контура, что вызывает возникновение неуспокоенных электрических токов в проводящих горных породах и рудах. В интервалах между импульсами измеряется вторичное магнитное поле, создаваемое затухающими токами через проводящие объекты под землей, с использованием того же или иного контура.

В общем, метод ЗСБ использует “включатель” для генерации магнитного поля, образующего токи в различных рудных материалах, а затем измеряет магнитное поле этих токов во время остановки “включателя”, используя тот же или иной “включатель”.

### *2.1.1 Информационные объекты предметной области, их характеристика и взаимосвязь между ними*

В процессе анализа предметной области и учета требований к разрабатываемому программному обеспечению можно выделить следующие комплексные (структурированные) информационные объекты:

* Реляционная База Данных – совокупность таблиц и связей между ними, отражающая сущности геофизических измерений. БД используется для хранения информаций о геофизических измерениях.

### *2.1.2 Информационные и функциональные потребности пользователей разрабатываемого ПМ*

С учетом требований пользователей, основная задача программного обеспечения − предоставить возможность сохранять и обрабатывать полученные в результате измерений данные.

Система должна позволять добавлять новые данные, редактировать и удалять их, выполнять поиск и сортировку по различным параметрам, включая ключевые слова, даты, метки и другие атрибуты.

Еще одной важной функцией должно быть создание отчетов о проведенных измерениях и их анализ. Отчеты должны обладать гибкостью и возможностью настройки, чтобы пользователи имели возможность выбирать, какие данные включить и как их представить. Анализ отчетов должен предусматривать различные статистические методы, такие как вычисление среднего значения, стандартного отклонения, минимальных и максимальных значений и прочих.

Кроме того, программное обеспечение должно предоставлять возможность экспорта данных в различных форматах, например, CSV, Excel, PDF и других, чтобы пользователи могли анализировать данные с использованием других инструментов или применять их в других целях.

### *2.1.3 Методы работы с информационными объектами предметной области*

#### *2.1.3.1 Реализация реляционной базы данных*

Реляционная база данных (РБД) представляет собой основу для многих приложений и систем хранения данных. Она состоит из огромных таблиц с множеством строк и столбцов, где данные хранятся в отдельных ячейках и могут быть легко изменены или получены.

РБД структурирована таким образом, что данные связаны между собой через ключи, обеспечивая эффективное хранение и обработку информации. Это делает РБД подходящей для хранения структурированных данных и обработки большого количества информации.

При проектировании РБД учитываются такие показатели, как надёжность, эффективность, удобство использования и масштабируемость. Надёжность предполагает устойчивость к сбоям и возможность резервного копирования данных. Эффективность означает быструю обработку SQL-запросов, а удобство использования облегчает работу пользователей с базой данных. Масштабируемость позволяет адаптироваться к изменениям в объёме данных и количестве пользователей

#### *2.1.3.2 Применяемые программные технологии*

Программные технологии, использовавшиеся для разработки программного модуля:

* Логическое и физическое моделирование для проектирования БД.
* Проектирование программного модуля и интерфейса взаимодействия с БД.
* Программирование модуля.
* Заполнение БД синтетическими записями.
* Тестирование и отладка программного средства.
* Составление документации.

### *2.1.4 Концептуальное обоснование разработки*

Разработка программного обеспечения для управления данными ЗСБ направлена на повышение эффективности и точности обработки данных, улучшение качества результатов, оптимизацию рабочих процессов, снижение нагрузки на сотрудников и уменьшение вероятности ошибок, связанных с ручным вводом информации. Кроме того, автоматизация процессов способствует более эффективному использованию ресурсов и сокращению затрат на сбор и обработку данных.

Применение реляционной базы данных (РБД) позволяет избежать использования мощных коммерческих СУБД при выполнении полевых работ на малопроизводительных компьютерах. Благодаря РБД исследователи получают быстрый и удобный доступ к структурированным данным, что позволяет анализировать информацию в режиме реального времени и оперативно получать результаты.

## *2.2 Классы и характеристики пользователей*

Из требований заказчика, для данной программной системе характерны следующие пользовательские роли: оператор, обработчик данных и Супервайзер. Их характеристики указанные в таблице 2.2.1.

Таблица 2.2.1 - Пользовательские роли

|  |  |  |
| --- | --- | --- |
| Пользователи | Оператор | Супервайзер |
| Социальные  характеристики | Мужчины, женщины  Начиная с 25 лет  Русскоязычные  Средний уровень владения компьютером | Мужчины, женщины  Начиная с 25 лет  Русскоязычные  Средний уровень владения компьютером |
| Мотивационно целевая среда | Материалы в высоком разрешении.  Возможность дополнять/редактировать данные.  Мотивация к обучению средняя | Материалы в высоком разрешении.  Мотивация к обучению высокая |
| Навыки и умения | Должны иметь средние навыки работы с компьютером | Должны иметь средние навыки работы с компьютером |
| Требования к ПО ИС | Обеспечение информацией о геофизических работах, где участвует оператор  Возможность ввода полученных данных  Возможность использовать на локальном ПК | Возможность просмотра и изменения информации о геофизических работах  Возможность использовать на локальном ПК |
| Задачи пользователя | Просмотр данных  Корректирование информации  Выполнение заказов | Формирование заказа  Проверка полученных данных |
| Рабочая среда | Стандартизированный ПК, доступ к интернету | Стандартизированный ПК, доступ к интернету |

## *2.3 Функциональные требования*

### *2.3.1 Определение функциональных возможностей программного модуля*

Для различных типов пользователей в данной системе доступны следующие функции:

Операторы базы данных имеют возможность:

– создавать новые записи;

– просматривать имеющиеся записи;

– редактировать информацию в записях;

– удалять записи.

Супервайзеры имеют доступ к следующим функциям:

– просмотр списка операторов;

– просмотр доступных записей;

– добавление новых операторов;

– редактирование данных операторов;

– создавать новые записи;

– просматривать имеющиеся записи;

– редактировать информацию в записях;

– удалять записи.

Функциональные возможности для программной системы представлены на рисунке 2.3.1.1.



Рисунок 2.3.1.1 − Функциональные возможности программного средства

### *2.3.2 Описание прецедентов*

Можно выделить для программной системы следующие успешные прецеденты:

Таблица 2.3.1 – Описание прецедента «Добавление записей в БД»

|  |  |
| --- | --- |
| Название прецедента | Добавление записей |
| Исполнитель | Оператор/ Супервайзер |
| Цель | Добавить запись в базу данных |
| Основной  успешный  сценарий | Оператор переходит в раздел добавления данных, вводит их. Система проверяет корректность данных. Система связывается с базой данных, сохраняет данные. |
| Ссылки | Уведомление пользователя о успешности операции,  добавление записи в базу данных |

Таблица 2.3.1 – Описание прецедента «Удаление операторов»

|  |  |
| --- | --- |
| Название прецедента | Удаление оператора |
| Исполнитель | Супервайзер |
| Цель | Удалить оператора из проекта |
| Основной  успешный  сценарий | Супервайзер переходит в нужный проект. Находит нужного оператора, подтверждает удаление. Система связывается с базой данных, удаляет оператора из проекта. |
| Ссылки | Уведомление Супервайзера об успешности  операции. Удаление оператора из базы данных проекта. |

Таблица 2.3.3 – Описание прецедента «Просматривать данные о заказе»

|  |  |
| --- | --- |
| Название прецедента | Просматривать данные о заказе |
| Исполнитель | Оператор/Супервайзер |
| Цель | Посмотреть заказов, которые доступные |
| Основной  успешный  сценарий | Пользователь переходит в раздел просмотра заказов. Система получает из базы данных, все доступные заказы для данного пользователя. |
| Ссылки | Вывод всех доступных записей. |

Таблица 2.3.4 – Описание прецедента «Добавление операторов»

|  |  |
| --- | --- |
| Название прецедента | Добавление оператора |
| Исполнитель | Супервайзер |
| Цель | Добавить оператора в проект |
| Основной  успешный  сценарий | Супервайзер переходит в нужный проект. Находит нужного оператора, подтверждает добавление. Система связывается с базой данных, добавляет оператора в проект. |
| Ссылки | Уведомление Супервайзера об успешности  операции. Добавление оператора в базу данных проекта |

Таблица 2.3.5 − Описание прецедента «Редактирование записей»

|  |  |
| --- | --- |
| Название прецедента | Редактирование записей |
| Исполнитель | Оператор/Супервайзер |
| Цель | Редактировать записи, которые доступные |
| Основной  успешный  сценарий | Пользователь переходит в раздел редактирование записей. Система получает из базы данных, все записи доступные для изменение. |
| Ссылки | Вывод всех доступных записей для изменения. |

Таблица 2.3.6 − Описание прецедента «Просмотр операторов»

|  |  |
| --- | --- |
| Название прецедента | Просмотр операторов |
| Исполнитель | Супервайзер |
| Цель | Просмотр операторов |
| Основной  успешный  сценарий | Пользователь переходит в раздел просмотра операторов. Система получает из базы данных, все записи о операторов. |
| Ссылки | Вывод всех операторов. |

Таблица 2.3.7 − Описание прецедента «Редактирование данных операторов»

|  |  |
| --- | --- |
| Название прецедента | Редактирование данных операторов |
| Исполнитель | Супервайзер |
| Цель | Изменить существующего оператора |
| Основной  успешный  сценарий | Пользователь переходит в раздел изменение текущего оператора. Система получает из базы данных, запись о операторе для данного пользователя. |
| Ссылки | Вывод информации о операторе. |

Таблица 2.3.8 − Описание прецедента «Удаление записей»

|  |  |
| --- | --- |
| Название прецедента | Изменение существующих записей |
| Исполнитель | Оператор/ Супервайзер |
| Цель | Удаление записи из базы данных |
| Основной  успешный  сценарий | Оператор переходит в раздел удаления данных, выбирает номер записи, присвоенной системой.. Система связывается с базой данных, удаляет данные. |
| Ссылки | Уведомление пользователя о успешности операции,  удаление записи из базы данных |

## *2.4 Нефункциональные требования*

Для корректной работы приложения предусмотрены несколько требовании к персональному ПК, на котором будут установлено данное приложение:

* Программные требования: операционная система Windows 10 или 11 для обработки архитектур х32, х64 или х86
* Аппаратные требования: оперативная память объемом не менее 4 ГБ, процессор с тактовой частотой не менее 3 ГГц, свободное место на жестком диске не менее 2 ГБ и наличие стандартных устройств ввода и вывода(монитор, мышь, клавиатура).
* Требования к надежности: программа должна адекватно реагировать на различные сбои, такие как неожиданное отключение питания компьютера, сохраняя при этом временно копию сделанной работы и визуализацию без повреждении внутренних файлов.
* Резервное копирование данных: Каждую минуту происходит резервное копирование данных. Во время непредвиденного сбоя происходит успешное восстановление данных.
* Удобство использования: Программа должна предоставлять наглядный и понятный интерфейс для пользователей.

# *3 ВЫБОР ПРОГРАММНЫХ СРЕД И СРЕДСТВ РАЗРАБОТКИ*

## *3.1 Сравнительный анализ имеющихся возможностей по выбору средств разработки*

В настоящее время на рынке представлено множество сред и инструментов для разработки настольных приложений под Windows. Они предлагают разные возможности и уровни функциональности, поэтому важно провести их сравнительный анализ.

Среди инструментов, поддерживающих язык программирования C#, стоит выделить Windows Forms (WinForms) и Windows Presentation Foundation (WPF). Оба они позволяют разработчикам создавать приложения для Windows, однако каждый из них имеет свои особенности.

Чтобы определить, какой инструмент лучше подходит для конкретного проекта, стоит учесть ряд ключевых факторов. Это скорость работы создаваемого приложения, скорость его компиляции, визуальное оформление, удобство использования самим разработчиком, функциональные возможности и доступность документации и обучающих материалов.

В таблице 3.1.1 будет представлена характеристика каждого инструмента по каждому из этих критериев.

Таблица 3.1.1 - Сравнение инструментов разработки

|  |  |  |
| --- | --- | --- |
| Критерий | WinForms | WPF |
| Скорость работы программного продукта | WinForms применяет “композитинг” для быстрого обновления экрана. | WPF обеспечивает высокую производительность за счет использования DirectX для визуализации графических элементов. |
| Скорость его компиляции | Время компиляции WinForms проекта может быть ниже, чем у WPF проекта. | Объем программы и потребление памяти в процессе работы приложения на WPF обычно несколько выше, чем на WinForms. |
| Визуальное оформление программного продукта | WinForms включает в себя набор элементов управления и возможностей для создания стильных и функциональных пользовательских интерфейсов в приложениях Windows. Однако, WinForms не предоставляет обширных возможностей для стилизации и оформления интерфейса на уровне кода, как это делают некоторые другие фреймворки и библиотеки. | WPF действительно предлагает множество возможностей для стилизации и оформления пользовательских интерфейсов. Он основан на DirectX, что позволяет разработчикам использовать 2D и 3D графику, сложные анимации и визуальные эффекты. Это делает его идеальным выбором для создания современных и интерактивных приложений. |
| Удобство использования инструментом | Разработка на WinForms является довольно простой и интуитивной. С помощью визуального редактора можно легко перетаскивать элементы управления на форму и настраивать их свойства. | WPF-приложения создаются с использованием XAML, который позволяет удобно и быстро разрабатывать интерфейс благодаря поддержке стилей и шаблонов. |
| Функциональные возможности инструмента | WinForms обеспечивает широкий спектр функциональных возможностей, в том числе поддерживает многопоточность. | WPF обеспечивает богатый набор инструментов для создания приложений, включая поддержку многопоточности, анимации, стилей, триггеров и привязки данных. |
| Качество документации и обучающих материалов | Документация Microsoft для Winforms тщательно проработана и включает в себя все аспекты разработки, начиная от базовых концепций и заканчивая продвинутыми темами. | Microsoft предлагает обширную и подробную документацию по всем аспектам разработки с использованием WPF. Она включает инструкции по работе с XAML, использованию контролов, применению механизмов привязки данных, анимации и многому другому. |

На основании проведенного сравнения между WinForms и WPF, можно сделать выводы, что WinForms является более предпочтительным выбором для создания простых программ благодаря своей простоте и удобству. В то время как WPF лучше подходит для создания сложных приложений с богатым функционалом и графическим интерфейсом.

WinForms остается идеальным решением для создания небольших программ, требующих минимального функционала, так как он прост в изучении и использовании. В то же время, WPF предоставляет более широкие возможности для создания профессиональных приложений с привлекательным и современным дизайном, но требует более глубокого знания .NET и XAML для разработки.

Таким образом, выбор между WinForms или WPF зависит от требований и сложности вашего проекта. Если вам нужно простое и быстрое решение для создания небольшой программы, то WinForms будет отличным выбором. Если же вы планируете создание более сложного приложения с богатым функционалом, то WPF может быть более подходящим вариантом

## *3.2 Характеристика выбранных программных сред и средств*

При выборе инструмента разработки отдано предпочтение WPF. WPF — это платформа для создания графических интерфейсов, разработанная Microsoft и включена в состав.NET Framework. Она позволяет разрабатывать насыщенные и интерактивные приложения с использованием XAML для определения пользовательского интерфейса. Одна из ключевых особенностей WPF — векторная графика и аппаратное ускорение, что обеспечивает высокую производительность и эффективность пользовательского опыта. Эта особенность позволяет разработчикам создавать привлекательные и функциональные приложения. Кроме того, WPF поддерживает стили и анимацию, обеспечивая создание динамичных и эстетически приятных интерфейсов. Еще одна важная особенность WPF — открытый исходный код с активным сообществом разработчиков, что гарантирует непрерывное развитие и улучшение платформы. Также WPF ориентирован на архитектурный шаблон MVVM (разделение логики приложения и интерфейса), облегчая чтение, тестирование и поддержку кода. Кроме того, WPF интегрируется с другими платформами.NET, такими как ASP.NET или Windows Communication Foundation, что позволяет создавать кросс-платформенные продукты с минимальными усилиями. Таким образом, WPF — это мощный и гибкий инструмент для разработки современных, производительных и стильных приложений для Windows, который мы выбрали для нашего нового проекта.

# *4 АЛГОРИТМ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ*

## *4.1 Этапы реализации ПМ*

Для реализации программного средства, необходимо выполнить действия в следующем порядке:

1. Изучение предметной области РБД.
2. Реализация механизма хранения базы данных.
3. Создание механизмов для поддержания целостности данных:
4. Разработка правил и ограничений для корректного ввода, обновления данных.
5. Реализация каскадного удаления и обновления данных.
6. Создание РБД для работы ЗБС:
7. Проектирование структур таблиц.
8. Определение связей между сущностями.
9. Создание интерфейса для работы с РБД:
10. Разработка пользовательского интерфейса для добавления, изменения и удаления данных;
11. Реализация запросов и отчётов для анализа данных.

## *4.2 Пользовательский интерфейс ПМ*

### *4.2.1 Взаимодействие пользователей с ПМ*

Программное средство обеспечивает взаимодействие пользователя с системой посредством предоставления следующих возможностей:

* Инициализация работы − пользователю при запуски программы предлагается войти в свой аккаунт. После чего он получает доступ к экрану выбора геофизической работы, в которой он участвует.
* Ввод геофизических данных: оператору на экране ввода данных предлагается ввести данные и инициировать их обработку и запись.
* Изменение геофизических данных: оператор может вносить изменения в загруженные данные (редактировать или удалять их).
* Экспорт геофизических данных: оператор имеет возможность экспортировать данные в разных форматах файлов TXT и XLS. Выбирается формат и указывается место сохранения файла.
* Изменение данных операторов: Супервайзер может вносить изменения в данные операторов.
* Формирование отчета − на основе полученных данных обработчик данных может создавать различные отчеты, например, статистические, графические или текстовые

Можно представить диаграмму активности для экспорта геофизических данных на рисунок 4.2.1.1:

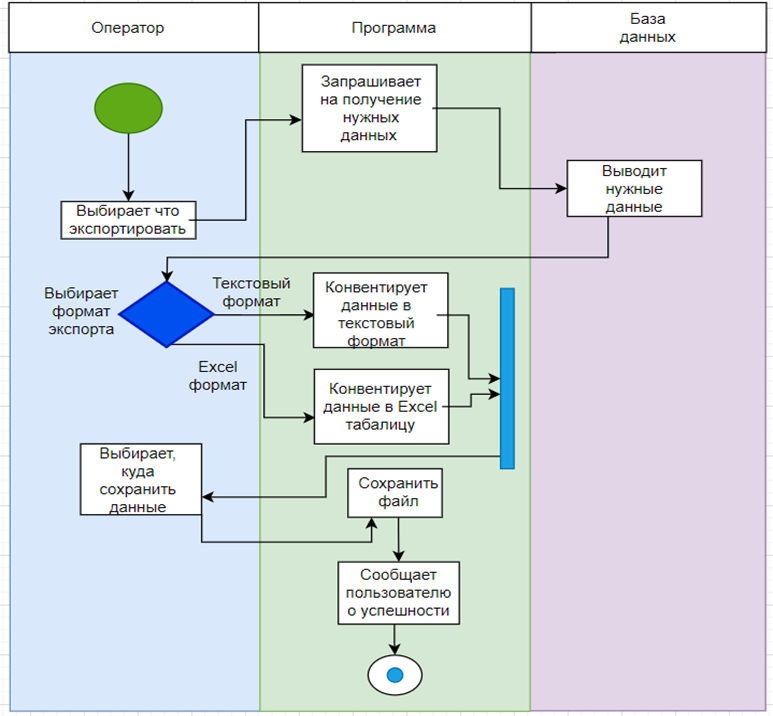


Рисунок 4.2.1.1 − Диаграмму активности для экспорта геофизических данных

Можно представить отчёт о проекте на рисунке 4.2.1.2:

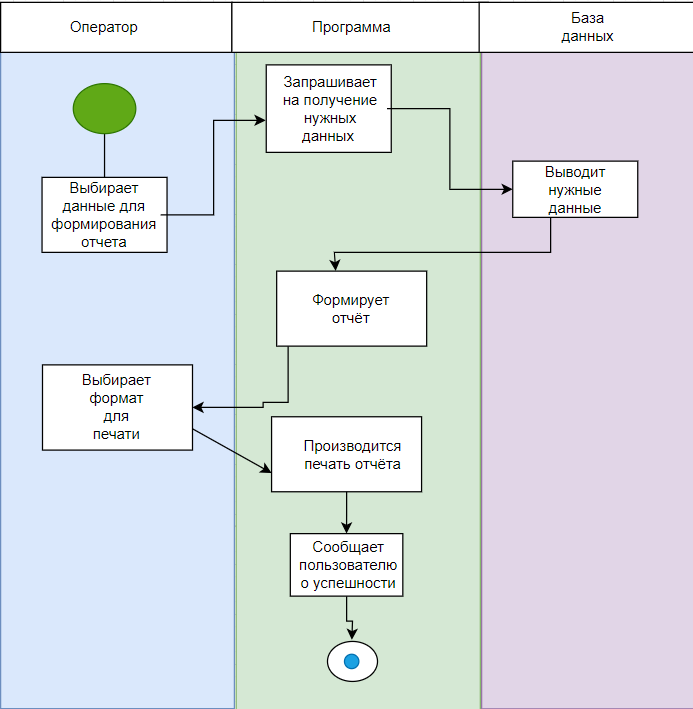


Рисунок 4.2.1.2 − Диаграмма создания отчёта

### *4.2.2 Проектирование пользовательских сценариев*

Программная система поддерживает несколько типичных пользовательских сценариев для разных категорий пользователей.

Для Супервайзера характерны следующие альтернативные сценарии:

* Супервайзер создал новый проект и распределил задачи между операторами.
* Супервайзер принял от оператора полученные данные и обнаружил ошибку, которую необходимо исправить.

Для оператора возможны такие альтернативные сценарии:

* внести результаты измерений: войти в нужную базу данных, выбрать нужный пикет, открыть экран добавления записи и создать новую запись;
* удалить некоторые результаты измерений: войти в нужную базу данных, выбрать нужную форму, выбрать найти нужную запись через поисковик и нажать кнопку удаления.

Можно привести следующую диаграмму последовательности для сценария ввода данных в базу данных на рисунке 4.2.2.1

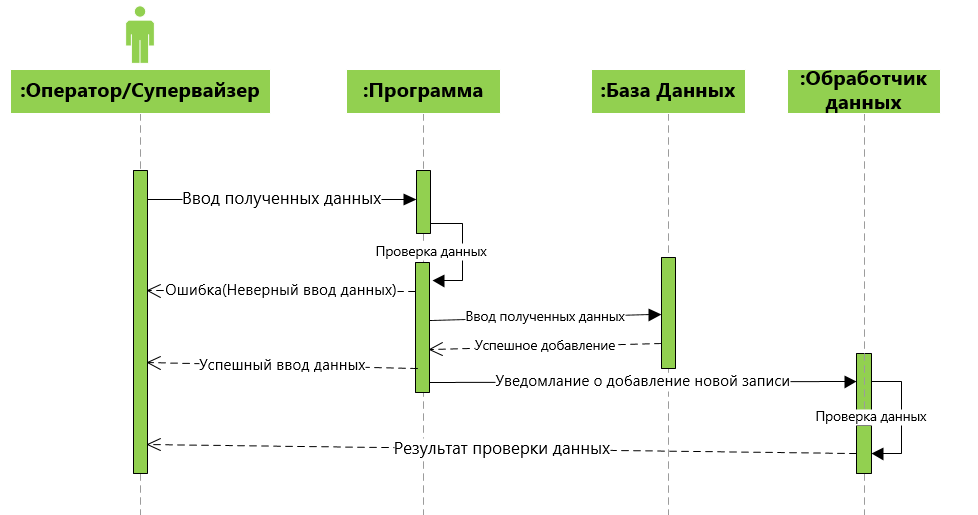


Рисунок 4.2.2.1 − Диаграмма последовательности

### *4.2.3 Определение операций пользователей*

В данном программном средстве при анализе предметной области с учете характеристик пользователей можно выделить следующие операции, которые пользователи могут совершать:

1. Просмотр списка проектов
2. Регистрация проекта
3. Просмотр информации о проекте
4. Сохранение проекта
5. Просмотр списка профилей
6. Регистрация профиля
7. Просмотр информации о профили
8. Сохранение профиля
9. Просмотр списка площадей
10. Регистрация площади
11. Просмотр информации о площади
12. Сохранение площади
13. Просмотр списка точек записи
14. Регистрация точки записи
15. Просмотр информации о точки записи
16. Сохранение точки записи
17. Просмотр списка операторов
18. Регистрация оператора
19. Просмотр информации об операторе
20. Сохранение оператора
21. Просмотр списка клиентов
22. Регистрация клиента
23. Просмотр информации о клиенте
24. Сохранение клиента
25. Просмотр списка руководителей
26. Регистрация Супервайзера
27. Просмотр информации о руководителе
28. Сохранение Супервайзера
29. Просмотр списка измерительного оборудования
30. Регистрация измерительного оборудования
31. Просмотр информации об оборудовании
32. Сохранение оборудования
33. Просмотр списка обработчиков измерений
34. Регистрация обработчиков измерений
35. Просмотр информации о обработчике измерений
36. Сохранение обработчиков измерений
37. Просмотр списка полевых отрядов
38. Регистрация полевого отряда
39. Просмотр информации о полевом отряде
40. Сохранение полевого отряда
41. Просмотр списка результатов измерений
42. Запись результатов измерений
43. Просмотр информации о результате
44. Сохранение результатов измерений
45. Визуализация результатов измерений в виде графика ЭДС
46. Визуализация результатов измерений в виде изолиний ЭДС по

профилю

1. Визуализация измеренных кривых в виде графика ЭДС по профилю
2. Визуализация рядовых и контрольных измерений в виде графика

ЭДС по профилю

1. Визуализация рядовых и опытно-методических измерений в виде

графика ЭДС по профилю

1. Визуализация результатов измерений в виде изолиний ЭДС по

проекту

1. Визуализация измеренных кривых в виде графика ЭДС по проекту
2. Визуализация рядовых и контрольных измерений в виде графика

ЭДС по проекту

1. Визуализация рядовых и опытно-методических измерений в виде

графика ЭДС по проекту

1. Вывод отчета о расхождениях
2. Экспортирование отчета о расхождениях

### *4.2.4 Составление функциональных блоков*

Все вышеперечисленные операции можно разделить на несколько функциональных блоков, которые подразделяются в зависимости от работы пользователей с информацией:

1. Блок по проекту - данный блок определяет операции 1-4
2. Блок по площади - данный блок определяет операции 9-12, 45-48 и 54-55
3. Блок по профилю - данный блок определяет операции 5-8 и 50-55
4. Блок по точкам записей - - данный блок определяет операции 13-16
5. Блок по измерительному оборудованию - данный блок определяет операции 29-32
6. Блок по оператору - данный блок определяет операции 17-20
7. Блок по супервайзеру - данный блок определяет операции 25-28
8. Блок по клиенту - данный блок определяет операции 21-24
9. Блок по обработчику измерений- данный блок определяет операции 33-36
10. Блок по полевому отряду - данный блок определяет операции 37-40
11. Блок по результатом измерений - данный блок определяет операции 41-44

### *4.2.5 Проектирование структуры экранов (ПМ) и схемы навигации*

После тщательного изучения предметной области, получения данных о пользователях и определения потенциальных вариантов их действий, также учитывая набор функций будущего приложения, мы можем приступить к разработке структуры системы навигации. Представлено на рисунке 4.2.5.1.

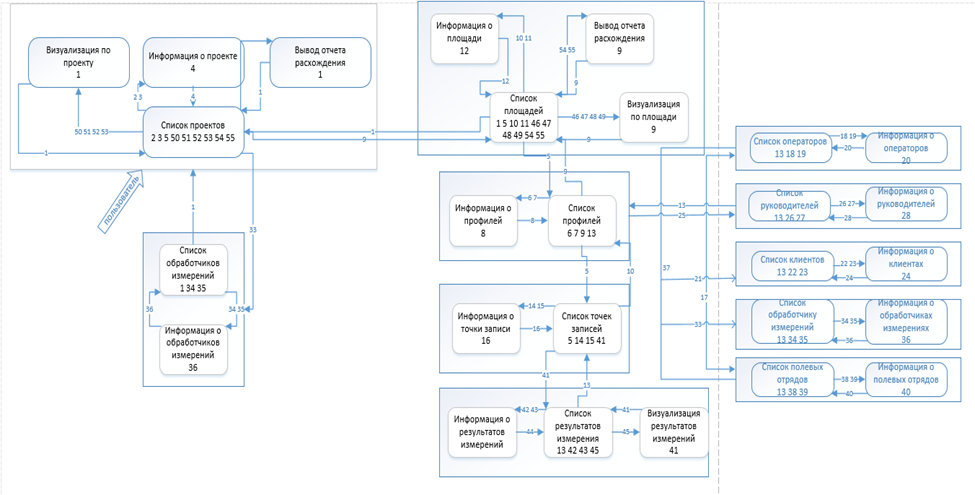


Рисунок 4.2.5.1. − Структура экранов программы

### *4.2.6 Низкоуровневое проектирование*

Пройдя этап планирования и высокоуровневого проектирования интерфейса пользователя, то есть, изучив предметную область, определив типы пользователей и их действия, разработав структуру навигации и установив основные функции, можно создать макеты экранных форм для каждого из функциональных блоков. Сами макеты представлены на рисунках 4.2.1 - 4.2.5



Рисунок 4.2.1 - Экран авторизации пользователя в системе

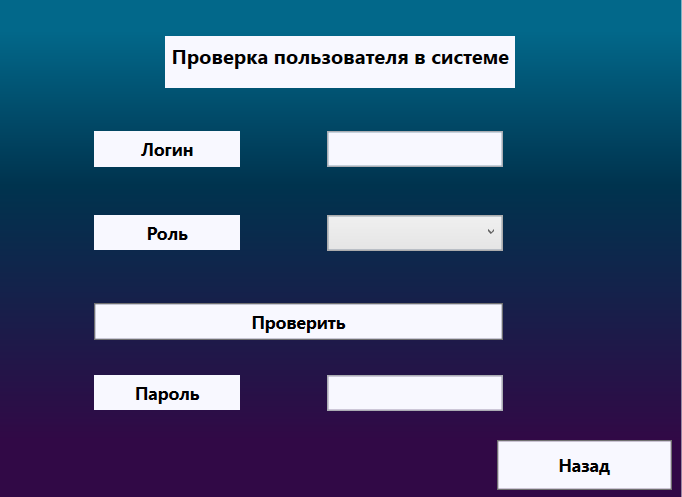


Рисунок 4.2.2 – Проверка пароля

2

6

5

3

4

1

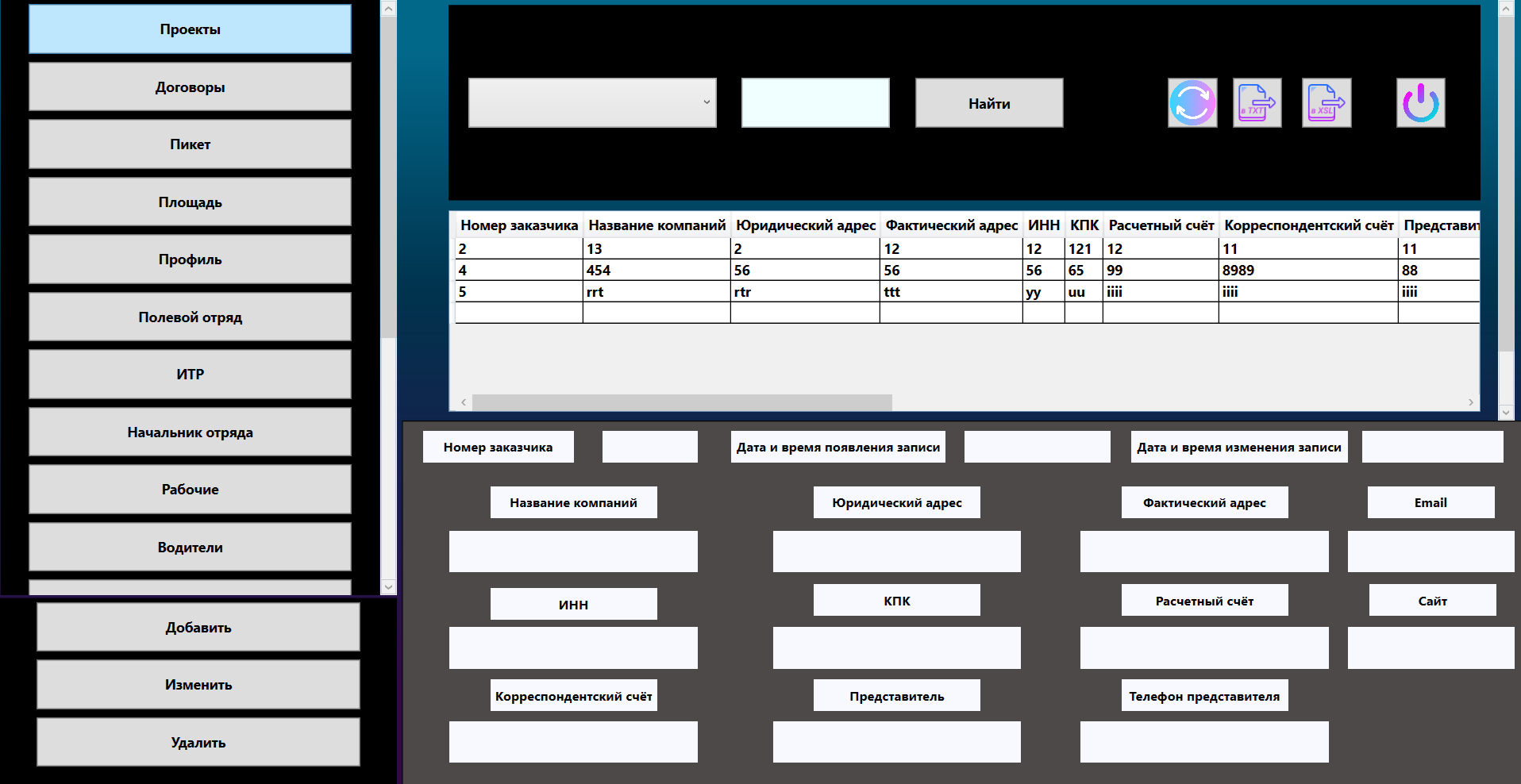


Рисунок 4.2.3 - Основной экран программы

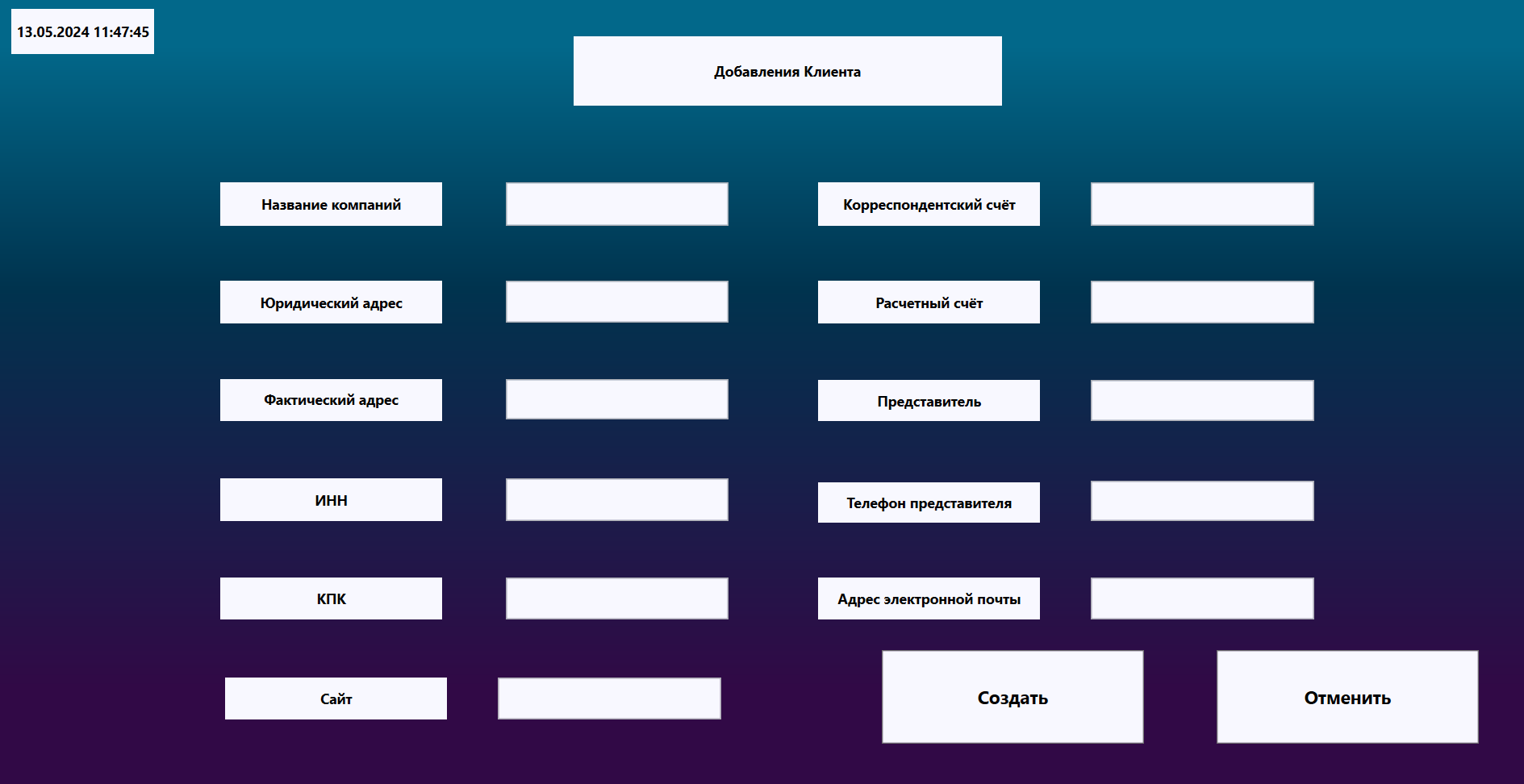


Рисунок 4.2.4 – Экран Добавления Записи

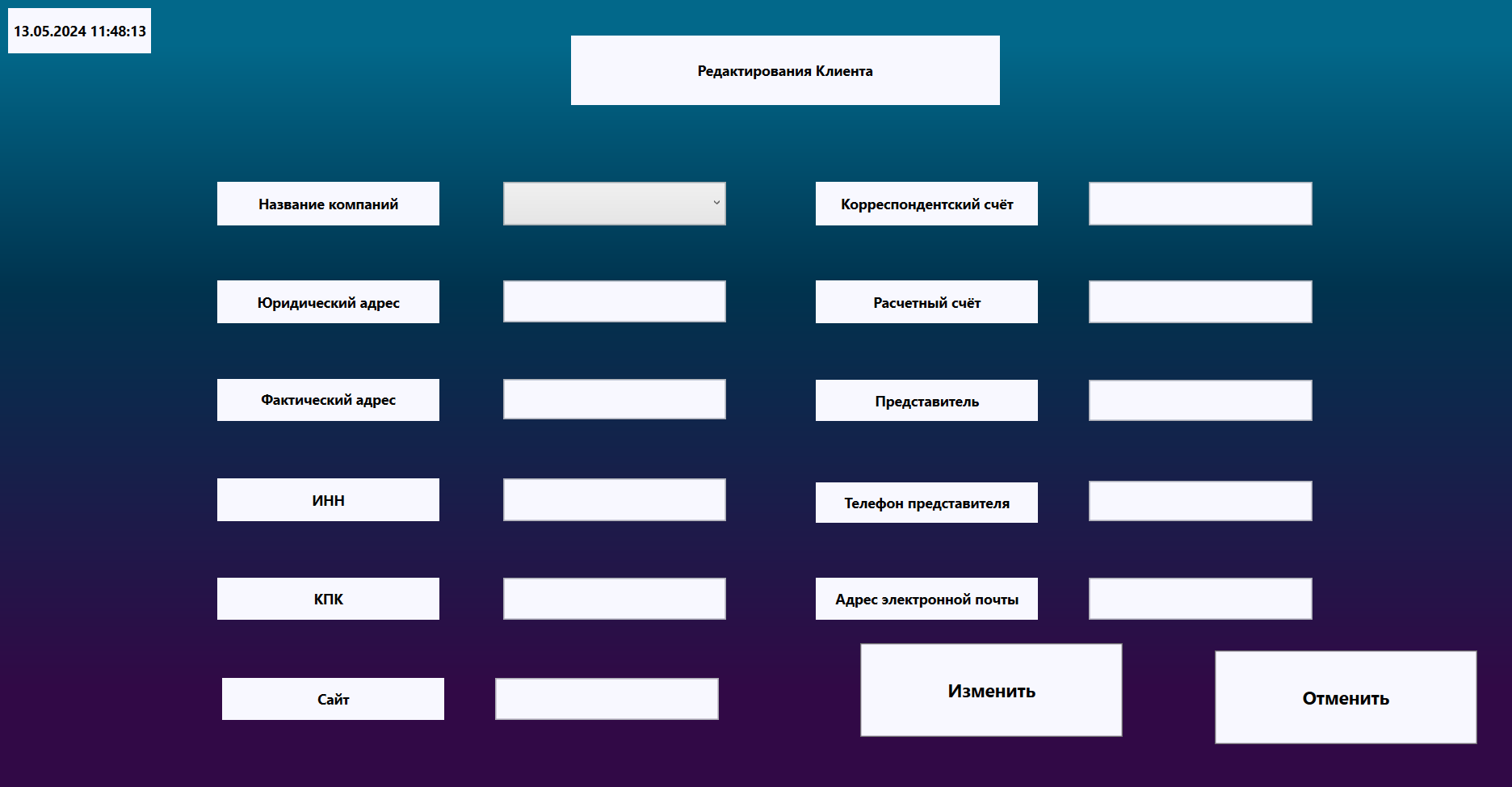


Рисунок 4.2.5 - Экран Редактирования Записи

## *4.3 Входные, выходные и промежуточные данные*

Для разных задач могут быть разные входные и выходные данные.

При регистрации проекта требуются следующие данные:

● Заказчик

● Представитель заказчика

● Контакты

● Количество площадей

● Названия площадей

При регистрации площади требуются следующие данные:

● Координаты углов периметра

При регистрации профилей требуются следующие данные:

● Проектные координаты начала линий съёмки

● Проектные координаты конца линий съёмки

● Проектные координаты точек изломов

При регистрации точки записи требуются следующие данные:

● Вид точек измерения

● Координаты точки

При регистрации измерительной аппаратуры требуются следующие данные:

● Название аппаратуры

## *4.4 Разработка базы данных, реализуемой в рамках (ПМ)*

В данном программном средстве используется иерархическая модель данных. Она содержит 19 таблиц, корнем которых является таблица “Проект”. Структура всех таблиц (таблица 4.4.1 - 4.4.17) описана ниже:

Таблица 4.4.1 - Описание полей таблицы “Заказчик”

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Свойства | Особенность |
| Номер заказчик | Числовое | Длинное целое | Обязательное |
| Название компаний | Текстовый | Длина 100 символов | Обязательное |
| Юридический адрес | Текстовый | Длина 100 символов | Обязательное |
| Фактический адрес | Текстовый | Длина 100 символов | Обязательное |
| ИНН | Текстовый | Длина 14 символов | Обязательное |
| КПК | Текстовый | Длина 10 символов | Обязательное |
| Расчетный счёт | Текстовый | Длина 23 символов | Обязательное |
| Коррекционный счёт | Текстовый | Длина 23 символов | Обязательное |
| Представитель | Текстовый | Длина 100 символов | Обязательное |
| Телефон представителя | Текстовый | Длина 15 символов | Обязательное |
| Адрес электронной почты | Текстовый | Длина 100 символов | Обязательное |
| Сайт | Текстовый | Длина 100 символов | Обязательное |
| Дата и время появления записи | Дата и время | ГГГГ-ММ-ДД чч:мм:сс[.ннн] | Необязательно |
| Дата и время изменения записи | Дата и время | ГГГГ-ММ-ДД чч:мм:сс[.ннн] | Необязательно |

Таблица 4.4.2 − Описание полей таблицы “Проект”

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Свойства | Особенность |
| Номер проекта | Числовое | Длинное целое | Обязательное |
| Наименование проекта | Текстовый | Длина 100 символов | Обязательное |
| Номер площади | Числовое | Длинное целое | Обязательное |
| Номер договора | Числовое | Длинное целое | Обязательное |
| Номер заказчик | Числовое | Длинное целое | Обязательное |
| Дата и время появления записи | Дата и время | ГГГГ-ММ-ДД чч:мм:сс[.ннн] | Необязательно |
| Дата и время изменения записи | Дата и время | ГГГГ-ММ-ДД чч:мм:сс[.ннн] | Необязательно |

Таблица 4.4.3 − Описание полей таблицы “Площади”

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Свойства | Особенность |
| Номер площади | Числовое | Длинное целое | Обязательное |
| Наименование площади | Текстовый | Длина 100 символов | Обязательное |
| Номер профиля | Числовое | Длинное целое | Обязательное |
| Список координат углов периметра | Текстовый | Длина 100 символов | Обязательное |
| Длина периметра | Числовое | Вещественное | Числовое |
| Величина площади окружённая периметром | Числовое | Вещественное | Числовое |
| Дата и время начала работ | Дата и время | ГГГГ-ММ-ДД чч:мм:сс[.ннн] | Необязательно |
| Дата и время окончания работ | Дата и время | ГГГГ-ММ-ДД чч:мм:сс[.ннн] | Необязательно |
| Дата и время появления записи | Дата и время | ГГГГ-ММ-ДД чч:мм:сс[.ннн] | Необязательно |
| Дата и время изменения записи | Дата и время | ГГГГ-ММ-ДД чч:мм:сс[.ннн] | Необязательно |
| Супервайзер полевых работ | Числовое | Длинное целое | Обязательное |
| Супервайзер обработки данных | Числовое | Длинное целое | Обязательное |

Таблица 4.4.4 − Описание полей таблицы “Договор”

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Свойства | Особенность |
| Номер договора | Числовое | Длинное целое | Обязательное |
| Наименование договора | Текстовый | Длина 30 символов | Обязательное |
| Начала | Дата | ГГГГ-ММ-ДД | Обязательно |
| Окончания | Дата | ГГГГ-ММ-ДД | Обязательно |
| Стоимость | Числовое | Длинное целое | Обязательное |
| Дата и время появления записи | Дата и время | ГГГГ-ММ-ДД чч:мм:сс[.ннн] | Необязательно |
| Дата и время изменения записи | Дата и время | ГГГГ-ММ-ДД чч:мм:сс[.ннн] | Необязательно |

Таблица 4.4.5 − Описание полей таблицы “Профиль”

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Свойства | Особенность |
| Номер профиля | Числовое | Длинное целое | Обязательное |
| Наименование профиля | Текстовый | Длина 30 символов | Обязательное |
| Номер пикета | Числовое | Длинное целое | Обязательное |
| Координаты начала | Числовое | Вещественное | Обязательное |
| Координаты изломов | Числовое | Вещественное | Обязательное |
| Координаты окончания | Числовое | Вещественное | Обязательное |
| Длина | Числовое | Вещественное | Обязательное |
| Дата и время начала работ | Дата и время | ГГГГ-ММ-ДД чч:мм:сс[.ннн] | Обязательно |
| Дата и время окончания работ | Дата и время | ГГГГ-ММ-ДД чч:мм:сс[.ннн] | Необязательно |
| Дата и время появления записи | Дата и время | ГГГГ-ММ-ДД чч:мм:сс[.ннн] | Обязательно |
| Дата и время изменения записи | Дата и время | ГГГГ-ММ-ДД чч:мм:сс[.ннн] | Необязательно |

Таблица 4.4.6 − Описание полей таблицы “Пикет”

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Свойства | Особенность |
| Номер пикета | Числовое | Длинное целое | Обязательное |
| Наименование пикета | Текстовый | Длина 50 символов | Обязательное |
| Координата | Числовое | Вещественное | Обязательное |
| Вид измерения (рядовое, контрольное, опытное) | Текстовый | Длина 30 символов | Обязательное |
| Окончательный результат измерения (кривая измерения) | Текстовый | Длина 30 символов | Обязательное |
| Номер отряда | Числовое | Длинное целое | Обязательное |
| Номер методики | Числовое | Длинное целое | Обязательное |
| Номер оборудования | Числовое | Длинное целое | Обязательное |
| Трансформанта измерения 1 | Текстовый | Длина 30 символов | Обязательное |
| Трансформанта измерения 2 | Текстовый | Длина 30 символов | Обязательное |
| Трансформанта измерения 3 | Текстовый | Длина 30 символов | Обязательное |
| Промежуточные результаты измерения | Текстовый | Длина 30 символов | Обязательное |
| Промежуточные результаты измерения 1 | Текстовый | Длина 30 символов | Обязательное |
| Промежуточные результаты измерения 2 | Текстовый | Длина 30 символов | Обязательное |
| Промежуточные результаты измерения 3 | Текстовый | Длина 30 символов | Обязательное |

Таблица 4.4.7 − Описание полей таблицы “Методика”

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Свойства | Особенность |
| Номер методики | Числовое | Длинное целое | Обязательное |
| Наименование методики | Текстовая | Длина 100 символов | Обязательное |
| Описание генераторной установки (размеры) | Текстовый | Длина 20 символов | Обязательное |
| Описание измерительной установки (размеры, разносы) | Текстовый | Длина 50 символов | Обязательное |
| Описание телеметрической установки | Текстовый | Длина 100 символов | Обязательное |
| Продолжительность импульса | Текстовый | Длина 10 символов | Обязательное |
| Продолжительность паузы, | Текстовый | Длина 10 символов | Обязательное |
| Сила тока | Числовое | Вещественное | Обязательное |

Таблица 4.4.8 − Описание полей таблицы “Полевой отряд”

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Свойства | Особенность |
| Номер отряда | Числовое | Длинное целое | Обязательное |
| Начальник отряда | Числовое | Длинное целое | Обязательное |
| ИТР (инженерно-технические работники) | Числовое | Длинное целое | Обязательное |
| Водители | Числовое | Длинное целое | Обязательное |
| Рабочие | Числовое | Длинное целое | Обязательное |
| Номер Супервайзера | Числовое | Длинное целое | Обязательное |

Таблица 4.4.9 − Описание полей таблицы “ИТР”

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Свойства | Особенность |
| Номер ИТР | Числовое | Длинное целое | Обязательное |
| Квалификация | Текстовая | Длина 100 символов | Обязательное |
| Опыт работы в коллективе | Текстовая | Длина 25 символов | Обязательное |
| Общий опыт работы по специальности | Текстовая | Длина 50 символов | Обязательное |
| Дата прохождения медосмотра | Дата | ГГГГ-ММ-ДД | Обязательно |

Таблица 4.4.10 − Описание полей таблицы “Рабочие”

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Свойства | Особенность |
| Номер Рабочего | Числовое | Длинное целое | Обязательное |
| Квалификация | Текстовая | Длина 100 символов | Обязательное |
| Опыт работы в коллективе | Текстовая | Длина 25 символов | Обязательное |
| Общий опыт работы по специальности | Текстовая | Длина 50 символов | Обязательное |
| Дата прохождения медосмотра | Дата | ГГГГ-ММ-ДД | Обязательно |

Таблица 4.4.11 − Описание полей таблицы “Водители”

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Свойства | Особенность |
| Номер Водителя | Числовое | Длинное целое | Обязательное |
| Квалификация | Текстовая | Длина 100 символов | Обязательное |
| Опыт работы в коллективе | Текстовая | Длина 25 символов | Обязательное |
| Общий опыт работы по специальности | Текстовая | Длина 50 символов | Обязательное |
| Дата прохождения медосмотра | Дата | ГГГГ-ММ-ДД | Обязательно |

Таблица 4.4.12 − Описание полей таблицы “Супервайзер”

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Свойства | Особенность |
| Номер Супервайзера | Числовое | Длинное целое | Обязательное |
| Квалификация | Текстовая | Длина 100 символов | Обязательное |
| Опыт работы в коллективе | Текстовая | Длина 25 символов | Обязательное |
| Общий опыт работы по специальности | Текстовая | Длина 50 символов | Обязательное |
| Дата прохождения медосмотра | Дата | ГГГГ-ММ-ДД | Обязательно |

Таблица 4.4.13 − Описание полей таблицы “Начальник отряда”

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Свойства | Особенность |
| Номер Начальника | Числовое | Длинное целое | Обязательное |
| Квалификация | Текстовая | Длина 100 символов | Обязательное |
| Опыт работы в коллективе | Текстовая | Длина 25 символов | Обязательное |
| Общий опыт работы по специальности | Текстовая | Длина 50 символов | Обязательное |
| Дата прохождения медосмотра | Дата | ГГГГ-ММ-ДД | Обязательно |

Таблица 4.4.14 − Описание полей таблицы “Оборудования”

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Свойства | Особенность |
| Номер Оборудования | Числовое | Длинное целое | Обязательное |
| Номер генеративного оборудования | Числовое | Длинное целое | Обязательное |
| Номер измерительного оборудования | Числовое | Длинное целое | Обязательное |
| Номер телеметрического оборудования | Числовое | Длинное целое | Обязательное |

Таблица 4.4.15 − Описание полей таблицы “Генеративное оборудование”

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Свойства | Особенность |
| Номер генеративного оборудования | Числовое | Длинное целое | Обязательное |
| Наименование | Текстовая | Длина 100 символов | Обязательное |
| Инвентарный номер | Текстовая | Длина 100 символов | Обязательное |
| Дата приобретения | Дата | ГГГГ-ММ-ДД | Обязательно |
| Дата поверки | Дата | ГГГГ-ММ-ДД | Обязательно |
| Характеристики | Текстовая | Длина 100 символов | Обязательное |

Таблица 4.4.16 − Описание полей таблицы “Измерительное оборудование”

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Свойства | Особенность |
| Номер измерительного оборудования | Числовое | Длинное целое | Обязательное |
| Наименование | Текстовая | Длина 100 символов | Обязательное |
| Инвентарный номер | Текстовая | Длина 100 символов | Обязательное |
| Дата приобретения | Дата | ГГГГ-ММ-ДД | Обязательно |
| Дата поверки | Дата | ГГГГ-ММ-ДД | Обязательно |
| Характеристики | Текстовая | Длина 100 символов | Обязательное |

Таблица 4.4.17 − Описание полей таблицы “Телеметрическое оборудование”

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Свойства | Особенность |
| Номер телеметрического оборудования | Числовое | Длинное целое | Обязательное |
| Наименование | Текстовая | Длина 100 символов | Обязательное |
| Инвентарный номер | Текстовая | Длина 100 символов | Обязательное |
| Дата приобретения | Дата | ГГГГ-ММ-ДД | Обязательно |
| Дата поверки | Дата | ГГГГ-ММ-ДД | Обязательно |
| Характеристики | Текстовая | Длина 100 символов | Обязательное |

Также представлена схема сокращенной базы данных на рисунке 4.4.1. Полную схему базы данных можно посмотреть в приложений В.

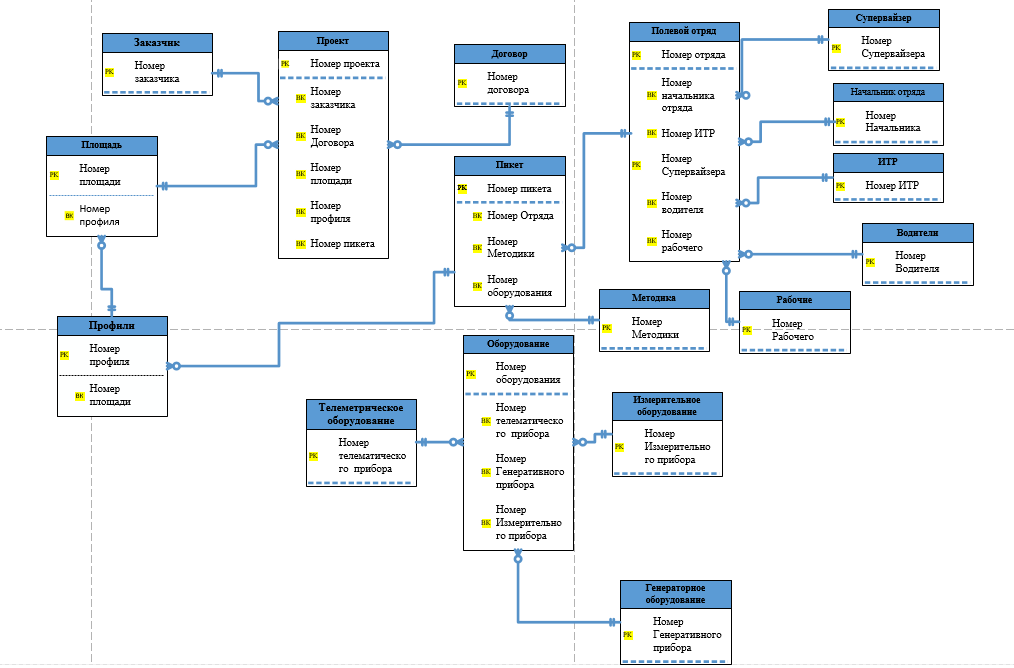


Рисунок 4.4.1 - Сокращённая схема базы данных

## *4.5 Алгоритмы передачи данных*

Клиент-серверная архитектура приложения: пользователь вводит данные через графический интерфейс, клиентская программа отправляет их через строку подключения к базе данных через запросы на сервер базы данных. Сервер базы данных обрабатывает данные, анализирует их и отправляет запрос на сервер приложения. Данные отправляются в двоичном формате, сервер приложения получает команды от сервера базы данных, исполняет запросы и отправляет результаты обратно пользователю. Во время исполнения запросов данные могут изменяться на носителе: сервер базы данных вызывает функции операционной системы и драйверов накопителей для модификации данных. Если сервер базы данных и сервер приложения находятся на одном хосте, используются механизмы межпроцессного взаимодействия (общая память, именованные каналы и др.).

## *4.6 Архитектура программной системы*

Программное обеспечение состоит из двух ключевых элементов: базы данных и пользовательского интерфейса. Их структура представлена на UML-диаграмме компонентов (рисунок 4.6.1).

База данных содержит все таблицы, подключённые к сетевой базе данных. С её помощью можно вносить изменения, удалять и просматривать данные.

Пользовательский интерфейс обеспечивает взаимодействие с базой данных, позволяя просматривать, удалять, добавлять и редактировать информацию, хранящуюся в ней.

Пользовательский интерфейс отправляет запросы к базе данных, которые затем обрабатываются и возвращаются пользователю в виде результатов.

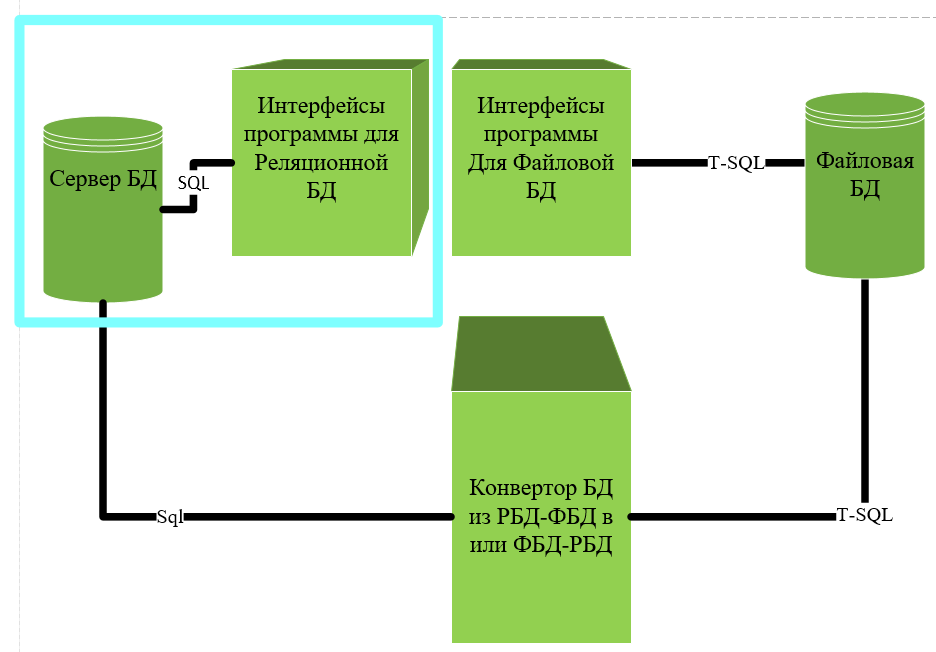


Рисунок 4.6.1– Диаграмма развертывания системы

## *4.7 Архитектура программного кода модуля*

**Диаграмма классов.**

Для данного программного модуля можно представить диаграмму классов рисунок 4.7.1. Она иллюстрирует операции и атрибуты классов и используется для изучения концепций области, понимания требований к программному обеспечению и описания подробных проектов.

# *5 ТЕСТИРОВАНИЕ И ОПТИМИЗАЦИЯМ*

## *5.1 План тестирования*

Для тестирования программы мы должны разработать план, включающий следующие основные виды тестирования:

* Функциональное тестирование: проверка корректности выполнения всех заявленных функций и возможностей программы.
* Тестирование удобства использования: оценка простоты и интуитивности интерфейса, скорости обучения и общего восприятия программы пользователем.
* Тестирование производительности: измерение скорости и эффективности работы программы при разных нагрузках и условиях использования.
* Тестирование установки: проверка процесса установки программы на различных платформах и версиях операционной системы, а также проверка работы после обновления или переустановки.

Функциональное тестирование проверяет следующие пункты:

1. Работа с базой данных

2. Работа основных функции

2.1. Авторизация

2.2. Навигация между окнами

2.3. Сохранение данных в базе данных

3. Корректная обработка ошибок.

4. Работа приложения на фоне

Тестирование удобство использования проверяет следующие пункты:

1. Корректный отображение элементов с различными разрешениями экрана

2. Отсутствие орфографических и грамматических ошибок

3. Корректные заголовки экранов

4. Корректный возврат на предыдущий экран

5. Информативные ошибки, подсказки

Тестирование производительности проверяет следующие пункты:

1. Время выполнения запросов в базу данных

2. Время загрузки приложения

Тестирование установки проверяет следующие пункты:

1. Запускает ли приложение после установки

2. Корректность установки с разными конфигурациями системы

3. Время установки

## *5.2 Результаты тестирования*

Тестирование проводится на компьютере со следующими характеристиками: ОС - Windows 11; Процессор - Intel Core i3-10110U; Оперативная память - 8 гб; SSD диск с объемом - 256 гб. Все результаты тестов расписаны в таблице 5.2.1

Таблица 5.2.1 - Тестирование

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название | Результат | Подробнее |
| 1 | Запуск приложения | Положительный |  |
| 2 | Работа с файлами | Положительный |  |
| 3 | Вход в программу | Положительный |  |
| 4 | Навигация между окнами | Отрицательный | Нет никаких навигационных кнопок, для возвращения на предыдущую страницу. Нет кнопок смены базы данных. Пример - рисунок 5.2.2 |
| 5 | Сохранение данных в базе данных | Положительный |  |
| 6 | Корректная обработка ошибок | Отрицательный | Есть ошибки, которые не обрабатываются. Пример - неправильный формат импорта. Рисунок 5.2.3 |
| 7 | Работа приложения на фоне | Положительный |  |
| 8 | Корректный отображение элементов с различными разрешениями экрана | Отрицательный | Некоторые элементы изменяются некорректно. Пример - рисунок 5.2.4 |
| 9 | Отсутствие орфографических и грамматических ошибок | Положительный |  |
| 10 | Корректные заголовки экранов | Положительный |  |
| 11 | Информативные ошибки, подсказки | Положительный | Все подсказки которые присутствуют в программе информативные. |
| 12 | Время выполнения запросов в базу данных | Положительный | Время на ПК:   * Изменения: 15 мс. * Добавления: 20 мс. * Удаление: 30 мс. |
| 13 | Время загрузки приложения | Положительный | Время загрузки на ПК = 1 сек. |
| 14 | Запускает ли приложение после установки | Положительный |  |
| 15 | Корректность установки с разными конфигурациями системы | Положительный |  |
| 16 | Время установки | Положительный | Время установки на ПК = 30 сек. С учетом настроек = 1 мин. 30 сек. |

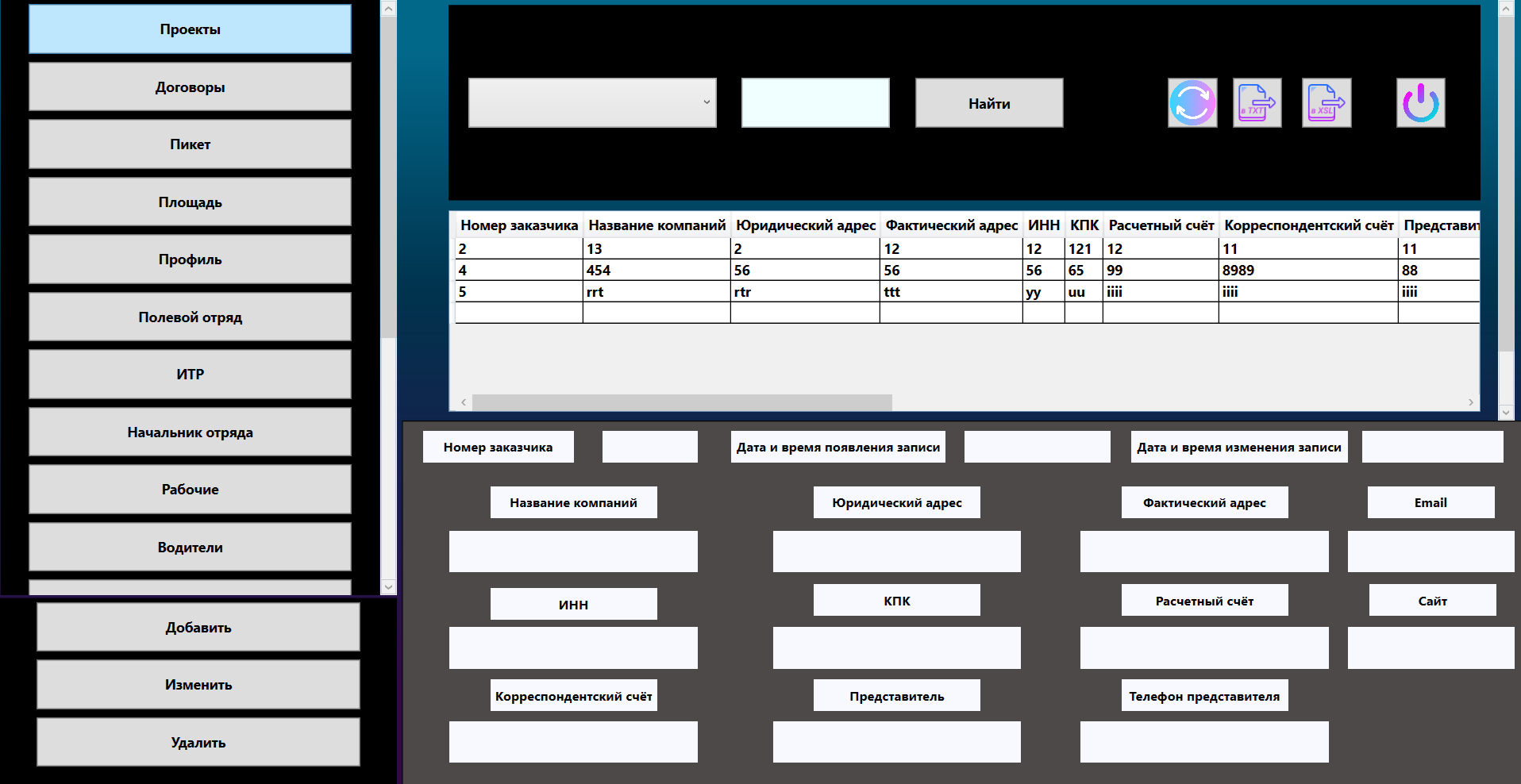


Рисунок 5.2.2 - стандартный экран

Из результатов тестирования, проделанных выше, можно сделать заключение, что основное внимание нужно уделить оптимизации пользовательского интерфейса. Основная программа и база данных функционируют без сбоев.

## *Оптимизация ПМ*

На данном этапе происходит оптимизация тех задач, которые не прошли тестирования, а именно: создание корректных обработок ошибок; создание навигаций между окнами; реализовать корректное отображение элементов с различными разрешениями экрана.

Для создания корректных обработок ошибок были добавлены проверки на наличие ошибок и исключений в некоторых точках программы. А именно: при экспорте базы данных в файл-xlsx; при экспорте базы данных в файл-txt; Итог изображен на рисунке 5.3.1.

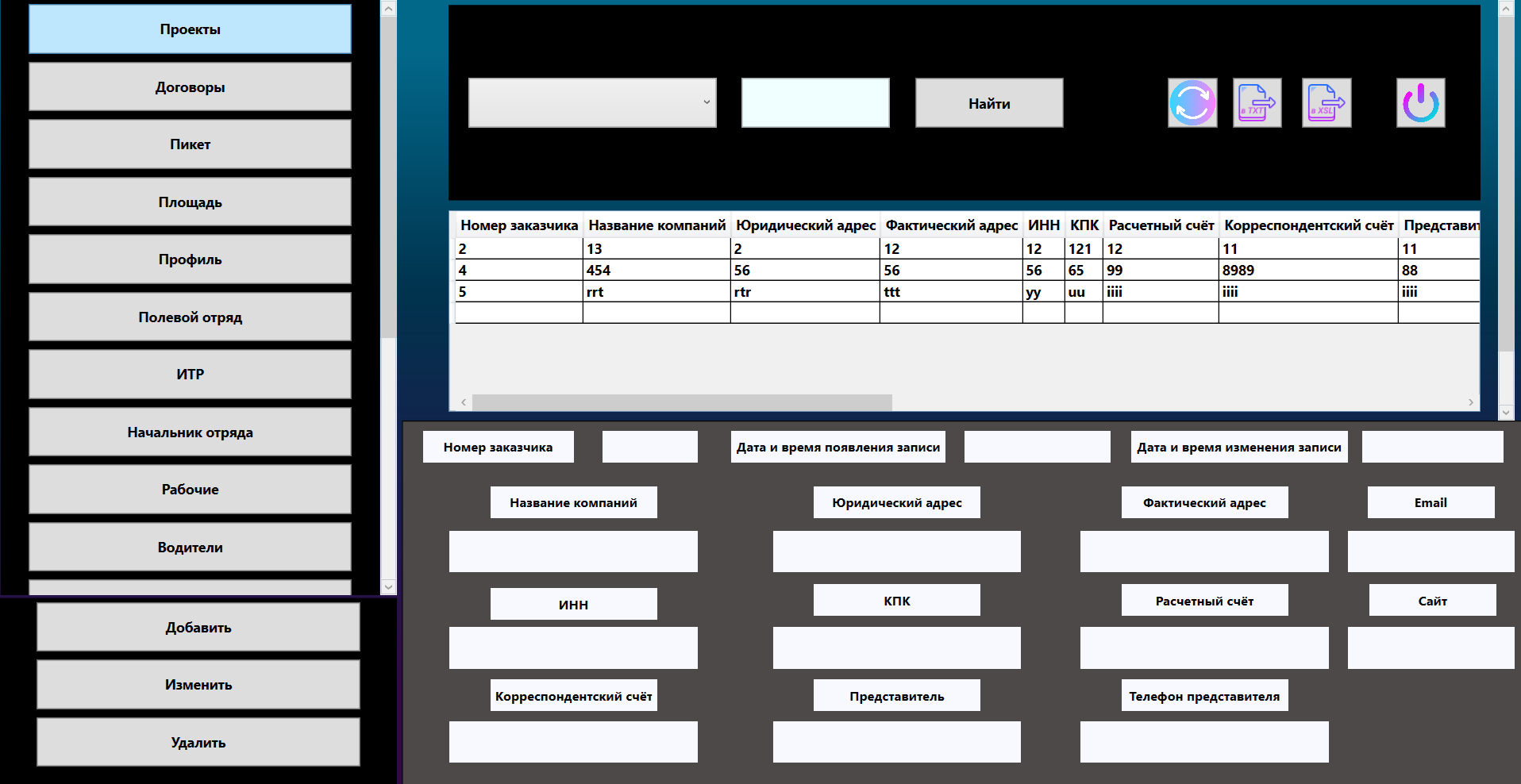
****

Рисунок 5.3.1 - стандартный экран

***6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ***

## *6.1 Установка приложения*

Для установки приложения требуется использовать установочный файл приложения. После его запуска откроется окно с выбором языка (см. рисунок 6.1.1). Затем необходимо указать путь для установки программы (см. рисунок 6.1.2). После чего мы продолжаем, пока не пойдет установка. На этого установка будет завершена.

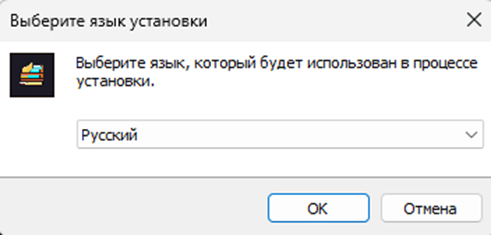


Рисунок 6.1.1 – Установка языка

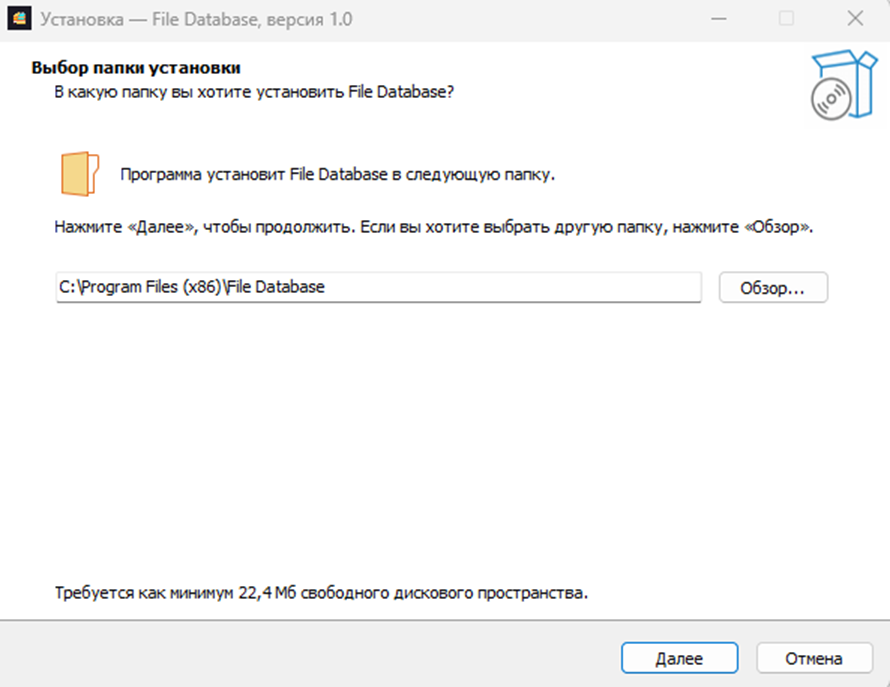


Рисунок 6.1.2 – Указания пути

***6.2 Основные операции***

***6.2.1 Первый запуск программы***

После запуска программы, у вас появляется окно Авторизации системы.



### *6.2.2 Создание записи в базе данных*

Выбрав нужную таблицу, мы попадаем на окно со списком записей этой таблицы. Слева также появляется кнопки для работы с данными, а именно: добавить, изменить и удалить. Пример можно увидеть на рисунке 6.2.2.1. Чтобы создать новую запись, нужно нажать кнопку "Добавить".

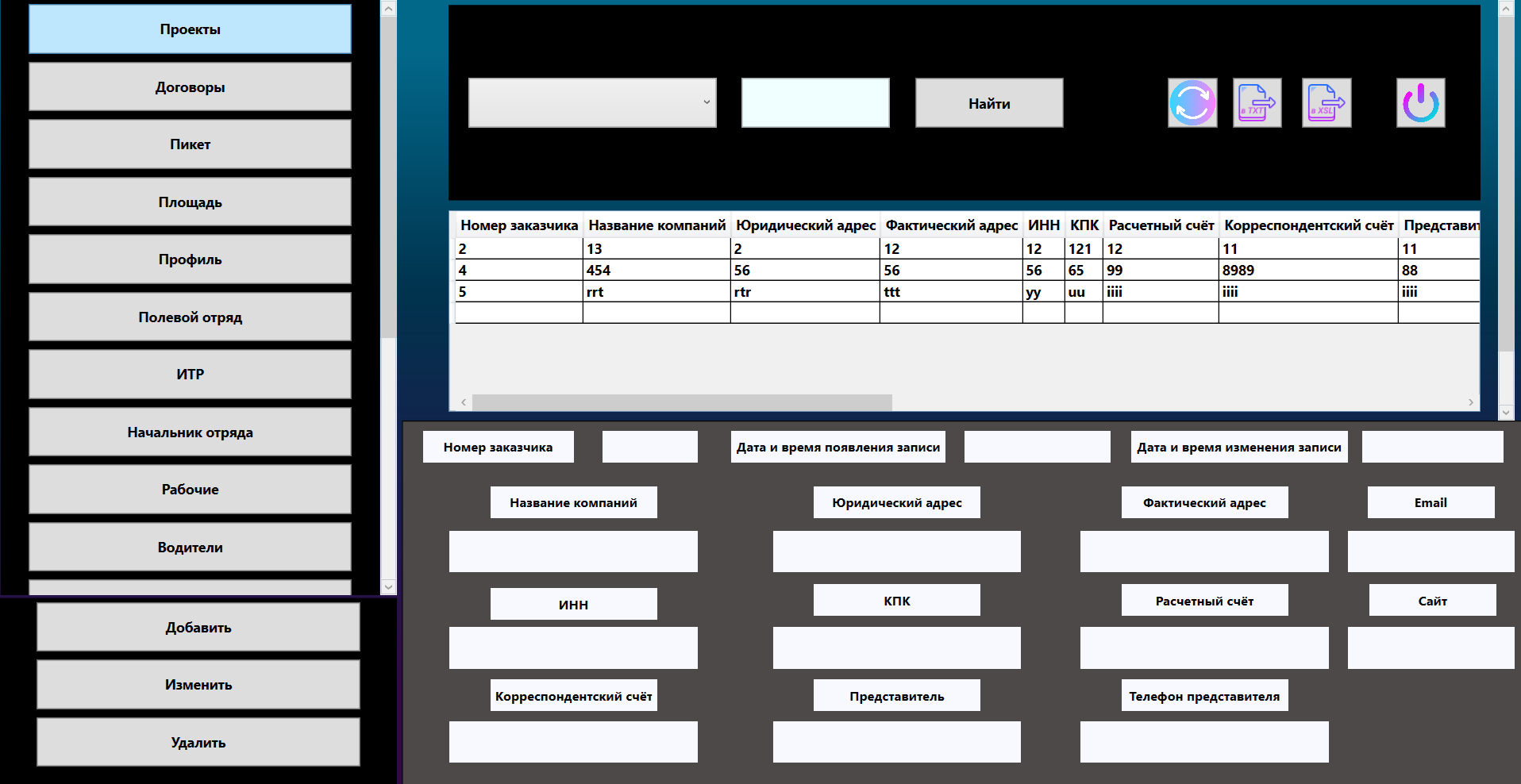


Рисунок 6.2.2.1 – Окно заказчиков

После чего мы заполняем все поля, которые вышли. Пример - у заказчика надо заполнить 11 полей (рисунок 6.2.2.2). После заполнения всех полей, мы сохраняем запись.

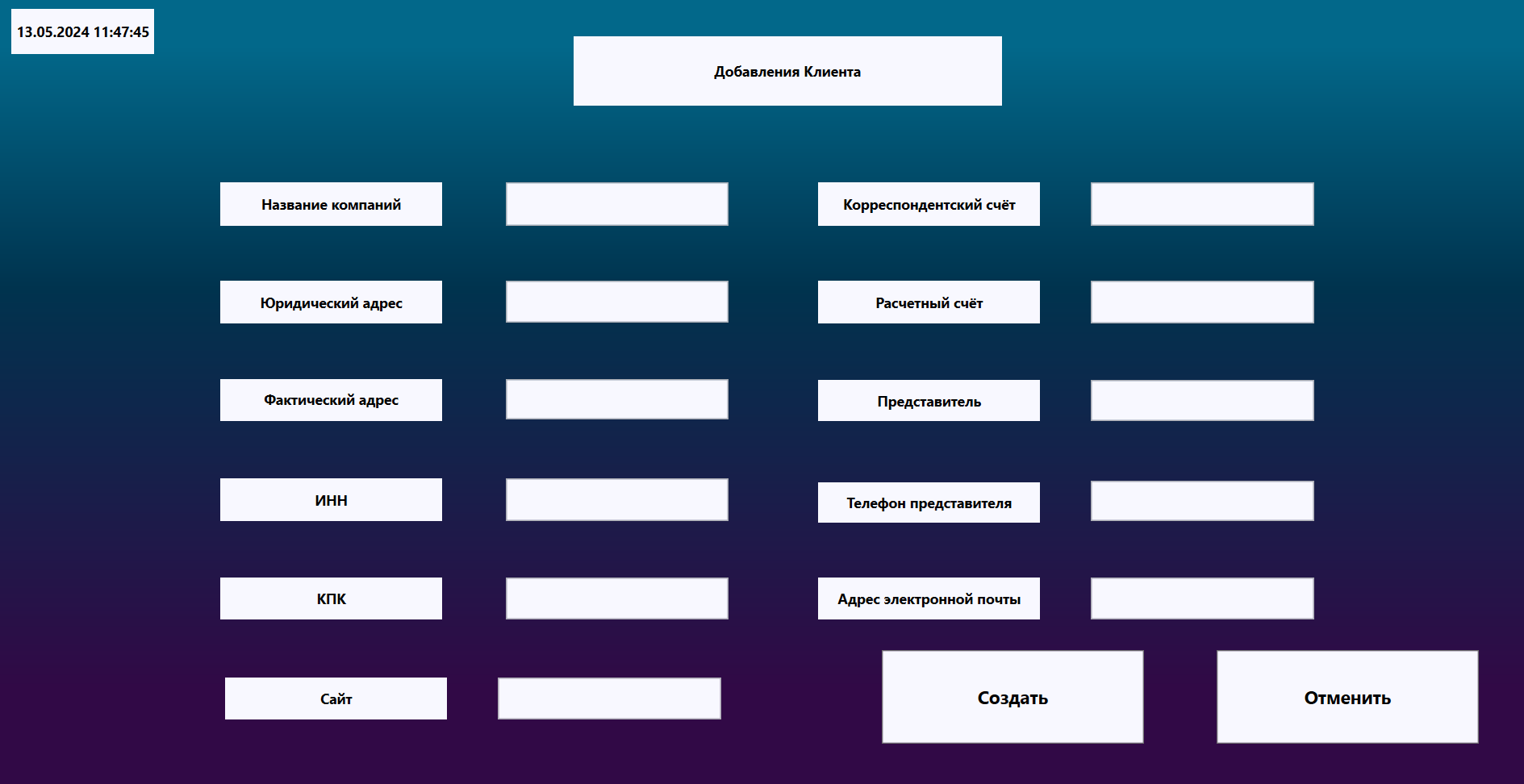


Рисунок 6.2.2.2 – Окно создания заказчика

### *6.2.3 Изменение записи в базе данных*

После выбора нужной таблицы, мы попадаем на окно с таблицей записей этой сущности. Для примера - таблица площади (рисунок 6.2.3.1). Для изменения записи нужно нажать на кнопку “Изменить”.

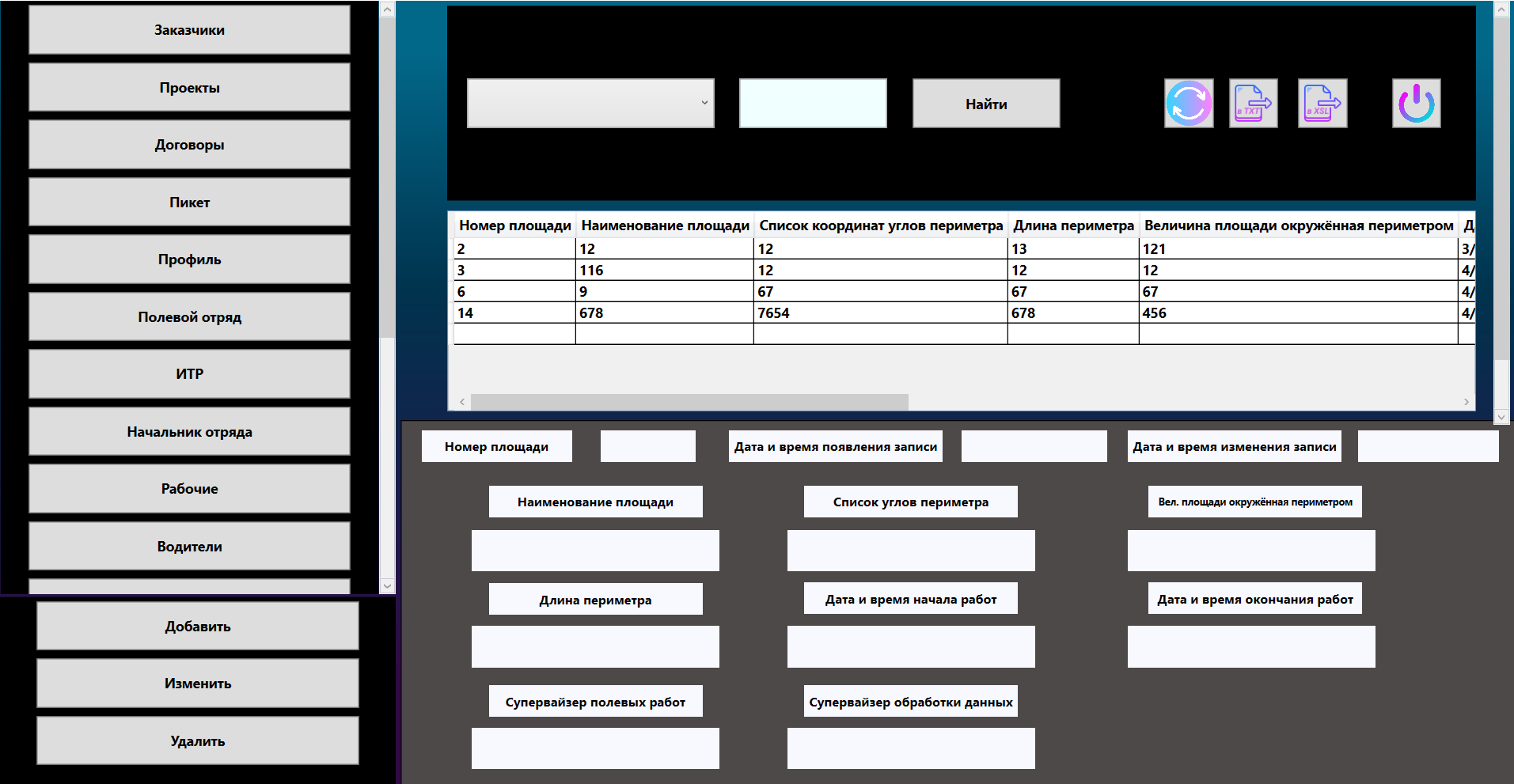


Рисунок 6.2.3.1 – Окно площадей

Открывается окно для изменения данных (рисунок 6.2.3.2). Выбираем нужную запись по номеру. Затем мы изменяем нужные поля. Однако мы не можем изменить первичный ключ (в данном случае номер), так как это запрещено. После внесения изменений мы сохраняем запись.

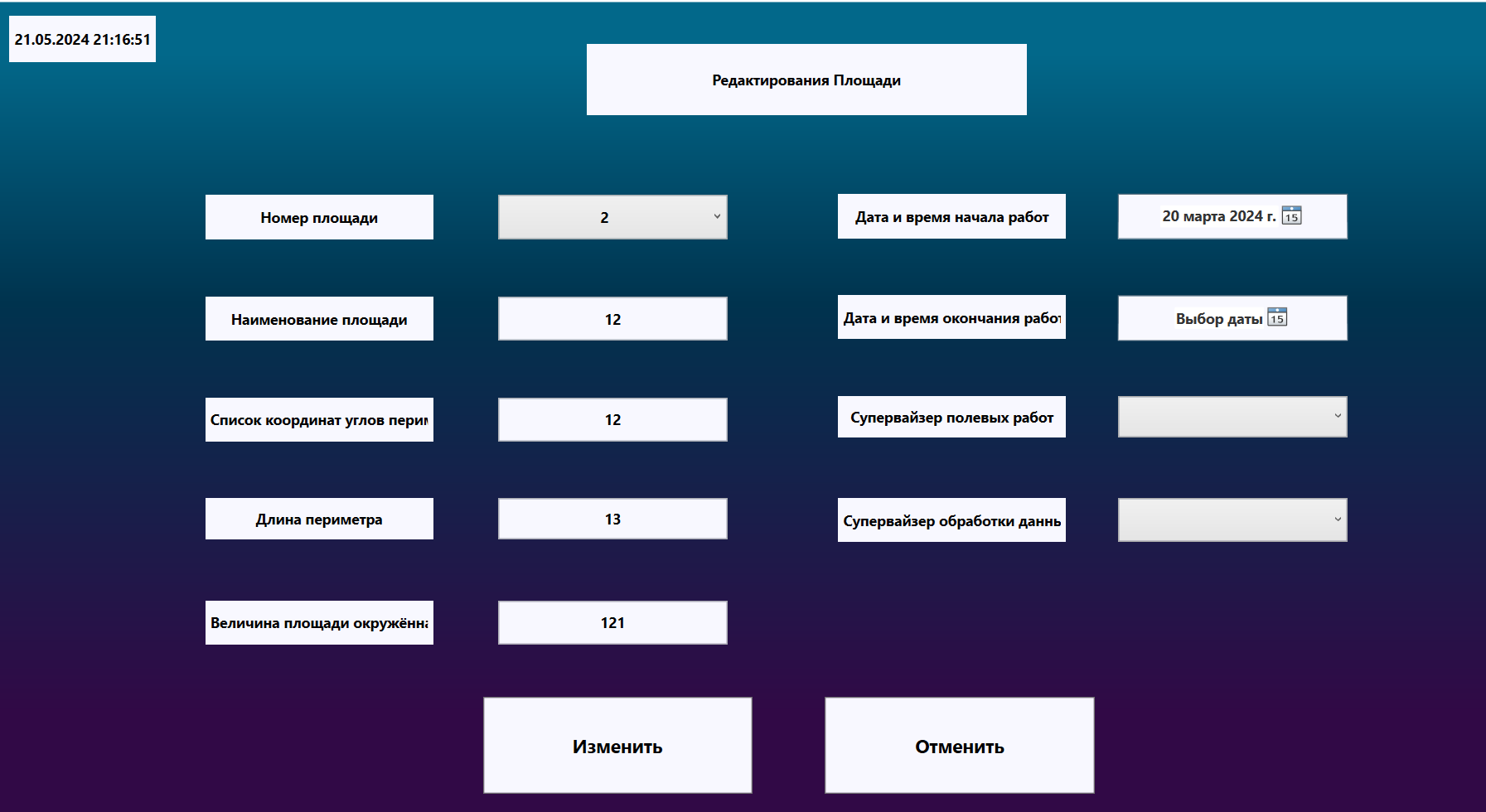


Рисунок 6.2.3.2 – Окно изменения площади

### *6.2.4 Удаление записи в базе данных*

После выбора нужной таблицы, мы попадаем на окно с таблицей записей этой сущности. Для примера - таблица водители (рисунок 6.2.4.1). Для удаления записи нужно поисковике найти необходимую запись, и нажать на кнопку “Удалить”. После чего запись в базе данных удалиться.

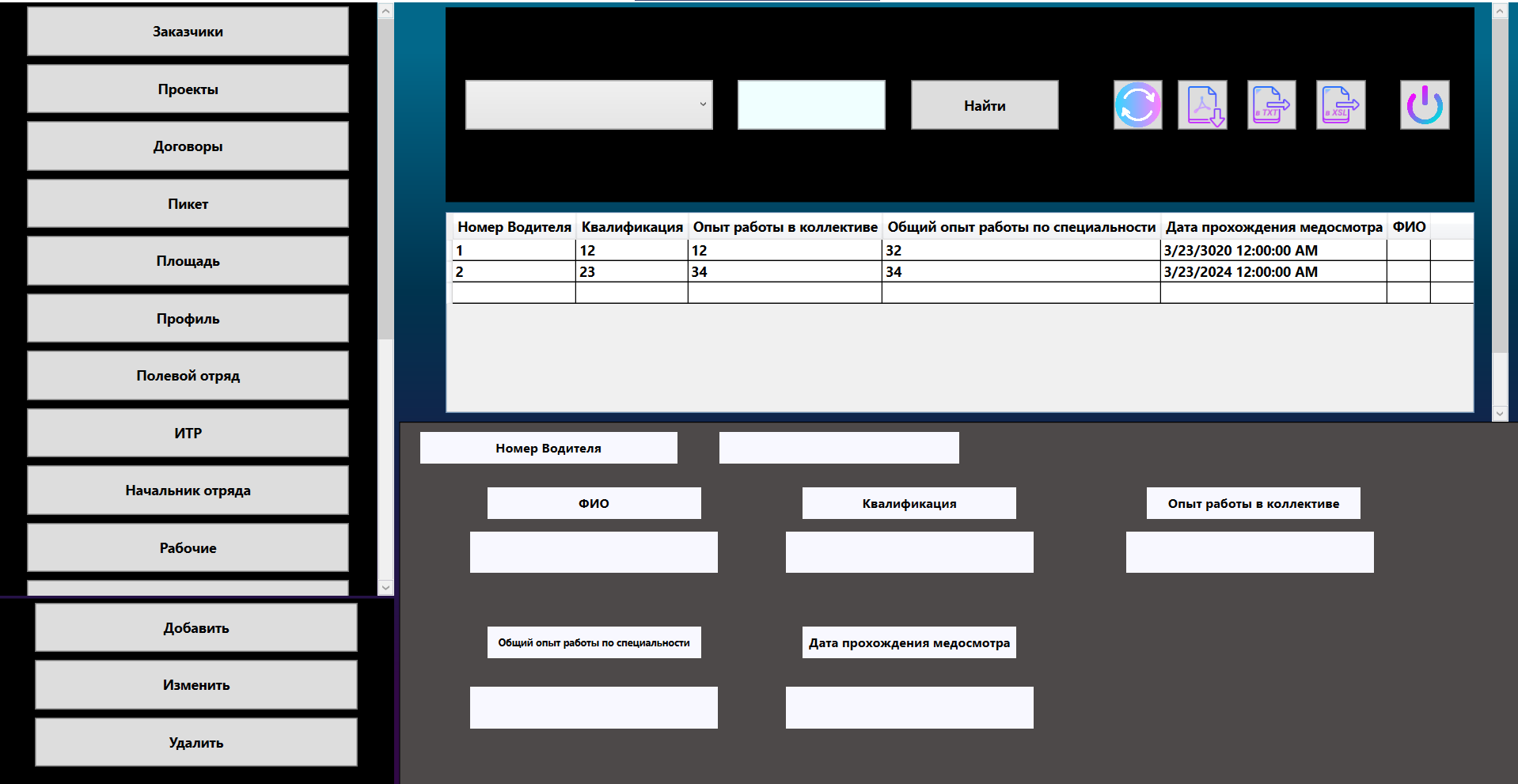


Рисунок 6.2.4.1 – Окно водителей

# *ЗАКЛЮЧЕНИЕ*

В ходе выполнения дипломной работы была достигнута цель по разработке программного средства для эффективного хранения и управления файловыми данными, полученными методом электромагнитных зондирований становлением поля в ближней зоне (ЗСБ).

Для достижения поставленной цели были успешно решены следующие задачи:

* Изучена предметная область метода измерений ЗСБ, что позволило определить требования к структуре и содержанию хранимых данных.
* Изучена предметная область реляционной базы данных, что позволило выбрать оптимальную технологию для реализации поставленных задач.
* Разработана реляционная база данных, соответствующая требованиям по хранению и управлению файловыми данными ЗСБ.
* Создан пользовательский интерфейс, обеспечивающий удобную и эффективную работу с базой данных.

Разработанное программное средство успешно прошло тестирование и показало высокую производительность и надежность. Таким образом, разработанное программное средство является эффективным и надежным инструментом для работы с синтетическими данными, полученными методом ЗСБ, и может быть успешно использовано в научных исследованиях и практических применениях.

# *СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ*

* “Что такое реляционная база данных” URL: <https://www.oracle.com/cis/database/what-is-a-relational-database/>
* “Метод ЗБС” URL: <https://azimut-geology.kz/glossary/geophysics/33-metod-zsb-zondirovanie-stanovleniem-polya-v-blizhney-zone.html>
* Электроразведка: учебное пособие /авт. - сост.: А.А.Иванов, К.В.Новиков, П.В.Новиков - М.: МГРИ, 2019. - 80 стр.
* Чейв Алан В. Морское ЭМЗ с контр ист поля эволюция и технологии
* Компаниец С.В., Кожевников Н.О., Мурзина Е.В., Емельянов В.С. Интерпретация данных зондирования становлением поля в ближней зоне с учетом индукционно-вызванной поляризации при площадных нефтегазопоисковых исследованиях на юге Сибирской платформы. Известия Сибирского отделения секции наук о Земле Российской академии естественных наук. Геология, разведка и разработка месторождений полезных ископаемых. 2019.

# *ПРИЛОЖЕНИЯ*

## *Приложение А*

**Техническое задание для разработки Реляционной Базы Данных (РБД) и интерфейса работы с ней для хранения и доступа к записям зондированием становления поля в ближней зоне (ЗСБ) по заказу Лаборатории Геоэлектрики института нефтегазовой геологии и геофизики (ИНГГ СО РАН)**

**Изучить предметную область геофизических электроразведочных измерений зондированием становления поля в ближней зоне (ЗСБ), проведении полевых измерений, обработке записей, визуального представления измерений и обработки**

**Спроектировать РБД.**

**Разработать РБД .**

В РБД записываются и хранятся сведения о Заказчике работ, Проекте, Площадях проекта, Профилях площадей, Точках записи (Пунктах измерений) профилей, Аппаратуре, Составе отряда, Операторах, Супервайзерах сопровождающих полевые измерения и обработку данных. Условиях измерений (погодные, уровень шумов, условия мешающие проведению измерений). В РБД записываются и хранятся сведения об обработке измерений, графе обработки (использованных фильтрах), составе обработчиков.

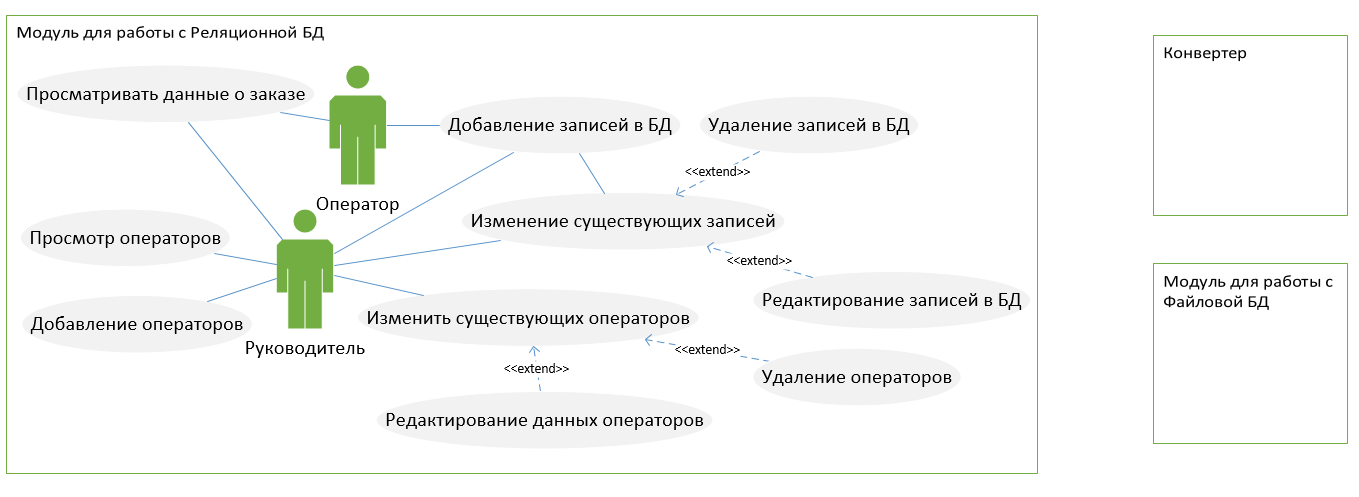
**Разработать интерфейс для работы с РБД.**

* Интерфейс для создания проекта, в котором указывается Заказчик, Представители заказчика, Контакты, Количество площадей, Названия площадей. Редактирования, Добавления и Удаления этих записей. Поддерживается конвертирование этих сведений из списка (из текстового файла и электронной таблицы).
* Интерфейс для регистрации площади, в котором указываются координаты углов периметра, ограничивающего площадь, и отображается контур площади. Редактирования, Добавления и Удаления площади. Поддерживается конвертирование списка координат из файла (из текстового файла и электронной таблицы). Координаты задаются в географической или прямоугольной системе координат.
* Интерфейс для регистрации профиля на площади, в котором указываются проектные координаты начала и окончания линий съёмки и точек изломов и отображается линия профиля на площади. Редактирования, Добавления и Удаления профиля. Поддерживается конвертирование списка проектных координат из файла (из текстового файла и электронной таблицы). Координаты задаются в географической или прямоугольной системе координат.
* Интерфейс для регистрации точки записи на профиле и её отображения на линии профиля и на площади. Редактирования, Добавления и Удаления точки измерения. Поддерживается несколько видов точек измерения: рядовое, контрольное, опытно-методическое. Поддерживается конвертирование списка проектных координат из файла (из текстового файла и электронной таблицы). Координаты задаются в географической или прямоугольной системе координат.
* Интерфейс для регистрации измерительной аппаратуры, измерительных датчиков, генераторной аппаратуры, модулей связи.
* Интерфейс для регистрации операторов и состава полевого отряда. Поддерживается конвертирование этих сведений из списка (из текстового файла и электронной таблицы)
* Интерфейс для регистрации супервайзера.
* Интерфейс для регистрации обработчиков измерений.
* Интерфейс для визуализации измеренных кривых в виде графиков ЭДС и кажущегося сопротивления по профилям и по площади.
* Интерфейс для представления результатов измерений в виде изолиний ЭДС и кажущегося сопротивления по профилям и по площади (на отдельных временах измерений).
* Интерфейс визуализации рядовых и контрольных, рядовых и опытно-методических измерений в виде графиков ЭДС и кажущегося сопротивления по профилям и по площади. Расчёт расхождений рядовых и контрольных, рядовых и опытно-методических измерений. Вывод и экспортирование отчёта о расхождениях.

## *Приложение Б*

{Содержит примеры входных данных в виде изображений, таблиц, фрагментов текстовых документов и.т.д.}

## *Приложение В*

Рисунок 2.3.1.1 − Функциональные возможности программного средства

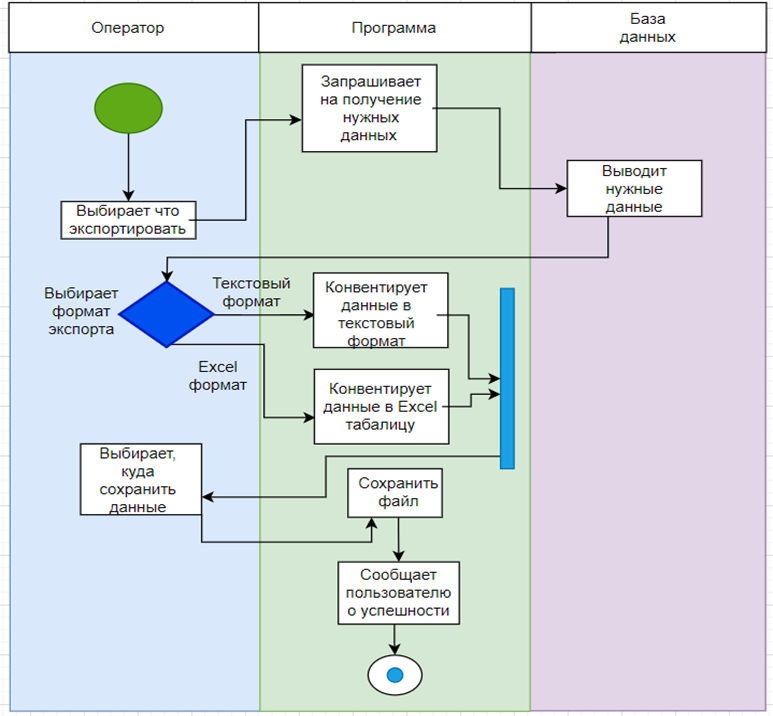


Рисунок 4.2.1.1 − Диаграмму активности для экспорта геофизических данных

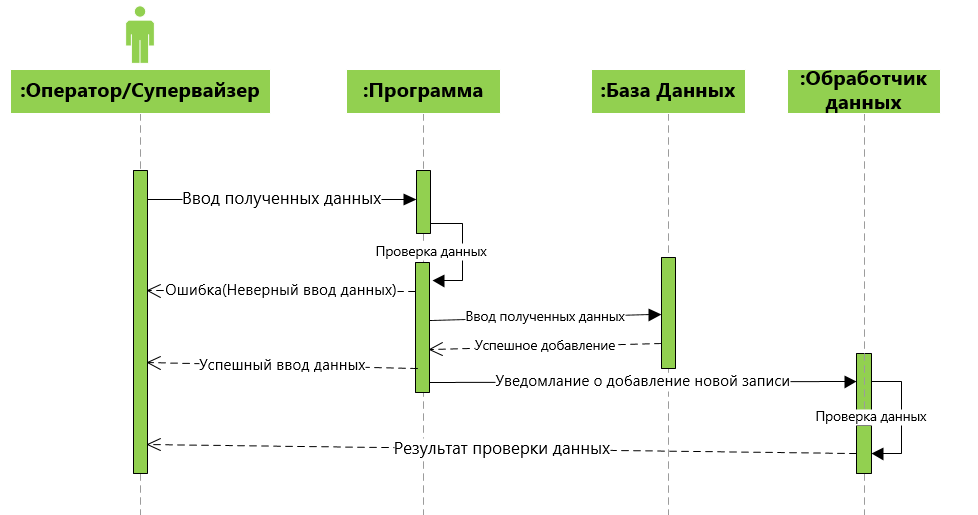
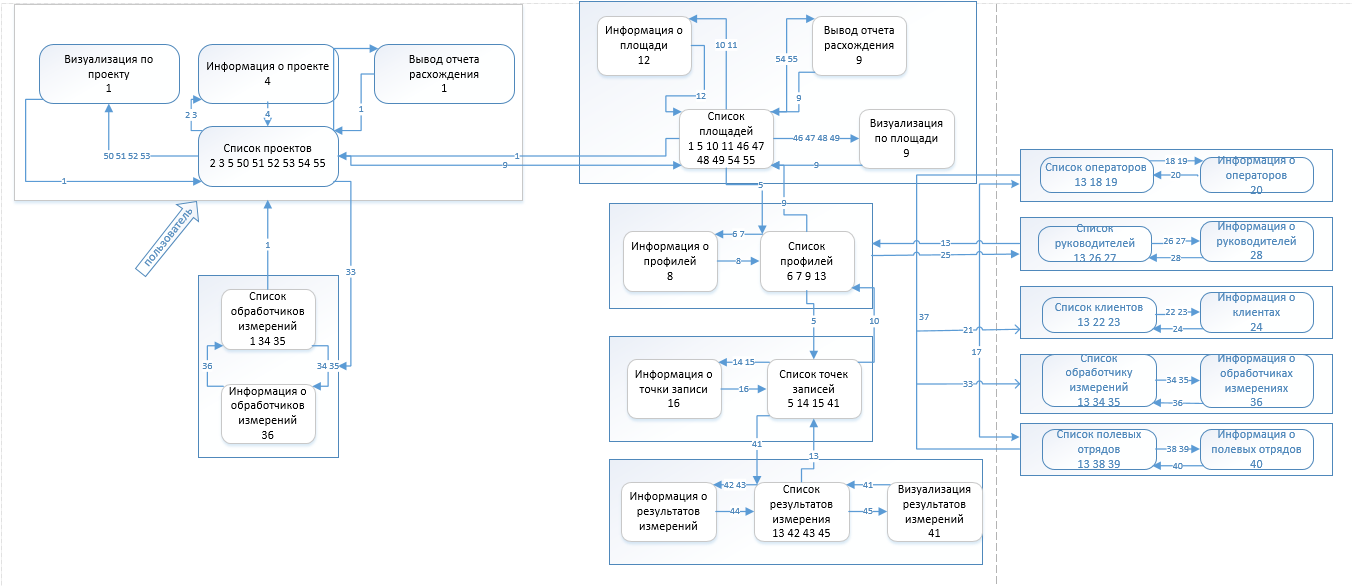


Рисунок 4.2.2.1 − Диаграмма последовательности для сценария ввода данных в базу данных

Рисунок 4.2.5.1. − Структура экранов программы

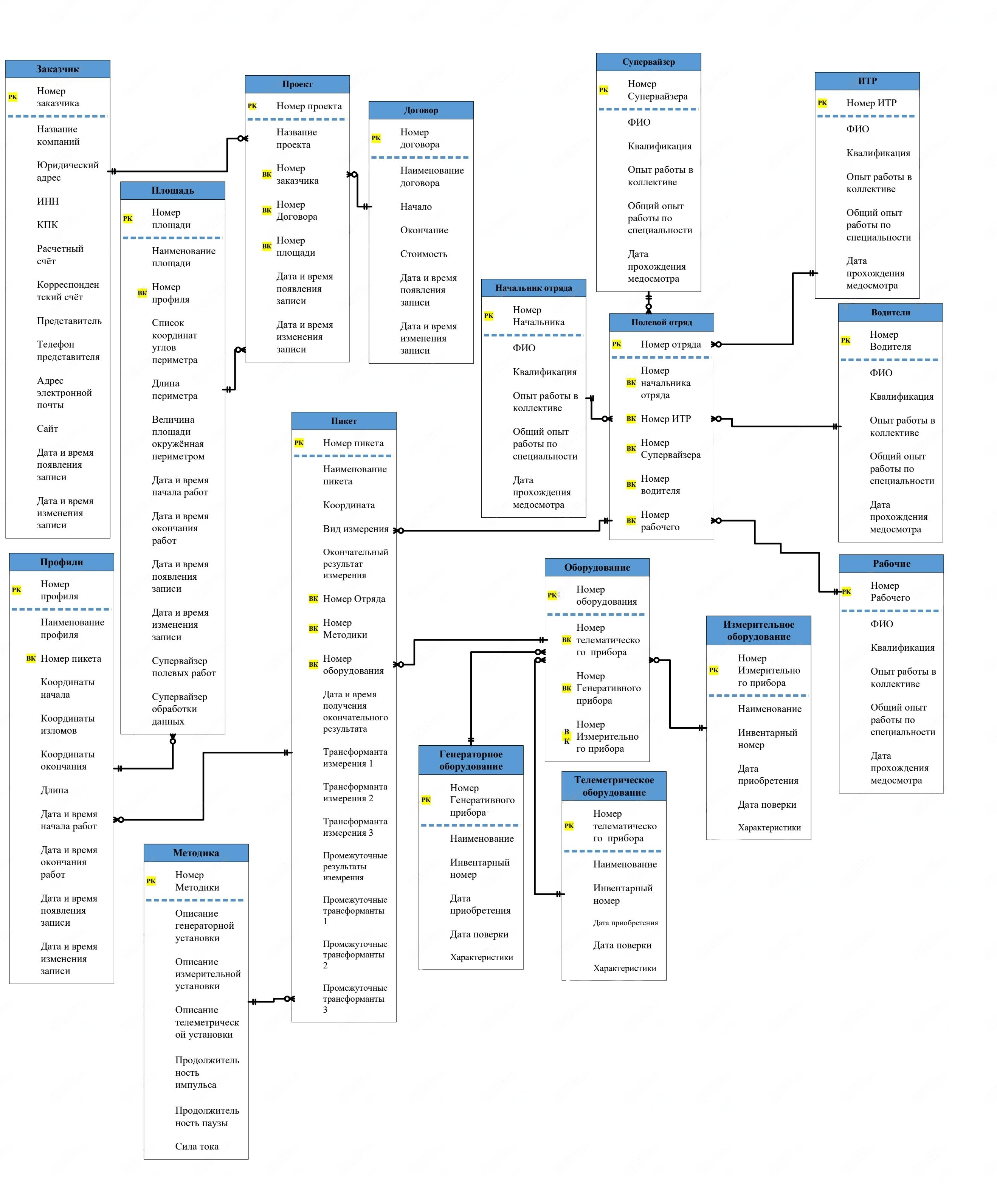


Рисунок 4.4.1 - Схема базы данных

## *Приложение Г*

{Содержит результаты тестовых испытаний на различных тестовых данных.

*Например, при наличии в составе расчетно-графического модуля, полученные в результате работы, иллюстрации и таблицы, по которым можно судить о корректной работе программы*.}

## *Приложение Д*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace WindowRBD1.Class

{

internal class BdCon

{

public static string Con = "Data Source=ANDREEVDAA\\SQLEXPRESS;Initial Catalog=Control;Integrated Security=True";

}

}

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.Forms

{

public partial class ForgotPassword : Window

{

public ForgotPassword()

{

InitializeComponent();

Role1();

}

SqlConnection con = new SqlConnection(BdCon.Con);

private void Role1() // Получение роли пользователя

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Роль] FROM Proekt.Пользователи";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

cmbRole.Items.Clear();

while (reader.Read())

{

cmbRole.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void BtnClose\_Click(object sender, RoutedEventArgs e) // закрытие формы

{

this.Close();

}

private void BtnProv\_Click(object sender, RoutedEventArgs e) // Проверка пользователя в базе данных

{

con.Open();

SqlCommand cmd = new SqlCommand("select Пароль from Proekt.Пользователи where Логин = @login and Роль = @Role", con);

cmd.Parameters.AddWithValue("@login", txtLogin.Text);

cmd.Parameters.AddWithValue("@Role", cmbRole.SelectedItem);

SqlDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

txtPassword.Text = dr[0].ToString();

}

else

{

MessageBox.Show("username not available");

txtPassword.Text = "";

}

con.Close();

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Net.Mail;

using System.Security.AccessControl;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.Forms

{

public partial class FormLoading : Window

{

public FormLoading(string name, string FIO)

{

InitializeComponent();

lineProgress();

label1.Content += name;

fio = FIO;

}

public static void checkTime() //секундомер

{

string time = string.Format("{0:mm:ss}", DateTime.Now);

int minute = Int32.Parse(string.Format("{0:mm}", DateTime.Now));

string sec = string.Format("{0:ss}", DateTime.Now);

if (minute <= 20)

minute++;

else

minute = 01;

string timer = minute + ":" + sec;

if (minute < 10)

timer = "0" + minute + ":" + sec;

while (true)

{

time = string.Format("{0:mm:ss}", DateTime.Now);

if (timer.Equals(time))

Environment.Exit(0);

}

}

string fio;

SqlConnection con = new SqlConnection(BdCon.Con);

private async void lineProgress() // Запуск ProgressBar

{

for (int i = 0; i <= 100; i++)

{

pgbEstatus.Value = i \* (100 / 100);

await Task.Delay(50);

label2.Content = i + "%";

}

con.Open();

SqlCommand cmd = new SqlCommand("select \* from Proekt.Пользователи where ФИО = @Fio", con); // Определение роли пользователя для того чтобы дать доступ к базе данных

cmd.Parameters.AddWithValue("@Fio", fio);

SqlDataAdapter sda = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

sda.Fill(dt);

if (dt.Rows.Count > 0)

{

string usertype = dt.Rows[0][4].ToString();

if (usertype == "Оператор")

{

//label1.Margin = new Thickness(580, 470, 0, 0);

this.Hide();

MainWindowOperator red = new MainWindowOperator();

red.ShowDialog();

Thread thread = new Thread(checkTime);

thread.Start();

}

if (usertype == "Супервайзер")

{

this.Hide();

MainWindow red = new MainWindow();

red.ShowDialog();

Thread thread = new Thread(checkTime);

thread.Start();

label1.Margin = new Thickness(500, 470, 0, 0);

}

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Reflection.Emit;

using System.Security.AccessControl;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.FormsEdit;

using static System.Net.Mime.MediaTypeNames;

namespace WindowRBD1.Forms

{

public partial class FormReg : Window

{

public FormReg()

{

InitializeComponent();

res.Source = new System.Windows.Media.ImageSourceConverter().ConvertFromString("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\Image\\iconsClosePassword.png") as System.Windows.Media.ImageSource;

Display.AddHandler(System.Windows.Controls.Button.PreviewMouseDownEvent, new MouseButtonEventHandler(Btn\_PreviewMouseDown));

Display.AddHandler(System.Windows.Controls.Button.PreviewMouseUpEvent, new MouseButtonEventHandler(Btn\_PreviewMouseUp));

}

SqlConnection con = new SqlConnection(BdCon.Con);

public string FIO;

private void BtnEntry\_Click(object sender, RoutedEventArgs e) //Происходит проверка логина и пароля введенные пользователем в бд

{

if (txtLog.Text != "" && pwdPasswordBox.Password != "")

{

con.Open();

SqlCommand cmd = new SqlCommand("select \* from Proekt.Пользователи where Логин = @login and Пароль = @password", con);

cmd.Parameters.AddWithValue("@login", txtLog.Text);

cmd.Parameters.AddWithValue("@password", pwdPasswordBox.Password);

SqlDataAdapter sda = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

sda.Fill(dt);

if (dt.Rows.Count > 0)

{

string usertype = dt.Rows[0][4].ToString();

if (usertype == "Оператор")

{

SqlCommand cmd1 = new SqlCommand("select ФИО from Proekt.Пользователи where Логин = @login", con);

cmd1.Parameters.AddWithValue("@login", txtLog.Text);

FIO = (string)cmd1.ExecuteScalar();

System.Windows.MessageBox.Show(FIO + ",добро пожаловать в меню Оператора!", "Успешно!", (MessageBoxButton)MessageBoxButtons.OK, (MessageBoxImage)MessageBoxIcon.Information);

new FormLoading(FIO + ",добро пожаловать в меню Оператора!",FIO).ShowDialog();

this.Hide();

}

if (usertype == "Супервайзер")

{

SqlCommand cmd1 = new SqlCommand("select ФИО from Proekt.Пользователи where Логин = @login", con);

cmd1.Parameters.AddWithValue("@login", txtLog.Text);

FIO = (string)cmd1.ExecuteScalar();

System.Windows.MessageBox.Show(FIO + ",добро пожаловать в меню Супервайзера!", "Успешно!", (MessageBoxButton)MessageBoxButtons.OK, (MessageBoxImage)MessageBoxIcon.Information);

new FormLoading(FIO + ",добро пожаловать в меню Супервайзера!", FIO).ShowDialog();

this.Hide();

}

}

else

{

System.Windows.MessageBox.Show("Такого пользователя нету!", "Успешно!", (MessageBoxButton)MessageBoxButtons.OK, (MessageBoxImage)MessageBoxIcon.Information);

}

con.Close();

}

}

private void BtnPassword\_Click(object sender, RoutedEventArgs e) // открытие формы

{

ForgotPassword td = new ForgotPassword();

td.ShowDialog();

}

private void Btn\_PreviewMouseDown(object sender, MouseButtonEventArgs e) //меняет местами Password и TextBox

{

res.Source = new System.Windows.Media.ImageSourceConverter().ConvertFromString("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\Image\\iconsDisplayPassword.png") as System.Windows.Media.ImageSource;

pwdTextBox.Text = pwdPasswordBox.Password; // скопируем в TextBox из PasswordBox

pwdTextBox.Visibility = Visibility.Visible; // TextBox - отобразить

pwdPasswordBox.Visibility = Visibility.Hidden; // PasswordBox - скрыть

}

private void Btn\_PreviewMouseUp(object sender, MouseButtonEventArgs e) //меняет местами TextBox и Password

{

res.Source = new System.Windows.Media.ImageSourceConverter().ConvertFromString("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\Image\\iconsClosePassword.png") as System.Windows.Media.ImageSource;

pwdPasswordBox.Password = pwdTextBox.Text; // скопируем в PasswordBox из TextBox

pwdTextBox.Visibility = Visibility.Hidden; // TextBox - скрыть

pwdPasswordBox.Visibility = Visibility.Visible; // PasswordBox - отобразить

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

using WindowRBD1.FormsMain;

namespace WindowRBD1.Forms

{

public partial class MainWindow : System.Windows.Window

{

public MainWindow()

{

InitializeComponent();

}

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer form1 = new Cursomer();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project form1 = new Project();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract form1 = new Contract();

form1.Show();

}

private void OrderCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Order form1 = new Order();

form1.Show();

}

private void EngineerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Engineer form1 = new Engineer();

form1.Show();

}

private void ChiefCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Chief form1 = new Chief();

form1.Show();

}

private void WorkerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Worker form1 = new Worker();

form1.Show();

}

private void DriverCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Driver form1 = new Driver();

form1.Show();

}

private void SupervisorCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Supervisor form1 = new Supervisor();

form1.Show();

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring form1 = new Measuring();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative form1 = new Generative();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry form1 = new Telemetry();

form1.Show();

}

private void MethodologyCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Methodology form1 = new Methodology();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket form1 = new Picket();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area form1 = new Area();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile form1 = new Profile();

form1.Show();

}

private void EquipmentCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Equipment form1 = new Equipment();

form1.Show();

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие формы

{

this.Close();

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e)

{

dataGridView1.IsReadOnly = true;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

using WindowRBD1.FormsMain;

using WindowRBD1.FormsMainOperator;

namespace WindowRBD1.Forms

{

public partial class MainWindowOperator : System.Windows.Window

{

public MainWindowOperator()

{

InitializeComponent();

}

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer1 form1 = new Cursomer1();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project1 form1 = new Project1();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract1 form1 = new Contract1();

form1.Show();

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring1 form1 = new Measuring1();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative1 form1 = new Generative1();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry1 form1 = new Telemetry1();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket1 form1 = new Picket1();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area1 form1 = new Area1();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile1 form1 = new Profile1();

form1.Show();

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие формы

{

this.Close();

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e)

{

dataGridView1.IsReadOnly = true;

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.Forms

{

public partial class NumberADDGen : Window

{

public NumberADDGen()

{

InitializeComponent();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void btnClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Close();

}

private void btnCreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "insert into Proekt.[Описание генераторной установки]([Размер])" +

$"values ('{Size}')";

SqlCommand comm = new SqlCommand(sql, conn);

comm.ExecuteNonQuery();

MessageBox.Show("Запись добавлена");

conn.Close();

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.Forms

{

public partial class NumberADDMeauring : Window

{

public NumberADDMeauring()

{

InitializeComponent();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Close();

}

private void btCreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "insert into Proekt.[Описание измерительной установки]([Размер],[Разносы])" +

$"values ('{Size}','{Postings}')";

SqlCommand comm = new SqlCommand(sql, conn);

comm.ExecuteNonQuery();

MessageBox.Show("Запись добавлена");

conn.Close();

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.Forms

{

public partial class NumberArea : Window

{

public NumberArea()

{

InitializeComponent();

dataGridView();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

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

{

this.Close();

}

private void dataGridView() //Происходит заполнение данными datagrid

{

string sql = "select \* from Proekt.[Площади]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.[Площади]");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.[Площади]"].DefaultView;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.Forms

{

public partial class NumberEq : Window

{

public NumberEq()

{

InitializeComponent();

dataGridView();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

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

{

this.Close();

}

private void dataGridView() //Происходит заполнение данными datagrid

{

string sql = "select \* from Proekt.Оборудования";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Оборудования");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Оборудования"].DefaultView;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.Forms

{

public partial class NumberGenerative : Window

{

public NumberGenerative()

{

InitializeComponent();

dataGridView();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

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

{

this.Close();

}

private void dataGridView() //Происходит заполнение данными datagrid

{

string sql = "select \* from Proekt.[Описание генераторной установки]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.[Описание генераторной установки]");

dataGridView1.ItemsSource = ds.Tables["Proekt.[Описание генераторной установки]"].DefaultView;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.Forms

{

public partial class NumberMeasuring : Window

{

public NumberMeasuring()

{

InitializeComponent();

dataGridView();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

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

{

this.Close();

}

private void dataGridView() //Происходит заполнение данными datagrid

{

string sql = "select \* from Proekt.[Описание измерительной установки]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.[Описание измерительной установки]");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.[Описание измерительной установки]"].DefaultView;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

namespace WindowRBD1.Forms

{

public partial class NumberMet : Window

{

public NumberMet()

{

InitializeComponent();

dataGridView();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

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

{

this.Close();

}

private void dataGridView() //Происходит заполнение данными datagrid

{

string sql = "select \* from Proekt.Методика";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Методика");

dataGridView1.ItemsSource = ds.Tables["Proekt.Методика"].DefaultView;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.Forms

{

public partial class NumberMethodology: Window

{

public NumberMethodology()

{

InitializeComponent();

dataGridView();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

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

{

this.Close();

}

private void dataGridView() //Происходит заполнение данными datagrid

{

string sql = "select \* from Proekt.Методика";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.[Методика]");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.[Методика]"].DefaultView;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.Forms

{

public partial class NumberOrder : Window

{

public NumberOrder()

{

InitializeComponent();

dataGridView();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

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

{

this.Close();

}

private void dataGridView() //Происходит заполнение данными datagrid

{

string sql = "select \* from Proekt.[Полевой отряд]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.[Полевой отряд]");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.[Полевой отряд]"].DefaultView;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.Forms

{

public partial class NumberPicket : Window

{

public NumberPicket()

{

InitializeComponent();

dataGridView();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

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

{

this.Close();

}

private void dataGridView() //Происходит заполнение данными datagrid

{

string sql = "select \* from Proekt.Пикет";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Пикет");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Пикет"].DefaultView;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.Forms

{

public partial class NumberProfile : Window

{

public NumberProfile()

{

InitializeComponent();

dataGridView();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

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

{

this.Close();

}

private void dataGridView() //Происходит заполнение данными datagrid

{

string sql = "select \* from Proekt.Профили";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Профили");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Профили"].DefaultView;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

namespace WindowRBD1.FormsCreate

{

public partial class CreateArea : Window

{

public CreateArea()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

txtSupervisorOrder1();

txtSupervisorData1();

txtProfile1();

txtList1();

}

private void BtClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void btCreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "insert into Proekt.Площади([Наименование площади],[Номер профиля],[Номер списка углов периметра],[Длина периметра],[Величина площади окружённая периметром],[Дата и время появления записи],[Супервайзер полевых работ],[Супервайзер обработки данных])" +

$"values ('{txtArea.Text}','{txtProfile.SelectedItem}','{txtList.SelectedItem}','{txtPerimeterLength.Text}', '{txtSizeArea.Text}','{Time.Content}','{txtSupervisorOrder.SelectedItem}','{txtSupervisorData.SelectedItem}')";

SqlCommand comm = new SqlCommand(sql, conn);

comm.ExecuteNonQuery();

MessageBox.Show("Запись добавлена");

}

}

private void txtSupervisorOrder1() //Собирает [Номера Супервайзеров] которые существуют в сущности Proekt.Супервайзер на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Супервайзера] FROM Proekt.Супервайзер";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtSupervisorOrder.Items.Clear();

while (reader.Read())

{

txtSupervisorOrder.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void txtList1() //Собирает [Номер списка углов периметра] которые существуют в сущности Proekt.[Список координат углов периметра] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер списка углов периметра] FROM Proekt.[Список координат углов периметра]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtList.Items.Clear();

while (reader.Read())

{

txtList.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void txtProfile1() //Собирает [Номер профиля] которые существуют в сущности Proekt.Профили на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер профиля] FROM Proekt.Профили";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtProfile.Items.Clear();

while (reader.Read())

{

txtProfile.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void txtSupervisorData1() //Собирает [Номера Супервайзеров] которые существуют в сущности Proekt.Супервайзер на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Супервайзера] FROM Proekt.Супервайзер";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtSupervisorData.Items.Clear();

while (reader.Read())

{

txtSupervisorData.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btPoisk\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

NumberProfile tv = new NumberProfile();

tv.ShowDialog();

}

private void btAdd\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

CreateProfile tv = new CreateProfile();

tv.ShowDialog();

}

}

}

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.FormsCreate

{

public partial class CreateContract : Window

{

public CreateContract()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

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

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "insert into Proekt.Договор([Наименование договора],[Начала],[Окончания],[Стоимость],[Дата и время появления записи])" +

$"values ('{txtNameContract.Text}','{txtBeginnings.Text}','{txtEndings.Text}', '{txtCost.Text}', '{Time.Content}')";

SqlCommand comm = new SqlCommand(sql, conn);

comm.ExecuteNonQuery();

MessageBox.Show("Запись добавлена");

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate.Equipment;

namespace WindowRBD1.FormsCreate

{

public partial class CreateEquipment : Window

{

public CreateEquipment()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

Generative1();

Measuring1();

Telemetry1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void Generative1() //Собирает [Номер генеративного оборудования] которые существуют в сущности Proekt.[Генеративное оборудование] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер генеративного оборудования] FROM Proekt.[Генеративное оборудование]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

cmbGenerative.Items.Clear();

while (reader.Read())

{

cmbGenerative.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void Measuring1() //Собирает [Номер измерительного оборудования] которые существуют в сущности Proekt.[Измерительное оборудование] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер измерительного оборудования] FROM Proekt.[Измерительное оборудование]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

cmbMeasuring.Items.Clear();

while (reader.Read())

{

cmbMeasuring.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void Telemetry1() //Собирает [Номер телеметрического оборудования] которые существуют в сущности Proekt.[Телеметрическое оборудование] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер телеметрического оборудования] FROM Proekt.[Телеметрическое оборудование]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

cmbTelemetry.Items.Clear();

while (reader.Read())

{

cmbTelemetry.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btClose\_Click(object sender, EventArgs e) //закрытие формы

{

this.Close();

}

private void btCreate\_Click(object sender, EventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "insert into Proekt.Оборудования([Номер генеративного оборудования],[Номер измерительного оборудования],[Номер телеметрического оборудования])" +

$"values ('{cmbGenerative.SelectedItem}','{cmbMeasuring.SelectedItem}','{cmbTelemetry.SelectedItem}')";

SqlCommand comm = new SqlCommand(sql, conn);

comm.ExecuteNonQuery();

MessageBox.Show("Запись добавлена");

}

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select [Номер генеративного оборудования],[Наименование] from Proekt.[Генеративное оборудование]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Водители");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Водители"].DefaultView;

}

}

private void btPoisk2\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select [Номер измерительного оборудования],[Наименование] from Proekt.[Измерительное оборудование]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Рабочие");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Рабочие"].DefaultView;

}

}

private void btPoisk3\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select [Номер телеметрического оборудования],[Наименование] from Proekt.[Телеметрическое оборудование]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Супервайзер");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Супервайзер"].DefaultView;

}

}

private void btPoisk4\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select [Номер Оборудования],[Наименование] from Proekt.Оборудования";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Супервайзер");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Супервайзер"].DefaultView;

}

}

private void btAdd2\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

CreateMeasuring tv = new CreateMeasuring();

tv.ShowDialog();

}

private void btAdd3\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

CreateGenerative tv = new CreateGenerative();

tv.ShowDialog();

}

private void btAdd4\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

CreateTelemetry tv = new CreateTelemetry();

tv.ShowDialog();

}

}

}

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

namespace WindowRBD1.FormsCreate

{

public partial class CreateMethodology : Window

{

public CreateMethodology()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

}

private void BtClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void btCreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "insert into Proekt.Методика([Наименование методики],[Описание генераторной установки],[Описание измерительной установки],[Описание телеметрической установки],[Продолжительность импульса],[Продолжительность паузы],[Сила тока])" +

$"values ('{txtNameMethodology.Text}','{txtGenerative.Text}','{txtMeasuring.Text}', '{txtTelemetry.Text}', '{txtImpulse.Text}','{txtPause.Text}','{txtCurrent.Text}')";

SqlCommand comm = new SqlCommand(sql, conn);

comm.ExecuteNonQuery();

MessageBox.Show("Запись добавлена");

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

NumberADDGen tr = new NumberADDGen();

tr.ShowDialog();

}

private void btAdd2\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

NumberADDMeauring tr = new NumberADDMeauring();

tr.ShowDialog();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

NumberGenerative tv = new NumberGenerative();

tv.ShowDialog();

}

private void btPoisk2\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

NumberMeasuring tv = new NumberMeasuring();

tv.ShowDialog();

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate.Works;

namespace WindowRBD1.FormsCreate

{

public partial class CreateOrder : Window

{

public CreateOrder()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

FioChief1();

FioWorker1();

FioDriver1();

FioSupervisor1();

FioITR1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void btCreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

int FioWorker2 = Convert.ToInt32(FioWorker.SelectedItem);

int FioITR2 = Convert.ToInt32(FioITR.SelectedItem);

int FioSupervisor2 = Convert.ToInt32(FioSupervisor.SelectedItem);

int FioDriver2 = Convert.ToInt32(FioDriver.SelectedItem);

int FioChief2 = Convert.ToInt32(FioChief.SelectedItem);

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "insert into Proekt.[Полевой отряд]([Номер Начальника],[Номер ИТР] ,[Номер Водителя],[Номер Рабочего],[Номер Супервайзера])" +

$"values ('{FioChief2}','{FioITR2}','{FioDriver2}', '{FioWorker2}', '{FioSupervisor2}')";

SqlCommand comm = new SqlCommand(sql, conn);

comm.ExecuteNonQuery();

MessageBox.Show("Запись добавлена");

conn.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void FioChief1() //Собирает [Номер Начальника] которые существуют в сущности Proekt.[Начальник отряда] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Начальника] FROM Proekt.[Начальник отряда]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

FioChief.Items.Clear();

while (reader.Read())

{

FioChief.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void FioDriver1() //Собирает [Номер Водителя] которые существуют в сущности Proekt.[Водители] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Водителя] FROM Proekt.[Водители]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

FioDriver.Items.Clear();

while (reader.Read())

{

FioDriver.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void FioITR1() //Собирает [Номер ИТР] которые существуют в сущности Proekt.[ИТР] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер ИТР] FROM Proekt.[ИТР]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

FioITR.Items.Clear();

while (reader.Read())

{

FioITR.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void FioSupervisor1() //Собирает [Номер Супервайзера] которые существуют в сущности Proekt.[Супервайзер] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Супервайзера] FROM Proekt.[Супервайзер]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

FioSupervisor.Items.Clear();

while (reader.Read())

{

FioSupervisor.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void FioWorker1() //Собирает [Номер Рабочего] которые существуют в сущности Proekt.[Рабочие] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Рабочего] FROM Proekt.[Рабочие]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

FioWorker.Items.Clear();

while (reader.Read())

{

FioWorker.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btPoisk2\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select [Номер Начальника],ФИО from Proekt.[Начальник отряда]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.[Начальник отряда]");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.[Начальник отряда]"].DefaultView;

}

}

private void btPoisk3\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select [Номер ИТР],ФИО from Proekt.ИТР";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.ИТР");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.ИТР"].DefaultView;

}

}

private void btPoisk4\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select [Номер Водителя],ФИО from Proekt.Водители";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Водители");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Водители"].DefaultView;

}

}

private void btPoisk5\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select [Номер Рабочего],ФИО from Proekt.Рабочие";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Рабочие");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Рабочие"].DefaultView;

}

}

private void btPoisk6\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select [Номер Супервайзера],ФИО from Proekt.Супервайзер";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Супервайзер");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Супервайзер"].DefaultView;

}

}

private void btAdd2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateChief tv = new CreateChief();

tv.ShowDialog();

}

private void btAdd3\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateITR tv = new CreateITR();

tv.ShowDialog();

}

private void btAdd4\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateDriver tv = new CreateDriver();

tv.ShowDialog();

}

private void btAdd5\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateWorker tv = new CreateWorker();

tv.ShowDialog();

}

private void btAdd6\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateSupervisor tv = new CreateSupervisor();

tv.ShowDialog();

}

}

}

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

namespace WindowRBD1.FormsCreate

{

public partial class CreatePicket : Window

{

public CreatePicket()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

NumberEquipment1();

Order1();

Methodology1();

Type1();

}

private void Type1() // Заполнение данными в ComboBox

{

txtTypeMeasurement.ItemsSource = new string[] { "Рядовое", "Контрольное", "Опытное" };

}

private void NumberEquipment1() //Собирает [Номер оборудования] которые существуют в сущности Proekt.Оборудования на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер оборудования] FROM Proekt.Оборудования";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberEquipment.Items.Clear();

while (reader.Read())

{

txtNumberEquipment.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void Methodology1() //Собирает [Номер методики] которые существуют в сущности Proekt.Методика на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер методики] FROM Proekt.Методика";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberMethodology.Items.Clear();

while (reader.Read())

{

txtNumberMethodology.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void Order1()//Собирает [Номер отряда] которые существуют в сущности Proekt.[Полевой отряд] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер отряда] FROM Proekt.[Полевой отряд]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberOrder.Items.Clear();

while (reader.Read())

{

txtNumberOrder.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btCreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "insert into Proekt.Пикет([Наименование пикета],[Координата],[Вид измерения],[Номер отряда],[Номер методики],[Номер оборудования])" +

$"values ('{txtNamePicket.Text}','{txtCoordinate.Text}','{txtTypeMeasurement.Text}', '{txtNumberOrder.Text}','{txtNumberMethodology.Text}','{txtNumberEquipment.Text}')";

SqlCommand comm = new SqlCommand(sql, conn);

comm.ExecuteNonQuery();

MessageBox.Show("Запись добавлена");

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

CreateMethodology tv = new CreateMethodology();

tv.ShowDialog();

}

private void btAdd2\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

CreateOrder tv = new CreateOrder();

tv.ShowDialog();

}

private void btAdd3\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

CreateEquipment tv = new CreateEquipment();

tv.ShowDialog();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

NumberMet met = new NumberMet();

met.ShowDialog();

}

private void btPoisk2\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

NumberOrder met = new NumberOrder();

met.ShowDialog();

}

private void btPoisk3\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

NumberEq met = new NumberEq();

met.ShowDialog();

}

}

}

using iTextSharp.text.pdf;

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Net.Sockets;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.FormsCreate

{

public partial class CreateProekt : Window

{

public CreateProekt()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

NumberArea1();

NumberClient1();

NumberContract1();

}

private void NumberClient1() //Собирает [Номер заказчика] которые существуют в сущности Proekt.[Заказчик] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер заказчика] FROM Proekt.Заказчик";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

NumberClient.Items.Clear();

while (reader.Read())

{

NumberClient.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void NumberContract1() //Собирает [Номер договора] которые существуют в сущности Proekt.Договор на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер договора] FROM Proekt.Договор";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

NumberContract.Items.Clear();

while (reader.Read())

{

NumberContract.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void NumberArea1()//Собирает [Номер площади] которые существуют в сущности Proekt.Площади на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер площади] FROM Proekt.Площади";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

NumberArea.Items.Clear();

while (reader.Read())

{

NumberArea.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void btCreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

int NumberArea2 = Convert.ToInt32(NumberArea.SelectedItem);

int NumberClient2 = Convert.ToInt32(NumberClient.SelectedItem);

int NumberContract2 = Convert.ToInt32(NumberContract.SelectedItem);

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "insert into Proekt.Проект([Название проекта],[Номер заказчика],[Номер договора],[Номер площади],[Дата и время появления записи])" +

$"values ('{txtProekt.Text}','{NumberClient2}', '{NumberContract2}', '{NumberArea2}','{Time.Content}')";

SqlCommand comm = new SqlCommand(sql, conn);

comm.ExecuteNonQuery();

MessageBox.Show("Запись добавлена");

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

using WindowRBD1.FormsCreate.Works;

namespace WindowRBD1.FormsCreate

{

public partial class CreateProfile : Window

{

public CreateProfile()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

txtNumberPicket1();

}

private void btCreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "insert into Proekt.Профили([Наименование профиля],[Номер пикета],[Координаты начала],[Координаты изломов],[Координаты окончания],[Длина],[Дата и время появления записи])" +

$"values ('{txtNameProfile.Text}','{txtNumberPicket.Text}','{txtCoordinatesBeginning.Text}', '{txtCoordinatesFracture.Text}', '{txtCoordinatesEnd.Text}', '{txtLength.Text}','{Time.Content}')";

SqlCommand comm = new SqlCommand(sql, conn);

comm.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись добавлена");

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void txtNumberPicket1() //Собирает [Номер пикета] которые существуют в сущности Proekt.[Пикет] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер пикета] FROM Proekt.Пикет";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberPicket.Items.Clear();

while (reader.Read())

{

txtNumberPicket.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btPoisk2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

NumberPicket tv = new NumberPicket();

tv.ShowDialog();

}

private void btAdd2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreatePicket tv = new CreatePicket();

tv.ShowDialog();

}

}

}

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using static System.Net.Mime.MediaTypeNames;

namespace WindowRBD1.FormsCreate.Equipment

{

public partial class CreateGenerative : Window

{

public CreateGenerative()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void btCreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "INSERT INTO Proekt.[Генеративное оборудование]([Наименование],[Инвентарный номер],[Дата приобретения],[Дата поверки],[Характеристики]) VALUES(@txtName, @txtInventory, @dateVerification, @datePurchases, @txtCharacteristic)";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@txtName", txtName.Text);

cmd.Parameters.AddWithValue("@txtInventory", txtInventory.Text);

cmd.Parameters.AddWithValue("@dateVerification", dateVerification.SelectedDate);

cmd.Parameters.AddWithValue("@datePurchases", datePurchases.SelectedDate);

cmd.Parameters.AddWithValue("@txtCharacteristic", txtCharacteristic.Text);

cmd.ExecuteNonQuery();

MessageBox.Show("Запись добавлена");

conn.Close();

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.FormsCreate.Equipment

{

public partial class CreateMeasuring : Window

{

public CreateMeasuring()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void btCreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "INSERT INTO Proekt.[Измерительное оборудование]([Наименование],[Инвентарный номер],[Дата приобретения],[Дата поверки],[Характеристики]) VALUES(@txtName, @txtInventory, @dateVerification, @datePurchases, @txtCharacteristic)";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@txtName", txtName.Text);

cmd.Parameters.AddWithValue("@txtInventory", txtInventory.Text);

cmd.Parameters.AddWithValue("@dateVerification", dateVerification.SelectedDate);

cmd.Parameters.AddWithValue("@datePurchases", datePurchases.SelectedDate);

cmd.Parameters.AddWithValue("@txtCharacteristic", txtCharacteristic.Text);

cmd.ExecuteNonQuery();

MessageBox.Show("Запись добавлена");

conn.Close();

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.FormsCreate.Equipment

{

public partial class CreateTelemetry : Window

{

public CreateTelemetry()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void btCreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "INSERT INTO Proekt.[Телеметрическое оборудование]([Наименование],[Инвентарный номер],[Дата приобретения],[Дата поверки],[Характеристики]) VALUES(@txtName, @txtInventory, @dateVerification, @datePurchases, @txtCharacteristic)";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@txtName", txtName.Text);

cmd.Parameters.AddWithValue("@txtInventory", txtInventory.Text);

cmd.Parameters.AddWithValue("@dateVerification", dateVerification.SelectedDate);

cmd.Parameters.AddWithValue("@datePurchases", datePurchases.SelectedDate);

cmd.Parameters.AddWithValue("@txtCharacteristic", txtCharacteristic.Text);

cmd.ExecuteNonQuery();

MessageBox.Show("Запись добавлена");

conn.Close();

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

namespace WindowRBD1.FormsCreate.Meaning.intermediateResult

{

public partial class CreatefinalResult : Window

{

public CreatefinalResult()

{

InitializeComponent();

Picket1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void btcreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "INSERT INTO Proekt.[Окончательный результат]([Номер Пикета],[Индекс Пикета],[Значения измерения(ЭДС)]) VALUES(@txtNumberPicket,@PicketIndex, @Meaning)";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@PicketIndex", PicketIndex.Text);

cmd.Parameters.AddWithValue("@txtNumberPicket", txtNumberPicket.SelectedItem);

cmd.Parameters.AddWithValue("@Meaning", Meaning.Text);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись добавлена");

conn.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void Picket1() //Собирает [Номер пикета] которые существуют в сущности Proekt.Пикет на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер пикета] FROM Proekt.Пикет";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberPicket.Items.Clear();

while (reader.Read())

{

txtNumberPicket.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.Пикет";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Пикет");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Пикет"].DefaultView;

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreatePicket tv = new CreatePicket();

tv.ShowDialog();

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

namespace WindowRBD1.FormsCreate.Meaning

{

public partial class CreateTransformant1 : Window

{

public CreateTransformant1()

{

InitializeComponent();

Picket1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void btcreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "INSERT INTO Proekt.Трансформанта1([Номер Пикета],[Индекс Пикета],[Значения трансформанты (ρτ)]) VALUES(@txtNumberPicket,@PicketIndex, @Meaning)";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@PicketIndex", PicketIndex.Text);

cmd.Parameters.AddWithValue("@txtNumberPicket", txtNumberPicket.SelectedItem);

cmd.Parameters.AddWithValue("@Meaning", Meaning.Text);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись добавлена");

conn.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void Picket1() //Собирает [Номер пикета] которые существуют в сущности Proekt.Пикет на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер пикета] FROM Proekt.Пикет";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberPicket.Items.Clear();

while (reader.Read())

{

txtNumberPicket.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.Пикет";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Пикет");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Пикет"].DefaultView;

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreatePicket tv = new CreatePicket();

tv.ShowDialog();

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

namespace WindowRBD1.FormsCreate.Meaning

{

public partial class CreateTransformant2 : Window

{

public CreateTransformant2()

{

InitializeComponent();

Picket1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void btcreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "INSERT INTO Proekt.Трансформанта2([Номер Пикета],[Индекс Пикета],[Значения трансформанты (ρτ)]) VALUES(@txtNumberPicket,@PicketIndex, @Meaning)";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@PicketIndex", PicketIndex.Text);

cmd.Parameters.AddWithValue("@txtNumberPicket", txtNumberPicket.SelectedItem);

cmd.Parameters.AddWithValue("@Meaning", Meaning.Text);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись добавлена");

conn.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void Picket1() //Собирает [Номер пикета] которые существуют в сущности Proekt.Пикет на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер пикета] FROM Proekt.Пикет";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberPicket.Items.Clear();

while (reader.Read())

{

txtNumberPicket.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.Пикет";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Пикет");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Пикет"].DefaultView;

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreatePicket tv = new CreatePicket();

tv.ShowDialog();

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

namespace WindowRBD1.FormsCreate.Meaning

{

/// <summary>

/// Логика взаимодействия для Transformant1.xaml

/// </summary>

public partial class CreateTransformant3 : Window

{

public CreateTransformant3()

{

InitializeComponent();

Picket1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void btcreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "INSERT INTO Proekt.Трансформанта3([Номер Пикета],[Индекс Пикета],[Значения трансформанты (ρτ)]) VALUES(@txtNumberPicket,@PicketIndex, @Meaning)";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@PicketIndex", PicketIndex.Text);

cmd.Parameters.AddWithValue("@txtNumberPicket", txtNumberPicket.SelectedItem);

cmd.Parameters.AddWithValue("@Meaning", Meaning.Text);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись добавлена");

conn.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void Picket1() //Собирает [Номер пикета] которые существуют в сущности Proekt.Пикет на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер пикета] FROM Proekt.Пикет";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberPicket.Items.Clear();

while (reader.Read())

{

txtNumberPicket.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.Пикет";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Пикет");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Пикет"].DefaultView;

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreatePicket tv = new CreatePicket();

tv.ShowDialog();

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

namespace WindowRBD1.FormsCreate.Meaning.intermediateResult

{

public partial class CreateIntermediateResult : Window

{

public CreateIntermediateResult()

{

InitializeComponent();

Picket1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void btcreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "INSERT INTO Proekt.[Промежуточный результат]([Номер Пикета],[Индекс Пикета],[Значения измерения(ЭДС)]) VALUES(@txtNumberPicket,@PicketIndex, @Meaning)";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@PicketIndex", PicketIndex.Text);

cmd.Parameters.AddWithValue("@txtNumberPicket", txtNumberPicket.SelectedItem);

cmd.Parameters.AddWithValue("@Meaning", Meaning.Text);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись добавлена");

conn.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void Picket1() //Собирает [Номер пикета] которые существуют в сущности Proekt.Пикет на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер пикета] FROM Proekt.Пикет";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberPicket.Items.Clear();

while (reader.Read())

{

txtNumberPicket.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.Пикет";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Пикет");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Пикет"].DefaultView;

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreatePicket tv = new CreatePicket();

tv.ShowDialog();

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

namespace WindowRBD1.FormsCreate.Meaning.intermediateResult

{

public partial class CreateIntermediateResult1 : Window

{

public CreateIntermediateResult1()

{

InitializeComponent();

Picket1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void btcreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "INSERT INTO Proekt.[Промежуточный результат 1]([Номер Пикета],[Индекс Пикета],[Значения измерения(ЭДС)]) VALUES(@txtNumberPicket,@PicketIndex, @Meaning)";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@PicketIndex", PicketIndex.Text);

cmd.Parameters.AddWithValue("@txtNumberPicket", txtNumberPicket.SelectedItem);

cmd.Parameters.AddWithValue("@Meaning", Meaning.Text);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись добавлена");

conn.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void Picket1() //Собирает [Номер пикета] которые существуют в сущности Proekt.Пикет на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер пикета] FROM Proekt.Пикет";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberPicket.Items.Clear();

while (reader.Read())

{

txtNumberPicket.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.Пикет";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Пикет");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Пикет"].DefaultView;

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreatePicket tv = new CreatePicket();

tv.ShowDialog();

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

namespace WindowRBD1.FormsCreate.Meaning.intermediateResult

{

public partial class CreateIntermediateResult2 : Window

{

public CreateIntermediateResult2()

{

InitializeComponent();

Picket1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void btcreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "INSERT INTO Proekt.[Промежуточный результат 2]([Номер Пикета],[Индекс Пикета],[Значения измерения(ЭДС)]) VALUES(@txtNumberPicket,@PicketIndex, @Meaning)";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@PicketIndex", PicketIndex.Text);

cmd.Parameters.AddWithValue("@txtNumberPicket", txtNumberPicket.SelectedItem);

cmd.Parameters.AddWithValue("@Meaning", Meaning.Text);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись добавлена");

conn.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void Picket1() //Собирает [Номер пикета] которые существуют в сущности Proekt.Пикет на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер пикета] FROM Proekt.Пикет";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberPicket.Items.Clear();

while (reader.Read())

{

txtNumberPicket.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.Пикет";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Пикет");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Пикет"].DefaultView;

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreatePicket tv = new CreatePicket();

tv.ShowDialog();

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

namespace WindowRBD1.FormsCreate.Meaning.intermediateResult

{

public partial class CreateIntermediateResult3 : Window

{

public CreateIntermediateResult3()

{

InitializeComponent();

Picket1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void btcreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "INSERT INTO Proekt.[Промежуточный результат 3]([Номер Пикета],[Индекс Пикета],[Значения измерения(ЭДС)]) VALUES(@txtNumberPicket,@PicketIndex, @Meaning)";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@PicketIndex", PicketIndex.Text);

cmd.Parameters.AddWithValue("@txtNumberPicket", txtNumberPicket.SelectedItem);

cmd.Parameters.AddWithValue("@Meaning", Meaning.Text);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись добавлена");

conn.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void Picket1() //Собирает [Номер пикета] которые существуют в сущности Proekt.Пикет на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер пикета] FROM Proekt.Пикет";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberPicket.Items.Clear();

while (reader.Read())

{

txtNumberPicket.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.Пикет";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Пикет");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Пикет"].DefaultView;

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreatePicket tv = new CreatePicket();

tv.ShowDialog();

}

}

}

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.FormsCreate.Works

{

public partial class CreateChief : Window

{

public CreateChief()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

}

private void btcreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "INSERT INTO Proekt.[Начальник отряда]([ФИО],[Квалификация],[Опыт работы в коллективе],[Общий опыт работы по специальности],[Дата прохождения медосмотра]) VALUES(@txtFIO, @txtCompetence, @txtExperience, @txtExperienced, @dateMedical)";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@txtFIO", txtFIO.Text);

cmd.Parameters.AddWithValue("@txtCompetence", txtCompetence.Text);

cmd.Parameters.AddWithValue("@txtExperience", txtExperience.Text);

cmd.Parameters.AddWithValue("@txtExperienced", txtExperienced.Text);

cmd.Parameters.AddWithValue("@dateMedical", dateMedical.SelectedDate);

cmd.ExecuteNonQuery();

MessageBox.Show("Запись добавлена");

conn.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

}

}

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.FormsCreate.Works

{

public partial class CreateDriver : Window

{

public CreateDriver()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

}

private void btcreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "INSERT INTO Proekt.[Водители]([ФИО],[Квалификация],[Опыт работы в коллективе],[Общий опыт работы по специальности],[Дата прохождения медосмотра]) VALUES(@txtFIO, @txtCompetence, @txtExperience, @txtExperienced, @dateMedical)";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@txtFIO", txtFIO.Text);

cmd.Parameters.AddWithValue("@txtCompetence", txtCompetence.Text);

cmd.Parameters.AddWithValue("@txtExperience", txtExperience.Text);

cmd.Parameters.AddWithValue("@txtExperienced", txtExperienced.Text);

cmd.Parameters.AddWithValue("@dateMedical", dateMedical.SelectedDate);

cmd.ExecuteNonQuery();

MessageBox.Show("Запись добавлена");

conn.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

}

}

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.FormsCreate.Works

{

public partial class CreateITR : Window

{

public CreateITR()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

}

private void btCreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "insert into Proekt.ИТР([ФИО],[Квалификация],[Опыт работы в коллективе],[Общий опыт работы по специальности],[Дата прохождения медосмотра])" +

$"values ('{txtFIO.Text}','{txtCompetence.Text}','{txtExperience.Text}','{txtExperienced.Text}','{dateMedical.SelectedDate}')";

SqlCommand comm = new SqlCommand(sql, conn);

comm.ExecuteNonQuery();

MessageBox.Show("Запись добавлена");

conn.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

}

}

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.FormsCreate.Works

{

public partial class CreateSupervisor : Window

{

public CreateSupervisor()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

}

private void btcreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "insert into Proekt.Супервайзер([ФИО],[Квалификация],[Опыт работы в коллективе],[Общий опыт работы по специальности],[Дата прохождения медосмотра])" +

$"values ('{txtFIO.Text}','{txtCompetence.Text}','{txtExperience.Text}','{txtExperienced.Text}','{dateMedical}')";

SqlCommand comm = new SqlCommand(sql, conn);

comm.ExecuteNonQuery();

MessageBox.Show("Запись добавлена");

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

}

}

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.FormsCreate.Works

{

public partial class CreateWorker : Window

{

public CreateWorker()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

}

private void btcreate\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "insert into Proekt.Рабочие([ФИО],[Квалификация],[Опыт работы в коллективе],[Общий опыт работы по специальности],[Дата прохождения медосмотра])" +

$"values ('{txtFIO.Text}','{txtCompetence.Text}','{txtExperience.Text}','{txtExperienced.Text}','{dateMedical}')";

SqlCommand comm = new SqlCommand(sql, conn);

comm.ExecuteNonQuery();

MessageBox.Show("Запись добавлена");

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

using WindowRBD1.FormsCreate;

namespace WindowRBD1.FormsEdit

{

public partial class EditArea : Window

{

public EditArea()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

txtList1();

txtArea1();

txtProfile1();

txtSupervisorOrder1();

txtSupervisorData1();

}

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для изменения введённых данных на Sql Server

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("UPDATE Proekt.Площади SET [Наименование площади] = @txtArea,[Номер профиля] = @txtProfile,[Номер списка углов периметра] =@txtList, [Длина периметра] =@txtPerimeterLength, [Величина площади окружённая периметром] =@txtSizeArea,[Дата и время начала работ] =@dateBeginning,[Дата и время окончания работ] =@dateEnd, [Дата и время изменения записи] =@Time, [Супервайзер полевых работ] =@txtSupervisorOrder, [Супервайзер обработки данных] =@txtSupervisorData where [Номер площади] = @txtNumber", con);

cmd.Parameters.AddWithValue("@txtNumber", txtNumber.SelectedItem);

cmd.Parameters.AddWithValue("@txtArea", txtArea.Text);

cmd.Parameters.AddWithValue("@txtProfile", txtProfile.Text);

cmd.Parameters.AddWithValue("@txtList", txtList.SelectedItem);

cmd.Parameters.AddWithValue("@txtSizeArea", txtSizeArea.Text);

cmd.Parameters.AddWithValue("@txtPerimeterLength", txtPerimeterLength.Text);

cmd.Parameters.AddWithValue("@txtSupervisorData", txtSupervisorData.Text);

cmd.Parameters.AddWithValue("@txtSupervisorOrder", txtSupervisorOrder.Text);

cmd.Parameters.AddWithValue("@dateBeginning", dateBeginning.SelectedDate);

cmd.Parameters.AddWithValue("@dateEnd", dateEnd.SelectedDate);

cmd.Parameters.AddWithValue("@Time", Time.Content);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

con.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void txtArea1() //Собирает [Номер площади] которые существуют в сущности Proekt.Площади на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер площади] FROM Proekt.Площади";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumber.Items.Clear();

while (reader.Read())

{

txtNumber.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void txtProfile1() //Собирает [Номер площади] которые существуют в сущности Proekt.Площади на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер профиля] FROM Proekt.Профили";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtProfile.Items.Clear();

while (reader.Read())

{

txtProfile.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void txtList1() //Собирает [Номер списка углов периметра] которые существуют в сущности Proekt.[Список координат углов периметра] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер списка углов периметра] FROM Proekt.[Список координат углов периметра]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtList.Items.Clear();

while (reader.Read())

{

txtList.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void txtNumber\_SelectionChanged(object sender, SelectionChangedEventArgs e) //При изменении значения происходит смена данных у всех элементов

{

string str = "Select \* from Proekt.Площади where [Номер площади] = " + txtNumber.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

DataTable dt = new DataTable();

da.Fill(dt);

txtArea.Text = dt.Rows[0][1].ToString();

txtProfile.Text = dt.Rows[0][2].ToString();

txtList.Text = dt.Rows[0][3].ToString();

txtPerimeterLength.Text = dt.Rows[0][4].ToString();

txtSizeArea.Text = dt.Rows[0][5].ToString();

dateBeginning.Text = dt.Rows[0][6].ToString();

dateEnd.Text = dt.Rows[0][7].ToString();

txtSupervisorData.Text = dt.Rows[0][10].ToString();

txtSupervisorOrder.Text = dt.Rows[0][11].ToString();

da.Dispose();

conn.Close();

}

}

private void txtSupervisorOrder1() //Собирает [Номера Супервайзеров] которые существуют в сущности Proekt.Супервайзер на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Супервайзера] FROM Proekt.Супервайзер";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtSupervisorOrder.Items.Clear();

while (reader.Read())

{

txtSupervisorOrder.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void txtSupervisorData1() //Собирает [Номера Супервайзеров] которые существуют в сущности Proekt.Супервайзер на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Супервайзера] FROM Proekt.Супервайзер";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtSupervisorData.Items.Clear();

while (reader.Read())

{

txtSupervisorData.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btPoisk\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

NumberProfile tv = new NumberProfile();

tv.ShowDialog();

}

private void btAdd\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

CreateProfile tv = new CreateProfile();

tv.ShowDialog();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

NumberArea tv = new NumberArea();

tv.ShowDialog();

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

CreateArea tv = new CreateArea();

tv.ShowDialog();

}

}

}

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using System.Security.Principal;

namespace WindowRBD1.FormsEdit

{

public partial class EditClient : Window

{

public EditClient()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

Client1();

}

private void Client1() //Собирает [Номер заказчика] которые существуют в сущности Proekt.Заказчик на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер заказчика] FROM Proekt.Заказчик";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumber.Items.Clear();

while (reader.Read())

{

txtNumber.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для изменения введённых данных на Sql Server

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("UPDATE Proekt.Заказчик SET [Название компаний] = @txtNameCompany,[Юридический адрес] =@txtLegalAddress,[Фактический адрес] =@txtActualAddress, [ИНН] =@txtINN,[КПК] = @txtPda, [Расчетный счёт] =@txtCalculated,[Корреспондентский счёт] =@txtCorrespondent, [Представитель] =@txtAgent,[Телефон представителя] = @txtPhone, [Адрес электронной почты] =@txtEmail,[Сайт] =@txtSite, [Дата и время изменения записи] =@Time where [Номер заказчика] = @txtNumber", con);

cmd.Parameters.AddWithValue("@txtNumber", txtNumber.SelectedItem);

cmd.Parameters.AddWithValue("@txtNameCompany", txtNameCompany.Text);

cmd.Parameters.AddWithValue("@txtLegalAddress", txtLegalAddress.Text);

cmd.Parameters.AddWithValue("@txtActualAddress", txtActualAddress.Text);

cmd.Parameters.AddWithValue("@txtINN", txtINN.Text);

cmd.Parameters.AddWithValue("@txtPda", txtPda.Text);

cmd.Parameters.AddWithValue("@txtCalculated", txtCalculated.Text);

cmd.Parameters.AddWithValue("@txtCorrespondent", txtCorrespondent.Text);

cmd.Parameters.AddWithValue("@txtAgent", txtAgent.Text);

cmd.Parameters.AddWithValue("@txtPhone", txtPhone.Text);

cmd.Parameters.AddWithValue("@txtEmail", txtEmail.Text);

cmd.Parameters.AddWithValue("@txtSite", txtSite.Text);

cmd.Parameters.AddWithValue("@Time", Time.Content);

cmd.ExecuteNonQuery();

MessageBox.Show("Запись изменена");

con.Close();

}

}

private void txtNameCompany\_SelectionChanged(object sender, SelectionChangedEventArgs e) //При изменении значения происходит смена данных у всех элементов

{

string str = "Select \* from Proekt.Заказчик where [Номер заказчика] = " + txtNumber.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

DataTable dt = new DataTable();

da.Fill(dt);

txtNameCompany.Text = dt.Rows[0][2].ToString();

txtLegalAddress.Text = dt.Rows[0][2].ToString();

txtActualAddress.Text = dt.Rows[0][3].ToString();

txtINN.Text = dt.Rows[0][4].ToString();

txtPda.Text = dt.Rows[0][5].ToString();

txtCalculated.Text = dt.Rows[0][6].ToString();

txtCorrespondent.Text = dt.Rows[0][7].ToString();

txtAgent.Text = dt.Rows[0][8].ToString();

txtPhone.Text = dt.Rows[0][9].ToString();

txtEmail.Text = dt.Rows[0][10].ToString();

txtSite.Text = dt.Rows[0][11].ToString();

da.Dispose();

conn.Close();

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.FormsEdit

{

public partial class EditContract : Window

{

public EditContract()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

Contract1();

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void Contract1() //Собирает [Номер договора] которые существуют в сущности Proekt.Договор на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер договора] FROM Proekt.Договор";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

Number.Items.Clear();

while (reader.Read())

{

Number.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для изменения введённых данных на Sql Server

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("UPDATE Proekt.Договор SET [Наименование договора] = @txtNameContract, [Начала] =@txtBeginnings,[Окончания] =@txtEndings, [Стоимость] =@txtCost,[Дата и время изменения записи] =@Time where [Номер договора] = @Number", con);

cmd.Parameters.AddWithValue("@Number", Number.SelectedItem);

cmd.Parameters.AddWithValue("@txtNameContract", txtNameContract.Text);

cmd.Parameters.AddWithValue("@txtBeginnings", txtBeginnings.Text);

cmd.Parameters.AddWithValue("@txtEndings", txtEndings.Text);

cmd.Parameters.AddWithValue("@txtCost", txtCost.Text);

cmd.Parameters.AddWithValue("@Time", Time.Content);

cmd.ExecuteNonQuery();

MessageBox.Show("Запись изменена");

con.Close();

}

}

private void txtNameContract\_SelectionChanged(object sender, SelectionChangedEventArgs e) //При изменении значения происходит смена данных у всех элементов

{

string str = "Select \* from Proekt.Договор where [Номер договора] = " + Number.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

DataTable dt = new DataTable();

da.Fill(dt);

txtNameContract.Text = dt.Rows[0][1].ToString();

txtBeginnings.Text = dt.Rows[0][2].ToString();

txtEndings.Text = dt.Rows[0][3].ToString();

txtCost.Text = dt.Rows[0][4].ToString();

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.Xml.Linq;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

namespace WindowRBD1.FormsEdit

{

public partial class EditEquipment : Window

{

public EditEquipment()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

cmbNumberEquipment1();

cmbNumberTelemetryEquipment1();

cmbNumberMesuringEquipment1();

cmbNumberGenEquipment1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void cmbNumberEquipment1() //Собирает [Номер Оборудования] которые существуют в сущности Proekt.Оборудования на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Оборудования] FROM Proekt.Оборудования";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

cmbNumberEquipment.Items.Clear();

while (reader.Read())

{

cmbNumberEquipment.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void cmbNumberGenEquipment1() //Собирает [Номер генеративного оборудования] которые существуют в сущности Proekt.[Генеративное оборудование] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер генеративного оборудования] FROM Proekt.[Генеративное оборудование]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

cmbNumberGenEquipment.Items.Clear();

while (reader.Read())

{

cmbNumberGenEquipment.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void cmbNumberMesuringEquipment1() //Собирает [Номер измерительного оборудования] которые существуют в сущности Proekt.[Измерительное оборудование] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер измерительного оборудования] FROM Proekt.[Измерительное оборудование]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

cmbNumberMesuringEquipment.Items.Clear();

while (reader.Read())

{

cmbNumberMesuringEquipment.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void cmbNumberTelemetryEquipment1() //Собирает [Номер Оборудования] которые существуют в сущности Proekt.Оборудования на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер телеметрического оборудования] FROM Proekt.[Телеметрическое оборудование]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

cmbNumberTelemetryEquipment.Items.Clear();

while (reader.Read())

{

cmbNumberTelemetryEquipment.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для изменения введённых данных на Sql Server

{

int cmbNumberEquipment2 = Convert.ToInt32(cmbNumberEquipment.SelectedItem);

int cmbNumberGenEquipment2 = Convert.ToInt32(cmbNumberGenEquipment.SelectedItem);

int cmbNumberMesuringEquipment2 = Convert.ToInt32(cmbNumberMesuringEquipment.SelectedItem);

int cmbNumberTelemetryEquipment2 = Convert.ToInt32(cmbNumberTelemetryEquipment.SelectedItem);

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("UPDATE Proekt.Оборудования SET [Номер генеративного оборудования] = @NumberGenEquipment, [Номер измерительного оборудования] = @NumberMesuringEquipment, [Номер телеметрического оборудования] = @NumberTelemetryEquipment where [Номер Оборудования] = @NumberEquipment", con);

cmd.Parameters.AddWithValue("@NumberEquipment", cmbNumberEquipment2);

cmd.Parameters.AddWithValue("@NumberGenEquipment", cmbNumberGenEquipment2);

cmd.Parameters.AddWithValue("@NumberMesuringEquipment", cmbNumberMesuringEquipment2);

cmd.Parameters.AddWithValue("@NumberTelemetryEquipment", cmbNumberTelemetryEquipment2);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

con.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select [Номер генеративного оборудования],[Наименование] from Proekt.[Генеративное оборудование]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Водители");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Водители"].DefaultView;

}

}

private void btPoisk2\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select [Номер измерительного оборудования],[Наименование] from Proekt.[Измерительное оборудование]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Рабочие");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Рабочие"].DefaultView;

}

}

private void btPoisk3\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select [Номер телеметрического оборудования],[Наименование] from Proekt.[Телеметрическое оборудование]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Супервайзер");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Супервайзер"].DefaultView;

}

}

private void btPoisk4\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select \* from Proekt.Оборудования";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Супервайзер");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Супервайзер"].DefaultView;

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

CreateEquipment tv = new CreateEquipment();

tv.ShowDialog();

}

private void btAdd2\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

CreateGenerative tv = new CreateGenerative();

tv.ShowDialog();

}

private void btAdd3\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

CreateMeasuring tv = new CreateMeasuring();

tv.ShowDialog();

}

private void btAdd4\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

CreateTelemetry tv = new CreateTelemetry();

tv.ShowDialog();

}

private void cmbNumberEquipment\_SelectionChanged\_1(object sender, SelectionChangedEventArgs e)

{

int cmbNumberEquipment2 = Convert.ToInt32(cmbNumberEquipment.SelectedItem);

string str = "Select \* from Proekt.Оборудования where [Номер Оборудования] = " + cmbNumberEquipment2;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

cmbNumberGenEquipment.Text = dr.Rows[0][1].ToString();

cmbNumberMesuringEquipment.Text = dr.Rows[0][2].ToString();

cmbNumberTelemetryEquipment.Text = dr.Rows[0][3].ToString();

da.Dispose();

conn.Close();

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using static System.Windows.Forms.VisualStyles.VisualStyleElement.Header;

using System.Configuration;

using System.Data.SqlTypes;

using Moq;

using Microsoft.Office.Interop.Excel;

using WindowRBD1.Forms;

using WindowRBD1.FormsCreate;

namespace WindowRBD1.FormsEdit

{

public partial class EditMethodology : System.Windows.Window

{

public EditMethodology()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время изменения записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

ListCategories();

ListCategories1();

ListCategories2();

}

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для изменения введённых данных на Sql Server

{

int test1 = Convert.ToInt32(txtGenerative.SelectedItem);

int test2 = Convert.ToInt32(txtMeasuring.SelectedItem);

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("UPDATE Proekt.Методика SET [Наименование методики] = @txtNameMethodology,[Номер описания ген.установки]= @txtGenerative, [Номер описания измер.установки] = @txtMeasuring, [Описание телеметрической установки] = @txtTelemetry,[Продолжительность импульса] = @txtImpulse, [Продолжительность паузы] = @txtPause,[Сила тока] = @txtCurrent where [Номер методики] = @txtNumber", con);

cmd.Parameters.AddWithValue("@txtNumber", cmbNumber.SelectedItem);

cmd.Parameters.AddWithValue("@txtNameMethodology", cmbNameMethodology.Text);

cmd.Parameters.AddWithValue("@txtGenerative", test1);

cmd.Parameters.AddWithValue("@txtMeasuring", test2);

cmd.Parameters.AddWithValue("@txtTelemetry", txtTelemetry.Text);

cmd.Parameters.AddWithValue("@txtImpulse", txtImpulse.Text);

cmd.Parameters.AddWithValue("@txtPause", txtPause.Text);

cmd.Parameters.AddWithValue("@txtCurrent", txtCurrent.Text);

cmd.ExecuteNonQuery();

MessageBox.Show("Запись изменена");

con.Close();

}

}

private void ListCategories()//Собирает [Номер методики] которые существуют в сущности Proekt.Методика на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер методики] FROM Proekt.Методика";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

cmbNumber.Items.Clear();

while (reader.Read())

{

cmbNumber.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void ListCategories1() //Собирает [Номер описания ген.установки] которые существуют в сущности Proekt.[Описание генераторной установки] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер описания ген.установки] FROM Proekt.[Описание генераторной установки]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtGenerative.Items.Clear();

while (reader.Read())

{

txtGenerative.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void ListCategories2() //Собирает [Номер описания измер.установки] которые существуют в сущности Proekt.[Описание измерительной установки] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер описания измер.установки] FROM Proekt.[Описание измерительной установки]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtMeasuring.Items.Clear();

while (reader.Read())

{

txtMeasuring.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

NumberADDGen tr = new NumberADDGen();

tr.ShowDialog();

}

private void btAdd2\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

NumberADDMeauring tr = new NumberADDMeauring();

tr.ShowDialog();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

NumberGenerative tv = new NumberGenerative();

tv.ShowDialog();

}

private void btPoisk2\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

NumberMeasuring tv = new NumberMeasuring();

tv.ShowDialog();

}

private void btAdd3\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

CreateMethodology tr = new CreateMethodology();

tr.ShowDialog();

}

private void btPoisk3\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

NumberMethodology tv = new NumberMethodology();

tv.ShowDialog();

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void cmbNameMethodology\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

string str = "Select \* from Proekt.Методика where [Номер методики] = " + cmbNumber.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

cmbNameMethodology.Text = dr.Rows[0][1].ToString();

txtGenerative.Text = dr.Rows[0][2].ToString();

txtMeasuring.Text = dr.Rows[0][3].ToString();

txtTelemetry.Text = dr.Rows[0][4].ToString();

txtImpulse.Text = dr.Rows[0][5].ToString();

txtPause.Text = dr.Rows[0][6].ToString();

txtCurrent.Text = dr.Rows[0][7].ToString();

da.Dispose();

conn.Close();

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Security.Policy;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Controls.Primitives;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Works;

using static System.Net.Mime.MediaTypeNames;

using static System.Windows.Forms.MonthCalendar;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace WindowRBD1.FormsEdit {

public partial class EditOrder : System.Windows.Window

{

public EditOrder()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

NumberOrder1();

FioChief1();

FioWorker1();

FioDriver1();

FioSupervisor1();

FioITR1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void NumberOrder1() //Собирает [Номер отряда] которые существуют в сущности Proekt.[Полевой отряд] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер отряда] FROM Proekt.[Полевой отряд]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

NumberOrder.Items.Clear();

while (reader.Read())

{

NumberOrder.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void FioChief1() //Собирает [Номер Начальника] которые существуют в сущности Proekt.[Начальник отряда] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Начальника] FROM Proekt.[Начальник отряда]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

FioChief.Items.Clear();

while (reader.Read())

{

FioChief.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void FioDriver1() //Собирает [Номер Водителя] которые существуют в сущности Proekt.[Водители] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Водителя] FROM Proekt.[Водители]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

FioDriver.Items.Clear();

while (reader.Read())

{

FioDriver.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void FioITR1() //Собирает [Номер ИТР] которые существуют в сущности Proekt.[ИТР] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер ИТР] FROM Proekt.[ИТР]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

FioITR.Items.Clear();

while (reader.Read())

{

FioITR.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void FioSupervisor1() //Собирает [Номер Супервайзера] которые существуют в сущности Proekt.[Супервайзер] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Супервайзера] FROM Proekt.[Супервайзер]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

FioSupervisor.Items.Clear();

while (reader.Read())

{

FioSupervisor.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void FioWorker1() //Собирает [Номер Рабочего] которые существуют в сущности Proekt.[Рабочие] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Рабочего] FROM Proekt.[Рабочие]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

FioWorker.Items.Clear();

while (reader.Read())

{

FioWorker.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для изменения введённых данных на Sql Server

{

int NumberOrder2 = Convert.ToInt32(NumberOrder.SelectedItem);

int FioWorker2 = Convert.ToInt32(FioWorker.SelectedItem);

int FioITR2 = Convert.ToInt32(FioITR.SelectedItem);

int FioSupervisor2 = Convert.ToInt32(FioSupervisor.SelectedItem);

int FioDriver2 = Convert.ToInt32(FioDriver.SelectedItem);

int FioChief2 = Convert.ToInt32(FioChief.SelectedItem);

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("UPDATE Proekt.[Полевой отряд] SET [Номер Начальника] = @FioChief, [Номер ИТР] = @FioITR, [Номер Водителя] = @FioDriver, [Номер Супервайзера] = @FioSupervisor, [Номер Рабочего] =@FioWorker where [Номер отряда] = @NumberOrder", con);

cmd.Parameters.AddWithValue("@NumberOrder", NumberOrder2);

cmd.Parameters.AddWithValue("@FioChief", FioChief2);

cmd.Parameters.AddWithValue("@FioDriver", FioDriver2);

cmd.Parameters.AddWithValue("@FioITR", FioITR2);

cmd.Parameters.AddWithValue("@FioSupervisor", FioSupervisor2);

cmd.Parameters.AddWithValue("@FioWorker", FioWorker2);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

con.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void NumberOrder\_SelectionChanged(object sender, SelectionChangedEventArgs e) //При изменении значения происходит смена данных у всех элементов

{

int NumberOrder2 = Convert.ToInt32(NumberOrder.SelectedItem);

string str = "Select \* from Proekt.[Полевой отряд] where [Номер отряда] = " + NumberOrder2;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

FioChief.Text = dr.Rows[0][1].ToString();

FioITR.Text = dr.Rows[0][2].ToString();

FioDriver.Text = dr.Rows[0][3].ToString();

FioWorker.Text = dr.Rows[0][4].ToString();

FioSupervisor.Text = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btPoisk2\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select [Номер Начальника],ФИО from Proekt.[Начальник отряда]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.[Начальник отряда]");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.[Начальник отряда]"].DefaultView;

}

}

private void btPoisk3\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select [Номер ИТР],ФИО from Proekt.ИТР";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.ИТР");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.ИТР"].DefaultView;

}

}

private void btPoisk4\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select [Номер Водителя],ФИО from Proekt.Водители";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Водители");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Водители"].DefaultView;

}

}

private void btPoisk5\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select [Номер Рабочего],ФИО from Proekt.Рабочие";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Рабочие");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Рабочие"].DefaultView;

}

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select \* from Proekt.[Полевой отряд]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.[Полевой отряд]");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.[Полевой отряд]"].DefaultView;

}

}

private void btPoisk6\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select [Номер Супервайзера],ФИО from Proekt.Супервайзер";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Супервайзер");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Супервайзер"].DefaultView;

}

}

private void btAdd2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateChief tv = new CreateChief();

tv.ShowDialog();

}

private void btAdd3\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateITR tv = new CreateITR();

tv.ShowDialog();

}

private void btAdd4\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateDriver tv = new CreateDriver();

tv.ShowDialog();

}

private void btAdd5\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateWorker tv = new CreateWorker();

tv.ShowDialog();

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateOrder tv = new CreateOrder();

tv.ShowDialog();

}

private void btAdd6\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateSupervisor tv = new CreateSupervisor();

tv.ShowDialog();

}

}

}

using System;

using System.Collections.Generic;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

using WindowRBD1.FormsCreate;

namespace WindowRBD1.FormsEdit {

public partial class EditPicket : Window

{

public EditPicket()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

Picket1();

Methodology1();

Order1();

Equipment1();

Type1();

}

private void Type1() // Заполнение данными в ComboBox

{

txtTypeMeasurement.ItemsSource = new string[] { "Рядовое", "Контрольное", "Опытное" };

}

private void Methodology1() //Собирает [Номер методики] которые существуют в сущности Proekt.Методика на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер методики] FROM Proekt.Методика";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

NumberMethodology.Items.Clear();

while (reader.Read())

{

NumberMethodology.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void Order1() //Собирает [Номер отряда] которые существуют в сущности Proekt.[Полевой отряд] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер отряда] FROM Proekt.[Полевой отряд]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

NumberOrder.Items.Clear();

while (reader.Read())

{

NumberOrder.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void Picket1() //Собирает [Номер Пикета] которые существуют в сущности Proekt.Пикет на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Пикета] FROM Proekt.Пикет";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumber.Items.Clear();

while (reader.Read())

{

txtNumber.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void Equipment1() //Собирает [Номер оборудования] которые существуют в сущности Proekt.Оборудования на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер оборудования] FROM Proekt.Оборудования";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

NumberEquipment.Items.Clear();

while (reader.Read())

{

NumberEquipment.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для изменения введённых данных на Sql Server

{

int NumberOrder2 = Convert.ToInt32(NumberOrder.SelectedItem);

int NumberMethodology2 = Convert.ToInt32(NumberMethodology.SelectedItem);

int NumberEquipment2 = Convert.ToInt32(NumberEquipment.SelectedItem);

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("UPDATE Proekt.Пикет SET [Наименование пикета] = @NamePicket,[Координата]=@Coordinate,[Вид измерения]=@TypeMeasurement,[Номер отряда]=@NumberOrder,[Номер методики]=@NumberMethodology,[Номер оборудования] =@NumberEquipment where [Номер Пикета] = @Number", con);

cmd.Parameters.AddWithValue("@NamePicket", txtNamePicket.Text);

cmd.Parameters.AddWithValue("@Coordinate", txtCoordinate);

cmd.Parameters.AddWithValue("@TypeMeasurement", txtTypeMeasurement);

cmd.Parameters.AddWithValue("@Number", txtNumber.SelectedItem);

cmd.Parameters.AddWithValue("@NumberOrder", NumberOrder2);

cmd.Parameters.AddWithValue("@NumberMethodology", NumberMethodology2);

cmd.Parameters.AddWithValue("@NumberEquipment", NumberEquipment2);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

con.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

NumberMet met = new NumberMet();

met.ShowDialog();

}

private void btPoisk2\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

NumberOrder met = new NumberOrder();

met.ShowDialog();

}

private void btPoisk3\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

NumberEq met = new NumberEq();

met.ShowDialog();

}

private void txtNumber\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

string str = "Select \* from Proekt.Пикет where [Номер Пикета] = " + txtNumber.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtNamePicket.Text = dr.Rows[0][1].ToString();

txtCoordinate.Text = dr.Rows[0][2].ToString();

txtTypeMeasurement.Text = dr.Rows[0][3].ToString();

NumberOrder.Text = dr.Rows[0][4].ToString();

NumberMethodology.Text = dr.Rows[0][5].ToString();

NumberEquipment.Text = dr.Rows[0][6].ToString();

da.Dispose();

conn.Close();

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

CreateMethodology tv = new CreateMethodology();

tv.ShowDialog();

}

private void btAdd2\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

CreateOrder tv = new CreateOrder();

tv.ShowDialog();

}

private void btAdd3\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

CreateEquipment tv = new CreateEquipment();

tv.ShowDialog();

}

private void btAdd4\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

CreatePicket tv = new CreatePicket();

tv.ShowDialog();

}

private void btPoisk4\_Click(object sender, RoutedEventArgs e) //Открытие формы

{

}

}

}

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

namespace WindowRBD1.FormsEdit

{

public partial class EditProekt : Window

{

public EditProekt()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

NumberArea1();

NumberClient1();

NumberProekt1();

NumberContract1();

}

private void NumberProekt1() //Собирает [Номер проекта] которые существуют в сущности Proekt.Проект на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер проекта] FROM Proekt.Проект";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

Number.Items.Clear();

while (reader.Read())

{

Number.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void NumberClient1() //Собирает [Номер заказчика] которые существуют в сущности Proekt.Заказчик на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер заказчика] FROM Proekt.Заказчик";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

NumberClient.Items.Clear();

while (reader.Read())

{

NumberClient.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void NumberContract1() //Собирает [Номер договора] которые существуют в сущности Proekt.Договор на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер договора] FROM Proekt.Договор";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

NumberContract.Items.Clear();

while (reader.Read())

{

NumberContract.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void NumberArea1() //Собирает [Номер площади] которые существуют в сущности Proekt.Площади на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер площади] FROM Proekt.Площади";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

NumberArea.Items.Clear();

while (reader.Read())

{

NumberArea.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для изменения введённых данных на Sql Server

{

int NumberClient2 = Convert.ToInt32(NumberClient.SelectedItem);

int NumberArea2 = Convert.ToInt32(NumberArea.SelectedItem);

int NumberContract2 = Convert.ToInt32(NumberContract.SelectedItem);

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("UPDATE Proekt.Проект SET [Название проекта] = @NameProekt,[Номер площади]=@NumberArea,[Номер заказчика]=@NumberClient,[Номер договора]=@NumberContract where [Номер проекта] = @Number", con);

cmd.Parameters.AddWithValue("@NumberClient", NumberClient2);

cmd.Parameters.AddWithValue("@NumberArea", NumberArea2);

cmd.Parameters.AddWithValue("@NumberContract", NumberContract2);

cmd.Parameters.AddWithValue("@Number", Number.SelectedItem);

cmd.Parameters.AddWithValue("@NameProekt", NameProekt.Text);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

con.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void NameProekt\_SelectionChanged(object sender, SelectionChangedEventArgs e) //При изменении значения происходит смена данных у всех элементов

{

string str = "Select \* from Proekt.Проект where [Номер проекта] = " + Number.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

NameProekt.Text = dr.Rows[0][1].ToString();

NumberClient.Text = dr.Rows[0][2].ToString();

NumberContract.Text = dr.Rows[0][3].ToString();

NumberArea.Text = dr.Rows[0][4].ToString();

da.Dispose();

conn.Close();

}

}

}

}

using iTextSharp.text.pdf;

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

using WindowRBD1.FormsCreate;

namespace WindowRBD1.FormsEdit

{

public partial class EditProfile : Window

{

public EditProfile()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

Picket1();

NameProfile1();

}

private void Picket1() //Собирает [Номер пикета] которые существуют в сущности Proekt.Пикет на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер пикета] FROM Proekt.Пикет";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberPicket.Items.Clear();

while (reader.Read())

{

txtNumberPicket.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void NameProfile1() //Собирает [Наименование профиля] которые существуют в сущности Proekt.Профили на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер профиля] FROM Proekt.Профили";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumber.Items.Clear();

while (reader.Read())

{

txtNumber.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void txtNameProfile\_SelectionChanged\_1(object sender, SelectionChangedEventArgs e) //При изменении значения происходит смена данных у всех элементов

{

string str = "Select \* from Proekt.Профили where [Номер профиля] = " + txtNumber.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtNameProfile.Text = dr.Rows[0][1].ToString();

txtNumberPicket.Text = dr.Rows[0][2].ToString();

txtCoordinatesBeginning.Text = dr.Rows[0][3].ToString();

txtCoordinatesFracture.Text = dr.Rows[0][4].ToString();

txtCoordinatesEnd.Text = dr.Rows[0][5].ToString();

txtLength.Text = dr.Rows[0][6].ToString();

dateBen.Text = dr.Rows[0][7].ToString();

dateEnd.Text = dr.Rows[0][8].ToString();

da.Dispose();

conn.Close();

}

}

private void btClose\_Click\_1(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для изменения введённых данных на Sql Server

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("UPDATE Proekt.Профили SET [Наименование профиля] = @txtNameProfile,[Номер пикета]=@NumberPicket,[Координаты начала]=@CoordinatesBeginning,[Координаты изломов]=@CoordinatesFracture,[Координаты окончания]=@CoordinatesEnd,[Длина]=@Length,[Дата и время начала работ]=@dateBen,[Дата и время окончания работ]=@dateEnd,[Дата и время изменения записи]=@Time where [Номер профиля] = @Number", con);

cmd.Parameters.AddWithValue("@Number", txtNumber.SelectedItem);

cmd.Parameters.AddWithValue("@txtNameProfile", txtNameProfile.Text);

cmd.Parameters.AddWithValue("@NumberPicket", txtNumberPicket.SelectedItem);

cmd.Parameters.AddWithValue("@CoordinatesBeginning", txtCoordinatesBeginning.Text);

cmd.Parameters.AddWithValue("@CoordinatesFracture", txtCoordinatesFracture.Text);

cmd.Parameters.AddWithValue("@CoordinatesEnd", txtCoordinatesEnd.Text);

cmd.Parameters.AddWithValue("@Length", txtLength.Text);

cmd.Parameters.AddWithValue("@dateBen", dateBen.SelectedDate);

cmd.Parameters.AddWithValue("@dateEnd", dateEnd.SelectedDate);

cmd.Parameters.AddWithValue("@Time", Time.Content);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

con.Close();

}

}

private void btPoisk2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

NumberPicket tv = new NumberPicket();

tv.ShowDialog();

}

private void btAdd2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreatePicket tv = new CreatePicket();

tv.ShowDialog();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

NumberProfile tv = new NumberProfile();

tv.ShowDialog();

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateProfile tv = new CreateProfile();

tv.ShowDialog();

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate.Equipment;

namespace WindowRBD1.FormsEdit.Equipment

{

public partial class EditGenerative : Window

{

public EditGenerative()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

txtName1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void txtName1() //Собирает [Наименование] которые существуют в сущности Proekt.[Генеративное оборудование] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер генеративного оборудования] FROM Proekt.[Генеративное оборудование]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumber.Items.Clear();

while (reader.Read())

{

txtNumber.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void txtName\_SelectionChanged(object sender, SelectionChangedEventArgs e) //При изменении значения происходит смена данных у всех элементов

{

//int txtName2 = Convert.ToInt32(txtName.SelectedItem);

string str = "Select \* from Proekt.[Генеративное оборудование] where [Номер генеративного оборудования] = " + txtNumber.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtName.Text = dr.Rows[0][1].ToString();

txtInventory.Text = dr.Rows[0][2].ToString();

dateVerification.Text = dr.Rows[0][3].ToString();

datePurchases.Text = dr.Rows[0][4].ToString();

txtCharacteristic.Text = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для изменения введённых данных на Sql Server

{

//int cmbNumberEquipment2 = Convert.ToInt32(txtName.SelectedItem);

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("UPDATE Proekt.[Генеративное оборудование] SET [Наименование] = @txtName,[Инвентарный номер] = @Inventory,[Дата приобретения]=@dateVerification,[Дата поверки]=@datePurchases, [Характеристики] = @Characteristic where [Номер генеративного оборудования] = @txtNumber", con);

cmd.Parameters.AddWithValue("@txtNumber", txtNumber.SelectedItem);

cmd.Parameters.AddWithValue("@txtName", txtName.Text);

cmd.Parameters.AddWithValue("@Inventory", txtInventory.Text);

cmd.Parameters.AddWithValue("@Characteristic", txtCharacteristic.Text);

cmd.Parameters.AddWithValue("@dateVerification", dateVerification.SelectedDate);

cmd.Parameters.AddWithValue("@datePurchases", datePurchases.SelectedDate);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

con.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void btPoisk\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select [Номер генеративного оборудования],[Наименование] from Proekt.[Генеративное оборудование]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.[Генеративное оборудование]");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.[Генеративное оборудование]"].DefaultView;

}

}

private void btAdd\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

CreateGenerative tv = new CreateGenerative();

tv.ShowDialog();

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate.Equipment;

namespace WindowRBD1.FormsEdit.Equipment

{

public partial class EditMeasuring : Window

{

public EditMeasuring()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

txtName1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void txtName1() //Собирает [Наименование] которые существуют в сущности Proekt.[Измерительное оборудование] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер измерительного оборудования] FROM Proekt.[Измерительное оборудование]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumber.Items.Clear();

while (reader.Read())

{

txtNumber.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void txtName\_SelectionChanged(object sender, SelectionChangedEventArgs e) //При изменении значения происходит смена данных у всех элементов

{

string str = "Select \* from Proekt.[Измерительное оборудование] where [Номер измерительного оборудования] = " + txtNumber.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtName.Text = dr.Rows[0][1].ToString();

txtInventory.Text = dr.Rows[0][2].ToString();

dateVerification.Text = dr.Rows[0][3].ToString();

datePurchases.Text = dr.Rows[0][4].ToString();

txtCharacteristic.Text = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для изменения введённых данных на Sql Server

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("UPDATE Proekt.[Измерительное оборудование] SET [Наименование] = @txtName,[Инвентарный номер] = @Inventory,[Дата приобретения]=@dateVerification,[Дата поверки]=@datePurchases, [Характеристики] = @Characteristic where [Номер измерительного оборудования] = @Number", con);

cmd.Parameters.AddWithValue("@Number", txtNumber.SelectedItem);

cmd.Parameters.AddWithValue("@txtName", txtName.Text);

cmd.Parameters.AddWithValue("@Inventory", txtInventory.Text);

cmd.Parameters.AddWithValue("@Characteristic", txtCharacteristic.Text);

cmd.Parameters.AddWithValue("@dateVerification", dateVerification.SelectedDate);

cmd.Parameters.AddWithValue("@datePurchases", datePurchases.SelectedDate);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

con.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void btPoisk\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select [Номер измерительного оборудования],[Наименование] from Proekt.[Измерительное оборудование]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.[Измерительное оборудование]");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.[Измерительное оборудование]"].DefaultView;

}

}

private void btAdd\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

CreateMeasuring tv = new CreateMeasuring();

tv.ShowDialog();

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate.Equipment;

namespace WindowRBD1.FormsEdit.Equipment

{

public partial class EditTelemetry : Window

{

public EditTelemetry()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

txtName1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void txtName1() //Собирает [Наименование] которые существуют в сущности Proekt.[Генеративное оборудование] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер телеметрического оборудования] FROM Proekt.[Телеметрическое оборудование]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumber.Items.Clear();

while (reader.Read())

{

txtNumber.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void txtName\_SelectionChanged(object sender, SelectionChangedEventArgs e) //При изменении значения происходит смена данных у всех элементов

{

string str = "Select \* from Proekt.[Телеметрическое оборудование] where [Номер телеметрического оборудования] = " + txtNumber.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtName.Text = dr.Rows[0][1].ToString();

txtInventory.Text = dr.Rows[0][2].ToString();

dateVerification.Text = dr.Rows[0][3].ToString();

datePurchases.Text = dr.Rows[0][4].ToString();

txtCharacteristic.Text = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для изменения введённых данных на Sql Server

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("UPDATE Proekt.[Телеметрическое оборудование] SET [Наименование] = @txtName,[Инвентарный номер] = @Inventory,[Дата приобретения]=@dateVerification,[Дата поверки]=@datePurchases, [Характеристики] = @Characteristic where [Номер телеметрического оборудования] = @txtNumber", con);

cmd.Parameters.AddWithValue("@txtNumber", txtNumber.SelectedItem);

cmd.Parameters.AddWithValue("@txtName", txtName.Text);

cmd.Parameters.AddWithValue("@Inventory", txtInventory.Text);

cmd.Parameters.AddWithValue("@Characteristic", txtCharacteristic.Text);

cmd.Parameters.AddWithValue("@dateVerification", dateVerification.SelectedDate);

cmd.Parameters.AddWithValue("@datePurchases", datePurchases.SelectedDate);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

con.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void btPoisk\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select [Номер телеметрического оборудования],[Наименование] from Proekt.[Телеметрическое оборудование]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.[Телеметрическое оборудование]");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.[Телеметрическое оборудование]"].DefaultView;

}

}

private void btAdd\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

CreateTelemetry tv = new CreateTelemetry();

tv.ShowDialog();

}

}

}

using DocumentFormat.OpenXml.Presentation;

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Meaning;

using WindowRBD1.FormsCreate.Meaning.intermediateResult;

using static System.Windows.Forms.MonthCalendar;

namespace WindowRBD1.FormsEdit.Meaning

{

public partial class EditfinalResult : Window

{

public EditfinalResult()

{

InitializeComponent();

Picket1();

Number1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "UPDATE Proekt.[Окончательный результат] SET [Номер Пикета] = @txtNumberPicket,[Индекс Пикета] = @PicketIndex,[Значения измерения(ЭДС)] = @Meaning where [Номер Окончательного результата] = @NumberTrans1 ";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@NumberTrans1", NumberTrans1.SelectedItem);

cmd.Parameters.AddWithValue("@PicketIndex", PicketIndex.Text);

cmd.Parameters.AddWithValue("@txtNumberPicket", txtNumberPicket.SelectedItem);

cmd.Parameters.AddWithValue("@Meaning", Meaning.Text);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

conn.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void Picket1() //Собирает [Номер пикета] которые существуют в сущности Proekt.Пикет на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер пикета] FROM Proekt.Пикет";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberPicket.Items.Clear();

while (reader.Read())

{

txtNumberPicket.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void Number1() //Собирает [Номер Трансформанты измерения 1] которые существуют в сущности Proekt.Трансформанта1 на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Окончательнго результата] FROM Proekt.[Окончательный результат]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

NumberTrans1.Items.Clear();

while (reader.Read())

{

NumberTrans1.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.Пикет";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Пикет");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Пикет"].DefaultView;

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreatePicket tv = new CreatePicket();

tv.ShowDialog();

}

private void btPoisk2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.[Окончательный результат]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.[Окончательный результат]");

dataGridView1.ItemsSource = ds.Tables["Proekt.[Окончательный результат]"].DefaultView;

}

}

private void btAdd2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreatefinalResult tv = new CreatefinalResult();

tv.ShowDialog();

}

private void NumberTrans1\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

string str = "Select \* from Proekt.[Окончательный результат] where [Номер Окончательнго результата] = " + NumberTrans1.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

DataTable dt = new DataTable();

da.Fill(dt);

txtNumberPicket.Text = dt.Rows[0][1].ToString();

PicketIndex.Text = dt.Rows[0][2].ToString();

Meaning.Text = dt.Rows[0][3].ToString();

da.Dispose();

conn.Close();

}

}

}

}

using DocumentFormat.OpenXml.Presentation;

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Meaning;

using static System.Windows.Forms.MonthCalendar;

namespace WindowRBD1.FormsEdit.Meaning

{

public partial class EditTransformant1 : Window

{

public EditTransformant1()

{

InitializeComponent();

Picket1();

Number1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "UPDATE Proekt.Трансформанта1 SET [Номер Пикета] = @txtNumberPicket,[Индекс Пикета] = @PicketIndex,[Значения трансформанты (ρτ)] = @Meaning where [Номер Трансформанты измерения 1] = @NumberTrans1 ";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@NumberTrans1", NumberTrans1.SelectedItem);

cmd.Parameters.AddWithValue("@PicketIndex", PicketIndex.Text);

cmd.Parameters.AddWithValue("@txtNumberPicket", txtNumberPicket.SelectedItem);

cmd.Parameters.AddWithValue("@Meaning", Meaning.Text);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

conn.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void Picket1() //Собирает [Номер пикета] которые существуют в сущности Proekt.Пикет на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер пикета] FROM Proekt.Пикет";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberPicket.Items.Clear();

while (reader.Read())

{

txtNumberPicket.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void Number1() //Собирает [Номер Трансформанты измерения 1] которые существуют в сущности Proekt.Трансформанта1 на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Трансформанты измерения 1] FROM Proekt.Трансформанта1";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

NumberTrans1.Items.Clear();

while (reader.Read())

{

NumberTrans1.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.Пикет";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Пикет");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Пикет"].DefaultView;

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreatePicket tv = new CreatePicket();

tv.ShowDialog();

}

private void btPoisk2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.Трансформанта1";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Трансформанта1");

dataGridView1.ItemsSource = ds.Tables["Proekt.Трансформанта1"].DefaultView;

}

}

private void btAdd2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateTransformant1 tv = new CreateTransformant1();

tv.ShowDialog();

}

private void NumberTrans1\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

string str = "Select \* from Proekt.Трансформанта1 where [Номер Трансформанты измерения 1] = " + NumberTrans1.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

DataTable dt = new DataTable();

da.Fill(dt);

txtNumberPicket.Text = dt.Rows[0][1].ToString();

PicketIndex.Text = dt.Rows[0][2].ToString();

Meaning.Text = dt.Rows[0][3].ToString();

da.Dispose();

conn.Close();

}

}

}

}

using DocumentFormat.OpenXml.Presentation;

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Meaning;

using static System.Windows.Forms.MonthCalendar;

namespace WindowRBD1.FormsEdit.Meaning

{

public partial class EditTransformant2 : Window

{

public EditTransformant2()

{

InitializeComponent();

Picket1();

Number1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "UPDATE Proekt.Трансформанта2 SET [Номер Пикета] = @txtNumberPicket,[Индекс Пикета] = @PicketIndex,[Значения трансформанты (ρτ)] = @Meaning where [Номер Трансформанты измерения 2] = @NumberTrans2 ";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@NumberTrans2", NumberTrans2.SelectedItem);

cmd.Parameters.AddWithValue("@PicketIndex", PicketIndex.Text);

cmd.Parameters.AddWithValue("@txtNumberPicket", txtNumberPicket.SelectedItem);

cmd.Parameters.AddWithValue("@Meaning", Meaning.Text);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

conn.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void Picket1() //Собирает [Номер пикета] которые существуют в сущности Proekt.Пикет на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер пикета] FROM Proekt.Пикет";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberPicket.Items.Clear();

while (reader.Read())

{

txtNumberPicket.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void Number1() //Собирает [Номер Трансформанты измерения 2] которые существуют в сущности Proekt.Трансформанта1 на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Трансформанты измерения 2] FROM Proekt.Трансформанта2";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

NumberTrans2.Items.Clear();

while (reader.Read())

{

NumberTrans2.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.Пикет";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Пикет");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Пикет"].DefaultView;

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreatePicket tv = new CreatePicket();

tv.ShowDialog();

}

private void btPoisk2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.Трансформанта2";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Трансформанта2");

dataGridView1.ItemsSource = ds.Tables["Proekt.Трансформанта2"].DefaultView;

}

}

private void btAdd2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateTransformant2 tv = new CreateTransformant2();

tv.ShowDialog();

}

private void NumberTrans2\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

string str = "Select \* from Proekt.Трансформанта2 where [Номер Трансформанты измерения 2] = " + NumberTrans2.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

DataTable dt = new DataTable();

da.Fill(dt);

txtNumberPicket.Text = dt.Rows[0][1].ToString();

PicketIndex.Text = dt.Rows[0][2].ToString();

Meaning.Text = dt.Rows[0][3].ToString();

da.Dispose();

conn.Close();

}

}

}

}

using DocumentFormat.OpenXml.Presentation;

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Meaning;

using static System.Windows.Forms.MonthCalendar;

namespace WindowRBD1.FormsEdit.Meaning

{

public partial class EditTransformant3 : Window

{

public EditTransformant3()

{

InitializeComponent();

Picket1();

Number1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "UPDATE Proekt.Трансформанта3 SET [Номер Пикета] = @txtNumberPicket,[Индекс Пикета] = @PicketIndex,[Значения трансформанты (ρτ)] = @Meaning where [Номер Трансформанты измерения 3] = @NumberTrans2 ";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@NumberTrans3", NumberTrans3.SelectedItem);

cmd.Parameters.AddWithValue("@PicketIndex", PicketIndex.Text);

cmd.Parameters.AddWithValue("@txtNumberPicket", txtNumberPicket.SelectedItem);

cmd.Parameters.AddWithValue("@Meaning", Meaning.Text);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

conn.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void Picket1() //Собирает [Номер пикета] которые существуют в сущности Proekt.Пикет на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер пикета] FROM Proekt.Пикет";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberPicket.Items.Clear();

while (reader.Read())

{

txtNumberPicket.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void Number1() //Собирает [Номер Трансформанты измерения 2] которые существуют в сущности Proekt.Трансформанта1 на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Трансформанты измерения 3] FROM Proekt.Трансформанта3";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

NumberTrans3.Items.Clear();

while (reader.Read())

{

NumberTrans3.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.Пикет";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Пикет");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Пикет"].DefaultView;

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreatePicket tv = new CreatePicket();

tv.ShowDialog();

}

private void btPoisk2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.Трансформанта3";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Трансформанта3");

dataGridView1.ItemsSource = ds.Tables["Proekt.Трансформанта3"].DefaultView;

}

}

private void btAdd2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateTransformant3 tv = new CreateTransformant3();

tv.ShowDialog();

}

private void NumberTrans3\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

string str = "Select \* from Proekt.Трансформанта3 where [Номер Трансформанты измерения 3] = " + NumberTrans3.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

DataTable dt = new DataTable();

da.Fill(dt);

txtNumberPicket.Text = dt.Rows[0][1].ToString();

PicketIndex.Text = dt.Rows[0][2].ToString();

Meaning.Text = dt.Rows[0][3].ToString();

da.Dispose();

conn.Close();

}

}

}

}

using DocumentFormat.OpenXml.Presentation;

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Meaning;

using WindowRBD1.FormsCreate.Meaning.intermediateResult;

using static System.Windows.Forms.MonthCalendar;

namespace WindowRBD1.FormsEdit.Meaning

{

public partial class EditIntermediateResult : Window

{

public EditIntermediateResult()

{

InitializeComponent();

Picket1();

Number1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "UPDATE Proekt.[Промежуточный результат] SET [Номер Пикета] = @txtNumberPicket,[Индекс Пикета] = @PicketIndex,[Значения измерения(ЭДС)] = @Meaning where [Номер Промежуточного результата] = @NumberTrans1 ";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@NumberTrans1", NumberTrans1.SelectedItem);

cmd.Parameters.AddWithValue("@PicketIndex", PicketIndex.Text);

cmd.Parameters.AddWithValue("@txtNumberPicket", txtNumberPicket.SelectedItem);

cmd.Parameters.AddWithValue("@Meaning", Meaning.Text);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

conn.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void Picket1() //Собирает [Номер пикета] которые существуют в сущности Proekt.Пикет на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер пикета] FROM Proekt.Пикет";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberPicket.Items.Clear();

while (reader.Read())

{

txtNumberPicket.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void Number1() //Собирает [Номер Трансформанты измерения 1] которые существуют в сущности Proekt.Трансформанта1 на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Промежуточного результата] FROM Proekt.[Промежуточный результат]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

NumberTrans1.Items.Clear();

while (reader.Read())

{

NumberTrans1.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.Пикет";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Пикет");

dataGridView1.ItemsSource = ds.Tables["Proekt.Пикет"].DefaultView;

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreatePicket tv = new CreatePicket();

tv.ShowDialog();

}

private void btPoisk2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.[Промежуточный результат]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.[Промежуточный результат]");

dataGridView1.ItemsSource = ds.Tables["Proekt.[Промежуточный результат]"].DefaultView;

}

}

private void btAdd2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateIntermediateResult tv = new CreateIntermediateResult();

tv.ShowDialog();

}

private void NumberTrans1\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

string str = "Select \* from Proekt.[Промежуточный результат] where [Номер Промежуточного результата] = " + NumberTrans1.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

DataTable dt = new DataTable();

da.Fill(dt);

txtNumberPicket.Text = dt.Rows[0][1].ToString();

PicketIndex.Text = dt.Rows[0][2].ToString();

Meaning.Text = dt.Rows[0][3].ToString();

da.Dispose();

conn.Close();

}

}

}

}

using DocumentFormat.OpenXml.Presentation;

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Meaning;

using WindowRBD1.FormsCreate.Meaning.intermediateResult;

using static System.Windows.Forms.MonthCalendar;

namespace WindowRBD1.FormsEdit.Meaning

{

public partial class EditIntermediateResult1 : Window

{

public EditIntermediateResult1()

{

InitializeComponent();

Picket1();

Number1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "UPDATE Proekt.[Промежуточный результат 1] SET [Номер Пикета] = @txtNumberPicket,[Индекс Пикета] = @PicketIndex,[Значения измерения(ЭДС)] = @Meaning where [Номер Промежуточного результата 1] = @NumberTrans1 ";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@NumberTrans1", NumberTrans1.SelectedItem);

cmd.Parameters.AddWithValue("@PicketIndex", PicketIndex.Text);

cmd.Parameters.AddWithValue("@txtNumberPicket", txtNumberPicket.SelectedItem);

cmd.Parameters.AddWithValue("@Meaning", Meaning.Text);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

conn.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void Picket1() //Собирает [Номер пикета] которые существуют в сущности Proekt.Пикет на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер пикета] FROM Proekt.Пикет";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberPicket.Items.Clear();

while (reader.Read())

{

txtNumberPicket.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void Number1() //Собирает [Номер Промежуточного результата 1] которые существуют в сущности Proekt.Промежуточный результат 1 на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Промежуточного результата 1] FROM Proekt.[Промежуточный результат 1]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

NumberTrans1.Items.Clear();

while (reader.Read())

{

NumberTrans1.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.Пикет";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Пикет");

dataGridView1.ItemsSource = ds.Tables["Proekt.Пикет"].DefaultView;

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreatePicket tv = new CreatePicket();

tv.ShowDialog();

}

private void btPoisk2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.[Промежуточный результат 1]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.[Промежуточный результат 1]");

dataGridView1.ItemsSource = ds.Tables["Proekt.[Промежуточный результат 1]"].DefaultView;

}

}

private void btAdd2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateIntermediateResult1 tv = new CreateIntermediateResult1();

tv.ShowDialog();

}

private void NumberTrans1\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

string str = "Select \* from Proekt.[Промежуточный результат 1] where [Номер Промежуточного результата 1] = " + NumberTrans1.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

DataTable dt = new DataTable();

da.Fill(dt);

txtNumberPicket.Text = dt.Rows[0][1].ToString();

PicketIndex.Text = dt.Rows[0][2].ToString();

Meaning.Text = dt.Rows[0][3].ToString();

da.Dispose();

conn.Close();

}

}

}

}

using DocumentFormat.OpenXml.Presentation;

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Meaning;

using WindowRBD1.FormsCreate.Meaning.intermediateResult;

using static System.Windows.Forms.MonthCalendar;

namespace WindowRBD1.FormsEdit.Meaning

{

public partial class EditIntermediateResult2 : Window

{

public EditIntermediateResult2()

{

InitializeComponent();

Picket1();

Number1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "UPDATE Proekt.[Промежуточный результат 2] SET [Номер Пикета] = @txtNumberPicket,[Индекс Пикета] = @PicketIndex,[Значения измерения(ЭДС)] = @Meaning where [Номер Промежуточного результата 2] = @NumberTrans1 ";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@NumberTrans1", NumberTrans1.SelectedItem);

cmd.Parameters.AddWithValue("@PicketIndex", PicketIndex.Text);

cmd.Parameters.AddWithValue("@txtNumberPicket", txtNumberPicket.SelectedItem);

cmd.Parameters.AddWithValue("@Meaning", Meaning.Text);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

conn.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void Picket1() //Собирает [Номер пикета] которые существуют в сущности Proekt.Пикет на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер пикета] FROM Proekt.Пикет";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberPicket.Items.Clear();

while (reader.Read())

{

txtNumberPicket.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void Number1() //Собирает [Номер Промежуточного результата 1] которые существуют в сущности Proekt.Промежуточный результат 1 на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Промежуточного результата 2] FROM Proekt.[Промежуточный результат 2]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

NumberTrans1.Items.Clear();

while (reader.Read())

{

NumberTrans1.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.Пикет";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Пикет");

dataGridView1.ItemsSource = ds.Tables["Proekt.Пикет"].DefaultView;

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreatePicket tv = new CreatePicket();

tv.ShowDialog();

}

private void btPoisk2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.[Промежуточный результат 2]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.[Промежуточный результат 2]");

dataGridView1.ItemsSource = ds.Tables["Proekt.[Промежуточный результат 2]"].DefaultView;

}

}

private void btAdd2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateIntermediateResult2 tv = new CreateIntermediateResult2();

tv.ShowDialog();

}

private void NumberTrans1\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

string str = "Select \* from Proekt.[Промежуточный результат 2] where [Номер Промежуточного результата 2] = " + NumberTrans1.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

DataTable dt = new DataTable();

da.Fill(dt);

txtNumberPicket.Text = dt.Rows[0][1].ToString();

PicketIndex.Text = dt.Rows[0][2].ToString();

Meaning.Text = dt.Rows[0][3].ToString();

da.Dispose();

conn.Close();

}

}

}

}

using DocumentFormat.OpenXml.Presentation;

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Forms;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.Forms;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Meaning;

using WindowRBD1.FormsCreate.Meaning.intermediateResult;

using static System.Windows.Forms.MonthCalendar;

namespace WindowRBD1.FormsEdit.Meaning

{

public partial class EditIntermediateResult3 : Window

{

public EditIntermediateResult3()

{

InitializeComponent();

Picket1();

Number1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для отправки введённых данных на Sql Server

{

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

string sql = "UPDATE Proekt.[Промежуточный результат 3] SET [Номер Пикета] = @txtNumberPicket,[Индекс Пикета] = @PicketIndex,[Значения измерения(ЭДС)] = @Meaning where [Номер Промежуточного результата 3] = @NumberTrans1 ";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@NumberTrans1", NumberTrans1.SelectedItem);

cmd.Parameters.AddWithValue("@PicketIndex", PicketIndex.Text);

cmd.Parameters.AddWithValue("@txtNumberPicket", txtNumberPicket.SelectedItem);

cmd.Parameters.AddWithValue("@Meaning", Meaning.Text);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

conn.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void Picket1() //Собирает [Номер пикета] которые существуют в сущности Proekt.Пикет на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер пикета] FROM Proekt.Пикет";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumberPicket.Items.Clear();

while (reader.Read())

{

txtNumberPicket.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void Number1() //Собирает [Номер Промежуточного результата 1] которые существуют в сущности Proekt.Промежуточный результат 1 на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Промежуточного результата 3] FROM Proekt.[Промежуточный результат 3]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

NumberTrans1.Items.Clear();

while (reader.Read())

{

NumberTrans1.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.Пикет";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Пикет");

dataGridView1.ItemsSource = ds.Tables["Proekt.Пикет"].DefaultView;

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreatePicket tv = new CreatePicket();

tv.ShowDialog();

}

private void btPoisk2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

string sql = "select \* from Proekt.[Промежуточный результат 3]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.[Промежуточный результат 3]");

dataGridView1.ItemsSource = ds.Tables["Proekt.[Промежуточный результат 3]"].DefaultView;

}

}

private void btAdd2\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateIntermediateResult3 tv = new CreateIntermediateResult3();

tv.ShowDialog();

}

private void NumberTrans1\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

string str = "Select \* from Proekt.[Промежуточный результат 3] where [Номер Промежуточного результата 3] = " + NumberTrans1.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

DataTable dt = new DataTable();

da.Fill(dt);

txtNumberPicket.Text = dt.Rows[0][1].ToString();

PicketIndex.Text = dt.Rows[0][2].ToString();

Meaning.Text = dt.Rows[0][3].ToString();

da.Dispose();

conn.Close();

}

}

}

}

using DocumentFormat.OpenXml.Wordprocessing;

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.Xml.Linq;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Works;

namespace WindowRBD1.FormsEdit.Works

{

public partial class EditChief : Window

{

public EditChief()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

txtName1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void txtName1() //Собирает [Начальник отряда] которые существуют в сущности Proekt.[Начальник отряда] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Начальника] FROM Proekt.[Начальник отряда]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumber.Items.Clear();

while (reader.Read())

{

txtNumber.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для изменения введённых данных на Sql Server

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("UPDATE Proekt.[Начальник отряда] SET [ФИО] = @Fio,[Квалификация] = @Competence, [Опыт работы в коллективе] = @Experience, [Общий опыт работы по специальности] = @Experienced,[Дата прохождения медосмотра]=@dateMedical where [Номер Начальника] = @txtNumber", con);

cmd.Parameters.AddWithValue("@txtNumber", txtNumber.SelectedItem);

cmd.Parameters.AddWithValue("@Fio", txtFIO.Text);

cmd.Parameters.AddWithValue("@Competence", txtCompetence.Text);

cmd.Parameters.AddWithValue("@Experience", txtExperience.Text);

cmd.Parameters.AddWithValue("@Experienced", txtExperienced.Text);

cmd.Parameters.AddWithValue("@dateMedical", dateMedical.SelectedDate);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

con.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void txtFIO\_SelectionChanged\_1(object sender, SelectionChangedEventArgs e) //При изменении значения происходит смена данных у всех элементов

{

string str = "Select \* from Proekt.[Начальник отряда] where [Номер Начальника] = " + txtNumber.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtFIO.Text = dr.Rows[0][1].ToString();

txtCompetence.Text = dr.Rows[0][2].ToString();

txtExperience.Text = dr.Rows[0][3].ToString();

txtExperienced.Text = dr.Rows[0][4].ToString();

dateMedical.Text = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateChief tv = new CreateChief();

tv.ShowDialog();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select \* from Proekt.[Начальник отряда]";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.[Начальник отряда]");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.[Начальник отряда]"].DefaultView;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate.Works;

namespace WindowRBD1.FormsEdit.Works

{

public partial class EditDriver : Window

{

public EditDriver()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

txtName1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void txtName1() //Собирает [Номер Водителя] которые существуют в сущности Proekt.[Водители] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Водителя] FROM Proekt.[Водители]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumber.Items.Clear();

while (reader.Read())

{

txtNumber.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для изменения введённых данных на Sql Server

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("UPDATE Proekt.[Водители] SET ФИО = @txtFIO,[Квалификация] = @Competence, [Опыт работы в коллективе] = @Experience, [Общий опыт работы по специальности] = @Experienced,[Дата прохождения медосмотра]=@dateMedical where [Номер Водителя] = @txtNumber", con);

cmd.Parameters.AddWithValue("@txtNumber", txtNumber.SelectedItem);

cmd.Parameters.AddWithValue("@txtFIO", txtFIO.Text);

cmd.Parameters.AddWithValue("@Competence", txtCompetence.Text);

cmd.Parameters.AddWithValue("@Experience", txtExperience.Text);

cmd.Parameters.AddWithValue("@Experienced", txtExperienced.Text);

cmd.Parameters.AddWithValue("@dateMedical", dateMedical.SelectedDate); ;

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

con.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void txtFIO\_SelectionChanged(object sender, SelectionChangedEventArgs e) //При изменении значения происходит смена данных у всех элементов

{

string str = "Select \* from Proekt.[Водители] where [Номер Водителя] = " + txtNumber.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtFIO.Text = dr.Rows[0][1].ToString();

txtCompetence.Text = dr.Rows[0][2].ToString();

txtExperience.Text = dr.Rows[0][3].ToString();

txtExperienced.Text = dr.Rows[0][4].ToString();

dateMedical.Text = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateDriver tv = new CreateDriver();

tv.ShowDialog();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select \* from Proekt.Водители";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Водители");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Водители"].DefaultView;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate.Works;

namespace WindowRBD1.FormsEdit.Works

{

public partial class EditITR : Window

{

public EditITR()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

txtName1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void txtName1() //Собирает [Номер ИТР] которые существуют в сущности Proekt.[ИТР] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер ИТР] FROM Proekt.[ИТР]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumber.Items.Clear();

while (reader.Read())

{

txtNumber.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для изменения введённых данных на Sql Server

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("UPDATE Proekt.[ИТР] SET ФИО = @txtFIO,[Квалификация] = @Competence, [Опыт работы в коллективе] = @Experience, [Общий опыт работы по специальности] = @Experienced,[Дата прохождения медосмотра]=@date where [Номер ИТР] = @txtNumber", con);

cmd.Parameters.AddWithValue("@txtNumber", txtNumber.SelectedItem);

cmd.Parameters.AddWithValue("@txtFIO", txtFIO.Text);

cmd.Parameters.AddWithValue("@Competence", txtCompetence.Text);

cmd.Parameters.AddWithValue("@Experience", txtExperience.Text);

cmd.Parameters.AddWithValue("@Experienced", txtExperienced.Text);

cmd.Parameters.AddWithValue("@date", dateMedical.SelectedDate);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

con.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e)//закрытие формы

{

this.Close();

}

private void txtFIO\_SelectionChanged(object sender, SelectionChangedEventArgs e) //При изменении значения происходит смена данных у всех элементов

{

string str = "Select \* from Proekt.ИТР where [Номер ИТР] = " + txtNumber.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtFIO.Text = dr.Rows[0][1].ToString();

txtCompetence.Text = dr.Rows[0][2].ToString();

txtExperience.Text = dr.Rows[0][3].ToString();

txtExperienced.Text = dr.Rows[0][4].ToString();

dateMedical.Text = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateITR tv = new CreateITR();

tv.ShowDialog();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select \* from Proekt.ИТР";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.ИТР");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.ИТР"].DefaultView;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate.Works;

namespace WindowRBD1.FormsEdit.Works

{

public partial class EditSupervisor : Window

{

public EditSupervisor()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

txtName1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void txtName1() //Собирает [Номер Супервайзера] которые существуют в сущности Proekt.[Супервайзер] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Супервайзера] FROM Proekt.[Супервайзер]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumber.Items.Clear();

while (reader.Read())

{

txtNumber.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btEdit\_Click(object sender, RoutedEventArgs e)//Подключение к базе дынных для изменения введённых данных на Sql Server

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("UPDATE Proekt.[Супервайзер] SET ФИО = @txtFIO,[Квалификация] = @Competence, [Опыт работы в коллективе] = @Experience, [Общий опыт работы по специальности] = @Experienced where [Номер Супервайзера] = @txtNumber", con);

cmd.Parameters.AddWithValue("@txtNumber", txtNumber.SelectedItem);

cmd.Parameters.AddWithValue("@txtFIO", txtFIO.Text);

cmd.Parameters.AddWithValue("@Competence", txtCompetence.Text);

cmd.Parameters.AddWithValue("@Experience", txtExperience.Text);

cmd.Parameters.AddWithValue("@Experienced", txtExperienced.Text);

//cmd.Parameters.AddWithValue("@NumberTelemetryEquipment", cmbNumberTelemetryEquipment2);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

con.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void txtFIO\_SelectionChanged(object sender, SelectionChangedEventArgs e) //При изменении значения происходит смена данных у всех элементов

{

string str = "Select \* from Proekt.[Супервайзер] where [Номер Супервайзера] = " + txtNumber.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtFIO.Text = dr.Rows[0][1].ToString();

txtCompetence.Text = dr.Rows[0][2].ToString();

txtExperience.Text = dr.Rows[0][3].ToString();

txtExperienced.Text = dr.Rows[0][4].ToString();

dateMedical.Text = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateSupervisor tv = new CreateSupervisor();

tv.ShowDialog();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select \* from Proekt.Супервайзер";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Супервайзер");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Супервайзер"].DefaultView;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate.Works;

namespace WindowRBD1.FormsEdit.Works

{

public partial class EditWorker : Window

{

public EditWorker()

{

InitializeComponent();

//Создание часов для добавления записи [Дата и время появления записи]

var timer = new System.Windows.Threading.DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.IsEnabled = true;

timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString(); };

timer.Start();

//

txtName1();

}

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

private void txtName1() //Собирает [Номер Рабочего] которые существуют в сущности Proekt.[Рабочие] на Sql Server

{

SqlConnection connection = new SqlConnection(BdCon.Con);

string sql = "SELECT [Номер Рабочего] FROM Proekt.[Рабочие]";

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

txtNumber.Items.Clear();

while (reader.Read())

{

txtNumber.Items.Add(reader[0].ToString());

}

reader.Close();

command.Dispose();

connection.Close();

}

private void btEdit\_Click(object sender, RoutedEventArgs e) //Подключение к базе дынных для изменения введённых данных на Sql Server

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("UPDATE Proekt.[Рабочие] SET ФИО = @txtFIO, [Квалификация] = @Competence, [Опыт работы в коллективе] = @Experience, [Общий опыт работы по специальности] = @Experienced where [Номер Рабочего] = @txtNumber", con);

cmd.Parameters.AddWithValue("@txtNumber", txtNumber.SelectedItem);

cmd.Parameters.AddWithValue("@txtFIO", txtFIO.Text);

cmd.Parameters.AddWithValue("@Competence", txtCompetence.Text);

cmd.Parameters.AddWithValue("@Experience", txtExperience.Text);

cmd.Parameters.AddWithValue("@Experienced", txtExperienced.Text);

//cmd.Parameters.AddWithValue("@NumberTelemetryEquipment", cmbNumberTelemetryEquipment2);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись изменена");

con.Close();

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) //закрытие формы

{

this.Close();

}

private void txtFIO\_SelectionChanged(object sender, SelectionChangedEventArgs e) //При изменении значения происходит смена данных у всех элементов

{

string str = "Select \* from Proekt.Рабочие where [Номер Рабочего] = " + txtNumber.SelectedItem;

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtFIO.Text = dr.Rows[0][1].ToString();

txtCompetence.Text = dr.Rows[0][2].ToString();

txtExperience.Text = dr.Rows[0][3].ToString();

txtExperienced.Text = dr.Rows[0][4].ToString();

dateMedical.Text = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btAdd1\_Click(object sender, RoutedEventArgs e) //Просмотр данных

{

CreateWorker tv = new CreateWorker();

tv.ShowDialog();

}

private void btPoisk1\_Click(object sender, RoutedEventArgs e) //Просмотр данных в datagidview

{

string sql = "select \* from Proekt.Рабочие";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Рабочие");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Рабочие"].DefaultView;

}

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

using static System.Windows.Forms.MonthCalendar;

namespace WindowRBD1.FormsMain

{

public partial class Area : System.Windows.Window

{

public Area()

{

InitializeComponent();

AreaCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer form1 = new Cursomer();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project form1 = new Project();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract form1 = new Contract();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, cmbTab.Text);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

} // end using

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void OrderCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Order form1 = new Order();

form1.Show();

}

private void EngineerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Engineer form1 = new Engineer();

form1.Show();

}

private void ChiefCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Chief form1 = new Chief();

form1.Show();

}

private void WorkerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Worker form1 = new Worker();

form1.Show();

}

private void DriverCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Driver form1 = new Driver();

form1.Show();

}

private void SupervisorCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Supervisor form1 = new Supervisor();

form1.Show();

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring form1 = new Measuring();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative form1 = new Generative();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry form1 = new Telemetry();

form1.Show();

}

private void MethodologyCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Methodology form1 = new Methodology();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket form1 = new Picket();

form1.Show();

}

private void AreaCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.Площади";

uri = 5;

cmbTab.ItemsSource = new string[] { "[Номер площади]", "[Наименование площади]" };

RoleBox = "Proekt.Площади";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Площади");

dataGridView1.ItemsSource = ds.Tables["Proekt.Площади"].DefaultView;

} // end using

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile form1 = new Profile();

form1.Show();

}

private void EquipmentCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Equipment form1 = new Equipment();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateArea form1 = new CreateArea();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditArea form1 = new EditArea();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, RoleBox);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

} // end using

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, RoleBox);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

} // end using

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.Площади where [Номер площади] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

DataTable dt = new DataTable();

da.Fill(dt);

txtNumberArea.Content = dt.Rows[0][0].ToString();

txtArea.Content = dt.Rows[0][1].ToString();

txtProfile.Content = dt.Rows[0][2].ToString();

txtList.Content = dt.Rows[0][3].ToString();

txtPerimeterLength.Content = dt.Rows[0][4].ToString();

txtSizeArea.Content = dt.Rows[0][5].ToString();

dateBeginning.Content = dt.Rows[0][6].ToString();

dateEnd.Content = dt.Rows[0][7].ToString();

DateCreate.Content = dt.Rows[0][8].ToString();

DateEdit.Content = dt.Rows[0][9].ToString();

txtSupervisorData.Content = dt.Rows[0][10].ToString();

txtSupervisorOrder.Content = dt.Rows[0][11].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportArea.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.Площади");

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

{

sw.WriteLine("[Номер площади]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Наименование площади]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Номер списка углов периметра]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Длина периметра]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Величина площади окружённая периметром]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Дата и время окончания работ]:" + dt.Rows[i][5].ToString());

sw.WriteLine("[Дата и время появления записи]:" + dt.Rows[i][6].ToString());

sw.WriteLine("[Дата и время изменения записи]:" + dt.Rows[i][7].ToString());

sw.WriteLine("[Супервайзер полевых работ]:" + dt.Rows[i][8].ToString());

sw.WriteLine("[Супервайзер обработки данных]:" + dt.Rows[i][9].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe",path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

namespace WindowRBD1.FormsMain

{

public partial class Chief : System.Windows.Window

{

public Chief()

{

InitializeComponent();

ChiefCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer form1 = new Cursomer();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project form1 = new Project();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract form1 = new Contract();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, cmbTab.Text);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

} // end using

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void OrderCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Order form1 = new Order();

form1.Show();

}

private void EngineerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Engineer form1 = new Engineer();

form1.Show();

}

private void ChiefCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.[Начальник отряда]";

uri = 9;

cmbTab.ItemsSource = new string[] { "[Номер Начальника]", "ФИО" };

RoleBox = "Proekt.[Начальник отряда]";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.[Начальник отряда]");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.[Начальник отряда]"].DefaultView;

} // end using

}

private void WorkerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Worker form1 = new Worker();

form1.Show();

}

private void DriverCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Driver form1 = new Driver();

form1.Show();

}

private void SupervisorCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Supervisor form1 = new Supervisor();

form1.Show();

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring form1 = new Measuring();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative form1 = new Generative();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry form1 = new Telemetry();

form1.Show();

}

private void MethodologyCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Methodology form1 = new Methodology();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket form1 = new Picket();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area form1 = new Area();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile form1 = new Profile();

form1.Show();

}

private void EquipmentCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Equipment form1 = new Equipment();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateChief form1 = new CreateChief();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditChief form1 = new EditChief();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, RoleBox);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, RoleBox);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

} // end using

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.[Начальник отряда] where [Номер Начальника] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtNumberChief.Content = dr.Rows[0][0].ToString();

txtFIO.Content = dr.Rows[0][1].ToString();

txtCompetence.Content = dr.Rows[0][2].ToString();

txtExperience.Content = dr.Rows[0][3].ToString();

txtExperienced.Content = dr.Rows[0][4].ToString();

dateMedical.Content = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportChief.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.[Начальник отряда]");

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

{

sw.WriteLine("[Номер Начальника]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[ФИО]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Квалификация]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Опыт работы в коллективе]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Общий опыт работы по специальности]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Дата прохождения медосмотра]:" + dt.Rows[i][5].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

namespace WindowRBD1.FormsMain

{

public partial class Contract : System.Windows.Window

{

public Contract()

{

InitializeComponent();

ContractCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer form1 = new Cursomer();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project form1 = new Project();

form1.Show();

}

private void ContractCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.Договор";

uri = 3;

cmbTab.ItemsSource = new string[] { "[Номер договора]", "[Наименование договора]" };

RoleBox = "Proekt.Договор";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Договор");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Договор"].DefaultView;

} // end using

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, cmbTab.Text);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

} // end using

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void OrderCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Order form1 = new Order();

form1.Show();

}

private void EngineerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Engineer form1 = new Engineer();

form1.Show();

}

private void ChiefCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Chief form1 = new Chief();

form1.Show();

}

private void WorkerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Worker form1 = new Worker();

form1.Show();

}

private void DriverCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Driver form1 = new Driver();

form1.Show();

}

private void SupervisorCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Supervisor form1 = new Supervisor();

form1.Show();

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring form1 = new Measuring();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative form1 = new Generative();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry form1 = new Telemetry();

form1.Show();

}

private void MethodologyCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Methodology form1 = new Methodology();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket form1 = new Picket();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area form1 = new Area();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile form1 = new Profile();

form1.Show();

}

private void EquipmentCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Equipment form1 = new Equipment();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateContract form1 = new CreateContract();

form1.Show();

//ds.Clear();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditContract form1 = new EditContract();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, RoleBox);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

} // end using

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, RoleBox);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

} // end using

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.Договор where [Номер договора] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

DataTable dt = new DataTable();

da.Fill(dt);

txtNumberContract.Content = dt.Rows[0][0].ToString();

txtNameContract.Content = dt.Rows[0][1].ToString();

txtBeginnings.Content = dt.Rows[0][2].ToString();

txtEndings.Content = dt.Rows[0][3].ToString();

txtCost.Content = dt.Rows[0][4].ToString();

DateCreate.Content = dt.Rows[0][5].ToString();

DateEdit.Content = dt.Rows[0][6].ToString();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportContract.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.Договор");

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

{

sw.WriteLine("[Номер договора]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Наименование договора]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Начала]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Окончания]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Стоимость]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Дата и время появления записи]:" + dt.Rows[i][5].ToString());

sw.WriteLine("[Дата и время изменения записи]:" + dt.Rows[i][5].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

using DocumentFormat.OpenXml.Drawing.Charts;

namespace WindowRBD1.FormsMain

{

public partial class Cursomer : System.Windows.Window

{

public Cursomer()

{

InitializeComponent();

CursomerCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.Заказчик";

cmbTab.ItemsSource = new string[] { "[Номер заказчика]", "[Название компаний]", "[Представитель]" };

RoleBox = "Proekt.Заказчик";

uri = 1;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Заказчик");

dataGridView1.ItemsSource = ds.Tables["Proekt.Заказчик"].DefaultView;

}

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project form1 = new Project();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract form1 = new Contract();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, cmbTab.Text);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

} // end using

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void OrderCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Order form1 = new Order();

form1.Show();

}

private void EngineerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Engineer form1 = new Engineer();

form1.Show();

}

private void ChiefCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Chief form1 = new Chief();

form1.Show();

}

private void WorkerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Worker form1 = new Worker();

form1.Show();

}

private void DriverCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Driver form1 = new Driver();

form1.Show();

}

private void SupervisorCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Supervisor form1 = new Supervisor();

form1.Show();

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring form1 = new Measuring();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative form1 = new Generative();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry form1 = new Telemetry();

form1.Show();

}

private void MethodologyCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Methodology form1 = new Methodology();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket form1 = new Picket();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area form1 = new Area();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile form1 = new Profile();

form1.Show();

}

private void EquipmentCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Equipment form1 = new Equipment();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateClient form1 = new CreateClient();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditClient form1 = new EditClient();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + " = '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, RoleBox);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

} // end using

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, RoleBox);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

} // end using

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e)//Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView1 = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.Заказчик where [Номер заказчика] = " + rowView1[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

DataTable dt = new DataTable();

da.Fill(dt);

NumberClient.Content = dt.Rows[0][0].ToString();

txtNameCompany.Content = dt.Rows[0][1].ToString();

txtLegalAddress.Text = dt.Rows[0][2].ToString();

txtActualAddress.Text = dt.Rows[0][3].ToString();

txtINN.Content = dt.Rows[0][4].ToString();

txtPda.Content = dt.Rows[0][5].ToString();

txtCalculated.Content = dt.Rows[0][6].ToString();

txtCorrespondent.Content = dt.Rows[0][7].ToString();

txtAgent.Content = dt.Rows[0][8].ToString();

txtPhone.Content = dt.Rows[0][9].ToString();

Email.Text = dt.Rows[0][10].ToString();

Site.Text = dt.Rows[0][11].ToString();

DateCreate.Content = dt.Rows[0][12].ToString();

DateEdit.Content = dt.Rows[0][13].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportCursomer.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.[Заказчик]");

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

{

sw.WriteLine("[Номер заказчика]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Название компаний]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Юридический адрес]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Фактический адрес]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[ИНН]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[КПК]:" + dt.Rows[i][5].ToString());

sw.WriteLine("[Расчетный счёт]:" + dt.Rows[i][6].ToString());

sw.WriteLine("[Корреспондентский счёт]:" + dt.Rows[i][7].ToString());

sw.WriteLine("[Представитель]:" + dt.Rows[i][8].ToString());

sw.WriteLine("[Телефон представителя]:" + dt.Rows[i][9].ToString());

sw.WriteLine("[Адрес электронной почты]:" + dt.Rows[i][10].ToString());

sw.WriteLine("[Сайт]:" + dt.Rows[i][11].ToString());

sw.WriteLine("[Дата и время появления записи]:" + dt.Rows[i][12].ToString());

sw.WriteLine("[Дата и время изменения записи]:" + dt.Rows[i][13].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Data;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Windows;

using System.Windows.Controls;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

using DocumentFormat.OpenXml;

using DocumentFormat.OpenXml.Packaging;

using DocumentFormat.OpenXml.Wordprocessing;

using System.Windows.Data;

using System.Reflection;

using System.Collections.Generic;

using word = Microsoft.Office.Interop.Word;

using Magnum.Collections;

using Org.BouncyCastle.Asn1.Ocsp;

using System.ComponentModel.Composition.Primitives;

using System.Collections.ObjectModel;

using application = Microsoft.Office.Interop.Word;

using Microsoft.Office.Interop.Word;

using System.Windows.Forms;

using \_Application = Microsoft.Office.Interop.Word.\_Application;

using DocumentFormat.OpenXml.ExtendedProperties;

using Microsoft.Win32;

using System.Reflection.Emit;

namespace WindowRBD1.FormsMain

{

public partial class Driver : System.Windows.Window

{

public Driver()

{

InitializeComponent();

DriverCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer form1 = new Cursomer();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project form1 = new Project();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract form1 = new Contract();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, cmbTab.Text);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

} // end using

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void OrderCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Order form1 = new Order();

form1.Show();

}

private void EngineerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Engineer form1 = new Engineer();

form1.Show();

}

private void ChiefCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Chief form1 = new Chief();

form1.Show();

}

private void WorkerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Worker form1 = new Worker();

form1.Show();

}

private void DriverCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.Водители";

uri = 11;

cmbTab.ItemsSource = new string[] { "[Номер Водителя]", "ФИО" };

RoleBox = "Proekt.Водители";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Водители");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Водители"].DefaultView;

} // end using

}

private void SupervisorCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Supervisor form1 = new Supervisor();

form1.Show();

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring form1 = new Measuring();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative form1 = new Generative();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry form1 = new Telemetry();

form1.Show();

}

private void MethodologyCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Methodology form1 = new Methodology();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket form1 = new Picket();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area form1 = new Area();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile form1 = new Profile();

form1.Show();

}

private void EquipmentCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Equipment form1 = new Equipment();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateDriver form1 = new CreateDriver();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditDriver form1 = new EditDriver();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, RoleBox);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

} // end using

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, RoleBox);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

} // end using

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.Водители where [Номер Водителя] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtNumberDriver.Content = dr.Rows[0][0].ToString();

txtFIO.Content = dr.Rows[0][1].ToString();

txtCompetence.Content = dr.Rows[0][2].ToString();

txtExperience.Content = dr.Rows[0][3].ToString();

txtExperienced.Content = dr.Rows[0][4].ToString();

dateMedical.Content = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportDriver.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.[Водители]");

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

{

sw.WriteLine("[Номер Водителя]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[ФИО]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Квалификация]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Опыт работы в коллективе]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Общий опыт работы по специальности]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Дата прохождения медосмотра]:" + dt.Rows[i][5].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

namespace WindowRBD1.FormsMain

{

public partial class Engineer: System.Windows.Window

{

public Engineer()

{

InitializeComponent();

EngineerCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer form1 = new Cursomer();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project form1 = new Project();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract form1 = new Contract();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, cmbTab.Text);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

} // end using

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void OrderCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Order form1 = new Order();

form1.Show();

}

private void EngineerCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.ИТР";

uri = 8;

cmbTab.ItemsSource = new string[] { "[Номер ИТР]", "[Наименование ИТР]", "ФИО" };

RoleBox = "Proekt.ИТР";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.ИТР");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.ИТР"].DefaultView;

}

}

private void ChiefCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Chief form1 = new Chief();

form1.Show();

}

private void WorkerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Worker form1 = new Worker();

form1.Show();

}

private void DriverCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Driver form1 = new Driver();

form1.Show();

}

private void SupervisorCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Supervisor form1 = new Supervisor();

form1.Show();

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring form1 = new Measuring();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative form1 = new Generative();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry form1 = new Telemetry();

form1.Show();

}

private void MethodologyCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Methodology form1 = new Methodology();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket form1 = new Picket();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area form1 = new Area();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile form1 = new Profile();

form1.Show();

}

private void EquipmentCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Equipment form1 = new Equipment();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateITR form1 = new CreateITR();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditITR form1 = new EditITR();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, RoleBox);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

} // end using

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, RoleBox);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

} // end using

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.[ИТР] where [Номер ИТР] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtNumberEngineer.Content = dr.Rows[0][0].ToString();

txtFIO.Content = dr.Rows[0][1].ToString();

txtCompetence.Content = dr.Rows[0][2].ToString();

txtExperience.Content = dr.Rows[0][3].ToString();

txtExperienced.Content = dr.Rows[0][4].ToString();

dateMedical.Content = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportEngineer.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.[ИТР]");

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

{

sw.WriteLine("[Номер ИТР]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[ФИО]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Квалификация]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Опыт работы в коллективе]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Общий опыт работы по специальности]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Дата прохождения медосмотра]:" + dt.Rows[i][5].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

namespace WindowRBD1.FormsMain

{

public partial class Equipment : System.Windows.Window

{

public Equipment()

{

InitializeComponent();

EquipmentCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer form1 = new Cursomer();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project form1 = new Project();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract form1 = new Contract();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, cmbTab.Text);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

} // end using

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void OrderCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Order form1 = new Order();

form1.Show();

}

private void EngineerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Engineer form1 = new Engineer();

form1.Show();

}

private void ChiefCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Chief form1 = new Chief();

form1.Show();

}

private void WorkerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Worker form1 = new Worker();

form1.Show();

}

private void DriverCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Driver form1 = new Driver();

form1.Show();

}

private void SupervisorCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Supervisor form1 = new Supervisor();

form1.Show();

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring form1 = new Measuring();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative form1 = new Generative();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry form1 = new Telemetry();

form1.Show();

}

private void MethodologyCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Methodology form1 = new Methodology();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket form1 = new Picket();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area form1 = new Area();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile form1 = new Profile();

form1.Show();

}

private void EquipmentCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.Оборудования";

uri = 17;

cmbTab.ItemsSource = new string[] { "Номер Оборудования" };

RoleBox = "Proekt.Оборудования";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Оборудования");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Оборудования"].DefaultView;

}

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateEquipment form1 = new CreateEquipment();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditEquipment form1 = new EditEquipment();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, RoleBox);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

} // end using

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, RoleBox);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

} // end using

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e)//Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.Оборудования where [Номер Оборудования] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

NumberEquipment.Content = dr.Rows[0][0].ToString();

NumberGenEquipment.Content = dr.Rows[0][1].ToString();

NumberMesuringEquipment.Content = dr.Rows[0][2].ToString();

NumberTelemetryEquipment.Content = dr.Rows[0][3].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportEquipment.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.Оборудования");

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

{

sw.WriteLine("[Номер Оборудования]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Номер генеративного оборудования]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Номер измерительного оборудования]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Номер телеметрического оборудования]:" + dt.Rows[i][3].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe",path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

namespace WindowRBD1.FormsMain

{

public partial class Generative : System.Windows.Window

{

public Generative()

{

InitializeComponent();

GenerativeCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer form1 = new Cursomer();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project form1 = new Project();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract form1 = new Contract();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, cmbTab.Text);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

} // end using

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void OrderCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Order form1 = new Order();

form1.Show();

}

private void EngineerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Engineer form1 = new Engineer();

form1.Show();

}

private void ChiefCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Chief form1 = new Chief();

form1.Show();

}

private void WorkerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Worker form1 = new Worker();

form1.Show();

}

private void DriverCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Driver form1 = new Driver();

form1.Show();

}

private void SupervisorCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Supervisor form1 = new Supervisor();

form1.Show();

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring form1 = new Measuring();

form1.Show();

}

private void GenerativeCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.[Генеративное оборудование]";

uri = 14;

cmbTab.ItemsSource = new string[] { "[Номер генеративного оборудования]", "[Наименование]" };

RoleBox = "Proekt.[Генеративное оборудование]";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.[Генеративное оборудование]");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.[Генеративное оборудование]"].DefaultView;

}

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry form1 = new Telemetry();

form1.Show();

}

private void MethodologyCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Methodology form1 = new Methodology();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket form1 = new Picket();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area form1 = new Area();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile form1 = new Profile();

form1.Show();

}

private void EquipmentCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Equipment form1 = new Equipment();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateGenerative form1 = new CreateGenerative();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditGenerative form1 = new EditGenerative();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.[Генеративное оборудование] where [Номер генеративного оборудования] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtNumber.Content = dr.Rows[0][0].ToString();

txtName.Content = dr.Rows[0][1].ToString();

txtInventory.Content = dr.Rows[0][2].ToString();

dateVerification.Content = dr.Rows[0][3].ToString();

datePurchases.Content = dr.Rows[0][4].ToString();

txtCharacteristic.Text = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportGenerative.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.[Генеративное оборудование]");

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

{

sw.WriteLine("[Номер генеративного оборудования]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Наименование]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Инвентарный номер]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Дата приобретения]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Дата поверки]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Характеристики]:" + dt.Rows[i][5].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe",path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

using static System.Windows.Forms.MonthCalendar;

using WindowRBD1.FormsCreate.Meaning.intermediateResult;

using WindowRBD1.FormsCreate.Meaning;

using WindowRBD1.FormsEdit.Meaning;

using WindowRBD1.Forms;

using static ClosedXML.Excel.XLPredefinedFormat;

namespace WindowRBD1.FormsMain

{

public partial class Meaning : System.Windows.Window

{

public Meaning()

{

InitializeComponent();

}

string con = BdCon.Con;

public string RoleBox;

public string NameBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

string tr;

string Number1, NumberPicket1, Index1, Size1;

public object openFileDialog1 { get; private set; }

int uri;

private void Transform1Command\_Click(object sender, RoutedEventArgs e) //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.Трансформанта1";

uri = 18;

Number.Content = "Номер Трансформанты измерения 1";

NumberPicket.Content = "Номер Пикета";

Index.Content = "Индекс Пикета";

Size.Content = "Значения трансформанты (ρτ)";

Number1 = "[Номер Трансформанты измерения 1]";

NumberPicket1 = "[Номер Пикета]";

Index1 = "[Индекс Пикета]";

Size1 = "[Значения трансформанты (ρτ)]";

tr = "Трансформанта1.txt";

cmbTab.ItemsSource = new string[] { "[Номер Трансформанты измерения 1]", "[Номер Пикета]","[Индекс Пикета]" };

RoleBox = "Proekt.Трансформанта1";

NameBox = "[Номер Трансформанты измерения 1]";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Трансформанта1");

dataGridView1.ItemsSource = ds.Tables["Proekt.Трансформанта1"].DefaultView;

}

}

private void Transform2Command\_Click(object sender, RoutedEventArgs e) //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.Трансформанта2";

uri = 19;

Number.Content = "Номер Трансформанты измерения 2";

NumberPicket.Content = "Номер Пикета";

Index.Content = "Индекс Пикета";

Size.Content = "Значения трансформанты (ρτ)";

Number1 = "[Номер Трансформанты измерения 2]";

NumberPicket1 = "[Номер Пикета]";

Index1 = "[Индекс Пикета]";

Size1 = "[Значения трансформанты (ρτ)]";

tr = "Трансформанта2.txt";

cmbTab.ItemsSource = new string[] { "[Номер Трансформанты измерения 2]", "[Номер Пикета]", "[Индекс Пикета]" };

RoleBox = "Proekt.Трансформанта2";

NameBox = "[Номер Трансформанты измерения 2]";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Трансформанта2");

dataGridView1.ItemsSource = ds.Tables["Proekt.Трансформанта2"].DefaultView;

}

}

private void Transform3Command\_Click(object sender, RoutedEventArgs e) //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.Трансформанта3";

uri = 20;

Number.Content = "Номер Трансформанты измерения 3";

NumberPicket.Content = "Номер Пикета";

Index.Content = "Индекс Пикета";

Size.Content = "Значения трансформанты (ρτ)";

Number1 = "[Номер Трансформанты измерения 3]";

NumberPicket1 = "[Номер Пикета]";

Index1 = "[Индекс Пикета]";

Size1 = "[Значения трансформанты (ρτ)]";

tr = "Трансформанта3.txt";

cmbTab.ItemsSource = new string[] { "[Номер Трансформанты измерения 3]", "[Номер Пикета]", "[Индекс Пикета]" };

RoleBox = "Proekt.Трансформанта3";

NameBox = "[Номер Трансформанты измерения 3]";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Трансформанта3");

dataGridView1.ItemsSource = ds.Tables["Proekt.Трансформанта3"].DefaultView;

}

}

private void finalCommand\_Click(object sender, RoutedEventArgs e) //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.[Окончательный результат]";

uri = 21;

Number.Content = "Номер Окончательного результата";

NumberPicket.Content = "Номер Пикета";

Index.Content = "Индекс Пикета";

Size.Content = "Значения измерения(ЭДС)";

Number1 = "[Номер Окончательного результата]";

NumberPicket1 = "[Номер Пикета]";

Index1 = "[Индекс Пикета]";

Size1 = "[Значения измерения(ЭДС)]";

tr = "Окончательный результат.txt";

cmbTab.ItemsSource = new string[] { "[Номер Окончательного результата]", "[Номер Пикета]", "[Индекс Пикета]" };

RoleBox = "Proekt.[Окончательный результат]";

NameBox = "[Номер Окончательного результата]";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.[Окончательный результат]");

dataGridView1.ItemsSource = ds.Tables["Proekt.[Окончательный результат]"].DefaultView;

}

}

private void IntermediateResultCommand\_Click(object sender, RoutedEventArgs e) //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.[Промежуточный результат]";

uri = 22;

Number.Content = "Номер Промежуточного результата";

NumberPicket.Content = "Номер Пикета";

Index.Content = "Индекс Пикета";

Size.Content = "Значения измерения(ЭДС)";

Number1 = "[Номер Промежуточного результата]";

NumberPicket1 = "[Номер Пикета]";

Index1 = "[Индекс Пикета]";

Size1 = "[Значения измерения(ЭДС)]";

tr = "Промежуточный результат.txt";

cmbTab.ItemsSource = new string[] { "[Номер Промежуточного результата]", "[Номер Пикета]", "[Индекс Пикета]" };

RoleBox = "Proekt.[Промежуточный результат]";

NameBox = "[Номер Промежуточного результата]";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.[Промежуточный результат]");

dataGridView1.ItemsSource = ds.Tables["Proekt.[Промежуточный результат]"].DefaultView;

}

}

private void IntermediateResult1Command\_Click(object sender, RoutedEventArgs e) //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.[Промежуточный результат 1]";

uri = 23;

Number.Content = "Номер Промежуточного результата 1";

NumberPicket.Content = "Номер Пикета";

Index.Content = "Индекс Пикета";

Size.Content = "Значения измерения(ЭДС)";

Number1 = "[Номер Промежуточного результата 1]";

NumberPicket1 = "[Номер Пикета]";

Index1 = "[Индекс Пикета]";

Size1 = "[Значения измерения(ЭДС)]";

tr = "Промежуточный результат 1.txt";

cmbTab.ItemsSource = new string[] { "[Номер Промежуточного результата 1]", "[Номер Пикета]", "[Индекс Пикета]" };

RoleBox = "Proekt.[Промежуточный результат 1]";

NameBox = "[Номер Промежуточного результата 1]";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.[Промежуточный результат 1]");

dataGridView1.ItemsSource = ds.Tables["Proekt.[Промежуточный результат 1]"].DefaultView;

}

}

private void IntermediateResult2Command\_Click(object sender, RoutedEventArgs e) //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.[Промежуточный результат 2]";

uri = 24;

Number.Content = "Номер Промежуточного результата 2";

NumberPicket.Content = "Номер Пикета";

Index.Content = "Индекс Пикета";

Size.Content = "Значения измерения(ЭДС)";

Number1 = "[Номер Промежуточного результата 2]";

NumberPicket1 = "[Номер Пикета]";

Index1 = "[Индекс Пикета]";

Size1 = "[Значения измерения(ЭДС)]";

tr = "Промежуточный результат 2.txt";

cmbTab.ItemsSource = new string[] { "[Номер Промежуточного результата 2]", "[Номер Пикета]", "[Индекс Пикета]" };

RoleBox = "Proekt.[Промежуточный результат 2]";

NameBox = "[Номер Промежуточного результата 2]";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.[Промежуточный результат 2]");

dataGridView1.ItemsSource = ds.Tables["Proekt.[Промежуточный результат 2]"].DefaultView;

}

}

private void IntermediateResult3Command\_Click(object sender, RoutedEventArgs e) //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.[Промежуточный результат 3]";

uri = 25;

Number.Content = "Номер Промежуточного результата 3";

NumberPicket.Content = "Номер Пикета";

Index.Content = "Индекс Пикета";

Size.Content = "Значения измерения(ЭДС)";

Number1 = "[Номер Промежуточного результата 3]";

NumberPicket1 = "[Номер Пикета]";

Index1 = "[Индекс Пикета]";

Size1 = "[Значения измерения(ЭДС)]";

tr = "Промежуточный результат 3.txt";

cmbTab.ItemsSource = new string[] { "[Номер Промежуточного результата 3]", "[Номер Пикета]", "[Индекс Пикета]" };

RoleBox = "Proekt.[Промежуточный результат 3]";

NameBox = "[Номер Промежуточного результата 3]";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.[Промежуточный результат 3]");

dataGridView1.ItemsSource = ds.Tables["Proekt.[Промежуточный результат 3]"].DefaultView;

}

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) // Открытие формы

{

if (RoleBox == "[Трансформанта1]")

{

CreateTransformant1 rt = new CreateTransformant1();

rt.ShowDialog();

}

if (RoleBox == "[Трансформанта2]")

{

CreateTransformant2 rt = new CreateTransformant2();

rt.ShowDialog();

}

if (RoleBox == "[Трансформанта3]")

{

CreateTransformant3 rt = new CreateTransformant3();

rt.ShowDialog();

}

if (RoleBox == "[Окончательный результат]")

{

CreatefinalResult rt = new CreatefinalResult();

rt.ShowDialog();

}

if (RoleBox == "[Промежуточный результат]")

{

CreateIntermediateResult rt = new CreateIntermediateResult();

rt.ShowDialog();

}

if (RoleBox == "[Промежуточный результат 1]")

{

CreateIntermediateResult1 rt = new CreateIntermediateResult1();

rt.ShowDialog();

}

if (RoleBox == "[Промежуточный результат 2]")

{

CreateIntermediateResult2 rt = new CreateIntermediateResult2();

rt.ShowDialog();

}

if (RoleBox == "[Промежуточный результат 3]")

{

CreateIntermediateResult3 rt = new CreateIntermediateResult3();

rt.ShowDialog();

}

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) // Открытие формы

{

if (RoleBox == "[Трансформанта1]")

{

EditTransformant1 rt = new EditTransformant1();

rt.ShowDialog();

}

if (RoleBox == "[Трансформанта2]")

{

EditTransformant2 rt = new EditTransformant2();

rt.ShowDialog();

}

if (RoleBox == "[Трансформанта3]")

{

EditTransformant3 rt = new EditTransformant3();

rt.ShowDialog();

}

if (RoleBox == "[Окончательный результат]")

{

EditfinalResult rt = new EditfinalResult();

rt.ShowDialog();

}

if (RoleBox == "[Промежуточный результат]")

{

EditIntermediateResult rt = new EditIntermediateResult();

rt.ShowDialog();

}

if (RoleBox == "[Промежуточный результат 1]")

{

EditIntermediateResult1 rt = new EditIntermediateResult1();

rt.ShowDialog();

}

if (RoleBox == "[Промежуточный результат 2]")

{

EditIntermediateResult2 rt = new EditIntermediateResult2();

rt.ShowDialog();

}

if (RoleBox == "[Промежуточный результат 3]")

{

EditIntermediateResult3 rt = new EditIntermediateResult3();

rt.ShowDialog();

}

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, cmbTab.Text);

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

}

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from " + RoleBox + " where " + NameBox.ToString() + " = '" + rowView[0].ToString() +"'";

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

DataTable dt = new DataTable();

da.Fill(dt);

txtNumberArea.Content = dt.Rows[0][0].ToString();

txtArea.Content = dt.Rows[0][1].ToString();

txtProfile.Content = dt.Rows[0][2].ToString();

txtPerimeterLength.Content = dt.Rows[0][3].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\" + tr.ToString();

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from " + RoleBox);

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

{

sw.WriteLine(Number1 + ":" + dt.Rows[i][0].ToString());

sw.WriteLine(NumberPicket1 + ":" + dt.Rows[i][1].ToString());

sw.WriteLine(Index1 + ":" + dt.Rows[i][2].ToString());

sw.WriteLine(Size1 + ":" + dt.Rows[i][3].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe",path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

namespace WindowRBD1.FormsMain

{

public partial class Measuring : System.Windows.Window

{

public Measuring()

{

InitializeComponent();

MeasuringCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer form1 = new Cursomer();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project form1 = new Project();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract form1 = new Contract();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, cmbTab.Text);

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

}

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void OrderCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Order form1 = new Order();

form1.Show();

}

private void EngineerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Engineer form1 = new Engineer();

form1.Show();

}

private void ChiefCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Chief form1 = new Chief();

form1.Show();

}

private void WorkerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Worker form1 = new Worker();

form1.Show();

}

private void DriverCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Driver form1 = new Driver();

form1.Show();

}

private void SupervisorCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Supervisor form1 = new Supervisor();

form1.Show();

}

private void MeasuringCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.[Измерительное оборудование]";

uri = 13;

cmbTab.ItemsSource = new string[] { "[Номер измерительного оборудования]", "[Наименование]" };

RoleBox = "Proekt.[Измерительное оборудование]";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.[Измерительное оборудование]");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.[Измерительное оборудование]"].DefaultView;

}

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative form1 = new Generative();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry form1 = new Telemetry();

form1.Show();

}

private void MethodologyCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Methodology form1 = new Methodology();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket form1 = new Picket();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area form1 = new Area();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile form1 = new Profile();

form1.Show();

}

private void EquipmentCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Equipment form1 = new Equipment();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateMeasuring form1 = new CreateMeasuring();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditMeasuring form1 = new EditMeasuring();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, RoleBox);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

} // end using

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.[Измерительное оборудование] where [Номер измерительного оборудования] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtNumber.Content = dr.Rows[0][0].ToString();

txtName.Content = dr.Rows[0][1].ToString();

txtInventory.Content = dr.Rows[0][2].ToString();

dateVerification.Content = dr.Rows[0][3].ToString();

datePurchases.Content = dr.Rows[0][4].ToString();

txtCharacteristic.Text = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportMeasuring.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.[Измерительное оборудование]");

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

{

sw.WriteLine("[Номер измерительного оборудования]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Наименование]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Инвентарный номер]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Дата приобретения]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Дата поверки]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Характеристики]:" + dt.Rows[i][5].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

namespace WindowRBD1.FormsMain

{

public partial class Methodology : System.Windows.Window

{

public Methodology()

{

InitializeComponent();

MethodologyCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer form1 = new Cursomer();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project form1 = new Project();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract form1 = new Contract();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, cmbTab.Text);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

} // end using

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void OrderCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Order form1 = new Order();

form1.Show();

}

private void EngineerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Engineer form1 = new Engineer();

form1.Show();

}

private void ChiefCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Chief form1 = new Chief();

form1.Show();

}

private void WorkerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Worker form1 = new Worker();

form1.Show();

}

private void DriverCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Driver form1 = new Driver();

form1.Show();

}

private void SupervisorCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Supervisor form1 = new Supervisor();

form1.Show();

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring form1 = new Measuring();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative form1 = new Generative();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry form1 = new Telemetry();

form1.Show();

}

private void MethodologyCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.Методика";

uri = 16;

cmbTab.ItemsSource = new string[] { "[Номер методики]", "[Наименование методики]" };

RoleBox = "Proekt.Методика";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Методика");

dataGridView1.ItemsSource = ds.Tables["Proekt.Методика"].DefaultView;

}

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket form1 = new Picket();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area form1 = new Area();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile form1 = new Profile();

form1.Show();

}

private void EquipmentCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Equipment form1 = new Equipment();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateMethodology form1 = new CreateMethodology();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditMethodology form1 = new EditMethodology();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.Методика where [Номер методики] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

NumberMethodology.Content = dr.Rows[0][1].ToString();

NameMethodology.Content = dr.Rows[0][1].ToString();

txtGenerative.Content = dr.Rows[0][2].ToString();

txtMeasuring.Content = dr.Rows[0][3].ToString();

txtTelemetry.Content = dr.Rows[0][4].ToString();

txtImpulse.Content = dr.Rows[0][5].ToString();

txtPause.Content = dr.Rows[0][6].ToString();

txtCurrent.Content = dr.Rows[0][7].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportMethodology.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.Методика");

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

{

sw.WriteLine("[Номер методики]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Наименование методики]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Номер описания ген.установки]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Номер описания измер.установки]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Описание телеметрической установки]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Продолжительность импульса]:" + dt.Rows[i][5].ToString());

sw.WriteLine("[Продолжительность паузы]:" + dt.Rows[i][6].ToString());

sw.WriteLine("[Сила тока]:" + dt.Rows[i][7].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

namespace WindowRBD1.FormsMain

{

public partial class Order : System.Windows.Window

{

public Order()

{

InitializeComponent();

OrderCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer form1 = new Cursomer();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project form1 = new Project();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract form1 = new Contract();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, cmbTab.Text);

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

}

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void OrderCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.[Полевой отряд]";

uri = 7;

cmbTab.ItemsSource = new string[] { "[Номер отряда]" };

RoleBox = "Proekt.[Полевой отряд]";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.[Полевой отряд]");

dataGridView1.ItemsSource = ds.Tables["Proekt.[Полевой отряд]"].DefaultView;

}

}

private void EngineerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Engineer form1 = new Engineer();

form1.Show();

}

private void ChiefCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Chief form1 = new Chief();

form1.Show();

}

private void WorkerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Worker form1 = new Worker();

form1.Show();

}

private void DriverCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Driver form1 = new Driver();

form1.Show();

}

private void SupervisorCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Supervisor form1 = new Supervisor();

form1.Show();

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring form1 = new Measuring();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative form1 = new Generative();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry form1 = new Telemetry();

form1.Show();

}

private void MethodologyCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Methodology form1 = new Methodology();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket form1 = new Picket();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area form1 = new Area();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile form1 = new Profile();

form1.Show();

}

private void EquipmentCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Equipment form1 = new Equipment();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateOrder form1 = new CreateOrder();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditOrder form1 = new EditOrder();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.[Полевой отряд] where [Номер отряда] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtNumberOrder.Content = dr.Rows[0][0].ToString();

FioChief.Content = dr.Rows[0][1].ToString();

FioITR.Content = dr.Rows[0][2].ToString();

FioDriver.Content = dr.Rows[0][3].ToString();

FioWorker.Content = dr.Rows[0][4].ToString();

FioSupervisor.Content = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportOrder.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.[Полевой отряд]");

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

{

sw.WriteLine("[Номер отряда]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Номер Начальника]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Номер ИТР]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Номер Водителя]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Номер Рабочего] :" + dt.Rows[i][4].ToString());

sw.WriteLine("[Номер Супервайзера]:" + dt.Rows[i][5].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe",path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

namespace WindowRBD1.FormsMain

{

public partial class Picket : System.Windows.Window

{

public Picket()

{

InitializeComponent();

PicketCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer form1 = new Cursomer();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project form1 = new Project();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract form1 = new Contract();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, cmbTab.Text);

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

}

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void OrderCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Order form1 = new Order();

form1.Show();

}

private void EngineerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Engineer form1 = new Engineer();

form1.Show();

}

private void ChiefCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Chief form1 = new Chief();

form1.Show();

}

private void WorkerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Worker form1 = new Worker();

form1.Show();

}

private void DriverCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Driver form1 = new Driver();

form1.Show();

}

private void SupervisorCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Supervisor form1 = new Supervisor();

form1.Show();

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring form1 = new Measuring();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative form1 = new Generative();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry form1 = new Telemetry();

form1.Show();

}

private void MethodologyCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Methodology form1 = new Methodology();

form1.Show();

}

private void PicketCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.Пикет";

uri = 4;

cmbTab.ItemsSource = new string[] { "[Номер пикета]", "[Наименование пикета]" };

RoleBox = "Proekt.Пикет";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Пикет");

dataGridView1.ItemsSource = ds.Tables["Proekt.Пикет"].DefaultView;

}

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area form1 = new Area();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile form1 = new Profile();

form1.Show();

}

private void EquipmentCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Equipment form1 = new Equipment();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreatePicket form1 = new CreatePicket();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditPicket form1 = new EditPicket();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.Пикет where [Номер пикета] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtNumberPicket.Content = dr.Rows[0][0].ToString();

txtNamePicket.Content = dr.Rows[0][1].ToString();

txtCoordinate.Content = dr.Rows[0][2].ToString();

txtTypeMeasurement.Content = dr.Rows[0][3].ToString();

NumberOrder.Content = dr.Rows[0][4].ToString();

NumberMethodology.Content = dr.Rows[0][5].ToString();

NumberEquipment.Content = dr.Rows[0][6].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportPicket.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.Пикет");

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

{

sw.WriteLine("[Номер пикета]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Наименование пикета]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Координата]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Вид измерения]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Номер отряда]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Номер методики]:" + dt.Rows[i][5].ToString());

sw.WriteLine("[Номер оборудования]:" + dt.Rows[i][6].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe",path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

namespace WindowRBD1.FormsMain

{

public partial class Profile : System.Windows.Window

{

public Profile()

{

InitializeComponent();

ProfileCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer form1 = new Cursomer();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project form1 = new Project();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract form1 = new Contract();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, cmbTab.Text);

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

}

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void OrderCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Order form1 = new Order();

form1.Show();

}

private void EngineerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Engineer form1 = new Engineer();

form1.Show();

}

private void ChiefCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Chief form1 = new Chief();

form1.Show();

}

private void WorkerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Worker form1 = new Worker();

form1.Show();

}

private void DriverCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Driver form1 = new Driver();

form1.Show();

}

private void SupervisorCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Supervisor form1 = new Supervisor();

form1.Show();

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring form1 = new Measuring();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative form1 = new Generative();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry form1 = new Telemetry();

form1.Show();

}

private void MethodologyCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Methodology form1 = new Methodology();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket form1 = new Picket();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area form1 = new Area();

form1.Show();

}

private void ProfileCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.Профили";

uri = 6;

cmbTab.ItemsSource = new string[] { "[Номер профиля]", "[Наименование профиля]" };

RoleBox = "Proekt.Профили";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Профили");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Профили"].DefaultView;

}

}

private void EquipmentCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Equipment form1 = new Equipment();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateProfile form1 = new CreateProfile();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditProfile form1 = new EditProfile();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.Профили where [Номер профиля] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtNumberProfile.Content = dr.Rows[0][0].ToString();

txtNameProfile.Content = dr.Rows[0][1].ToString();

txtNumberArea.Content = dr.Rows[0][2].ToString();

txtNumberPicket.Content = dr.Rows[0][3].ToString();

txtCoordinatesBeginning.Content = dr.Rows[0][4].ToString();

txtCoordinatesFracture.Content = dr.Rows[0][5].ToString();

txtCoordinatesEnd.Content = dr.Rows[0][6].ToString();

txtLength.Content = dr.Rows[0][7].ToString();

dateBeginning.Content = dr.Rows[0][8].ToString();

dateEnd.Content = dr.Rows[0][9].ToString();

DateCreate.Content = dr.Rows[0][10].ToString();

DateEdit.Content = dr.Rows[0][11].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportProfile.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.Профили");

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

{

sw.WriteLine("[Номер профиля]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Наименование профиля]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Номер площади]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Номер пикета]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Координаты начала]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Координаты изломов]:" + dt.Rows[i][5].ToString());

sw.WriteLine("[Координаты окончания]:" + dt.Rows[i][6].ToString());

sw.WriteLine("[Длина]:" + dt.Rows[i][7].ToString());

sw.WriteLine("[Дата и время начала работ]:" + dt.Rows[i][8].ToString());

sw.WriteLine("[Дата и время окончания работ]:" + dt.Rows[i][9].ToString());

sw.WriteLine("[Дата и время появления записи]:" + dt.Rows[i][10].ToString());

sw.WriteLine("[Дата и время изменения записи]:" + dt.Rows[i][11].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

using iTextSharp.text.pdf;

using Microsoft.Win32;

using Org.BouncyCastle.Asn1.X9;

using Spire.Xls;

using System.Collections.Generic;

using GrapeCity.Documents.Spread;

using Microsoft.Azure.ActiveDirectory.GraphClient;

using System.Runtime.InteropServices;

using System.Windows.Forms;

using Microsoft.Azure.ActiveDirectory.GraphClient.Internal;

using static System.Net.Mime.MediaTypeNames;

using System.Windows.Input;

using Magnum.FileSystem;

using DocumentFormat.OpenXml.Drawing.Charts;

namespace WindowRBD1.FormsMain

{

public partial class Project : System.Windows.Window

{

public Project()

{

InitializeComponent();

ProjectCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

public object ExcelFileReader { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer form1 = new Cursomer();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.Проект";

uri = 2;

cmbTab.ItemsSource = new string[] { "[Номер проекта]", "[Наименование проекта]" };

RoleBox = "Proekt.Проект";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Проект");

dataGridView1.ItemsSource = ds.Tables["Proekt.Проект"].DefaultView;

}

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract form1 = new Contract();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, cmbTab.Text);

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

}

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void OrderCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Order form1 = new Order();

form1.Show();

}

private void EngineerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Engineer form1 = new Engineer();

form1.Show();

}

private void ChiefCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Chief form1 = new Chief();

form1.Show();

}

private void WorkerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Worker form1 = new Worker();

form1.Show();

}

private void DriverCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Driver form1 = new Driver();

form1.Show();

}

private void SupervisorCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Supervisor form1 = new Supervisor();

form1.Show();

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring form1 = new Measuring();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative form1 = new Generative();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry form1 = new Telemetry();

form1.Show();

}

private void MethodologyCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Methodology form1 = new Methodology();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket form1 = new Picket();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area form1 = new Area();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile form1 = new Profile();

form1.Show();

}

private void EquipmentCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Equipment form1 = new Equipment();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateProekt form1 = new CreateProekt();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditProekt form1 = new EditProekt();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from Proekt.Проект where [Номер проекта] = " + rowView[0].ToString() , con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.Проект where [Номер проекта] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

NumberProekt.Content = dr.Rows[0][0].ToString();

NameProekt.Content = dr.Rows[0][1].ToString();

NumberClient.Content = dr.Rows[0][2].ToString();

NumberContract.Content = dr.Rows[0][3].ToString();

NumberArea.Content = dr.Rows[0][4].ToString();

DateCreate.Content = dr.Rows[0][5].ToString();

DateEdit.Content = dr.Rows[0][6].ToString();

da.Dispose();

conn.Close();

}

}

private void ExportToXSL\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Excel.Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Excel.Worksheet sheet1 = (Excel.Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportProject.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.Проект");

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

{

sw.WriteLine("[Номер проекта]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Название проекта]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Номер заказчика]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Номер договора]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Номер площади]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Дата и время появления записи]:" + dt.Rows[i][5].ToString());

sw.WriteLine("[Дата и время изменения записи]:" + dt.Rows[i][6].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

namespace WindowRBD1.FormsMain

{

public partial class Supervisor : System.Windows.Window

{

public Supervisor()

{

InitializeComponent();

SupervisorCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer form1 = new Cursomer();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project form1 = new Project();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract form1 = new Contract();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, cmbTab.Text);

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

}

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void OrderCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Order form1 = new Order();

form1.Show();

}

private void EngineerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Engineer form1 = new Engineer();

form1.Show();

}

private void ChiefCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Chief form1 = new Chief();

form1.Show();

}

private void WorkerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Worker form1 = new Worker();

form1.Show();

}

private void DriverCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Driver form1 = new Driver();

form1.Show();

}

private void SupervisorCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.Супервайзер";

uri = 12;

cmbTab.ItemsSource = new string[] { "[Номер Супервайзера]", "ФИО" };

RoleBox = "Proekt.Супервайзер";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Супервайзер");

dataGridView1.ItemsSource = ds.Tables["Proekt.Супервайзер"].DefaultView;

}

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring form1 = new Measuring();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative form1 = new Generative();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry form1 = new Telemetry();

form1.Show();

}

private void MethodologyCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Methodology form1 = new Methodology();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket form1 = new Picket();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area form1 = new Area();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile form1 = new Profile();

form1.Show();

}

private void EquipmentCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Equipment form1 = new Equipment();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateSupervisor form1 = new CreateSupervisor();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditSupervisor form1 = new EditSupervisor();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.[Супервайзер] where [Номер Супервайзера] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtNumberSupervisor.Content = dr.Rows[0][0].ToString();

txtFIO.Content = dr.Rows[0][1].ToString();

txtCompetence.Content = dr.Rows[0][2].ToString();

txtExperience.Content = dr.Rows[0][3].ToString();

txtExperienced.Content = dr.Rows[0][4].ToString();

dateMedical.Content = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportSupervisor.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.[Супервайзер]");

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

{

sw.WriteLine("[Номер Супервайзера]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[ФИО]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Квалификация]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Опыт работы в коллективе]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Общий опыт работы по специальности]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Дата прохождения медосмотра]:" + dt.Rows[i][5].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

using System.Xml.Linq;

namespace WindowRBD1.FormsMain

{

public partial class Telemetry : System.Windows.Window

{

public Telemetry()

{

InitializeComponent();

TelemetryCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer form1 = new Cursomer();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project form1 = new Project();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract form1 = new Contract();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, cmbTab.Text);

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

}

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void OrderCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Order form1 = new Order();

form1.Show();

}

private void EngineerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Engineer form1 = new Engineer();

form1.Show();

}

private void ChiefCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Chief form1 = new Chief();

form1.Show();

}

private void WorkerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Worker form1 = new Worker();

form1.Show();

}

private void DriverCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Driver form1 = new Driver();

form1.Show();

}

private void SupervisorCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Supervisor form1 = new Supervisor();

form1.Show();

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring form1 = new Measuring();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative form1 = new Generative();

form1.Show();

}

private void TelemetryCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.[Телеметрическое оборудование]";

uri = 15;

cmbTab.ItemsSource = new string[] { "[Номер телеметрического оборудования]", "[Наименование]" };

RoleBox = "Proekt.[Телеметрическое оборудование]";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.[Телеметрическое оборудование]");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.[Телеметрическое оборудование]"].DefaultView;

}

}

private void MethodologyCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Methodology form1 = new Methodology();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket form1 = new Picket();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area form1 = new Area();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile form1 = new Profile();

form1.Show();

}

private void EquipmentCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Equipment form1 = new Equipment();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateTelemetry form1 = new CreateTelemetry();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditTelemetry form1 = new EditTelemetry();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, RoleBox);

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

} // end using

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.[Телеметрическое оборудование] where [Номер телеметрического оборудования] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtNumber.Content = dr.Rows[0][0].ToString();

txtName.Content = dr.Rows[0][1].ToString();

txtInventory.Content = dr.Rows[0][2].ToString();

dateVerification.Content = dr.Rows[0][3].ToString();

datePurchases.Content = dr.Rows[0][4].ToString();

txtCharacteristic.Text = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportTelemetry.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.[Телеметрическое оборудование]");

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

{

sw.WriteLine("[Номер телеметрического оборудования]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Наименование]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Инвентарный номер]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Дата приобретения]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Дата поверки]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Характеристики]:" + dt.Rows[i][5].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

namespace WindowRBD1.FormsMain

{

public partial class Worker : System.Windows.Window

{

public Worker()

{

InitializeComponent();

WorkerCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer form1 = new Cursomer();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project form1 = new Project();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract form1 = new Contract();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, cmbTab.Text);

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

}

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void OrderCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Order form1 = new Order();

form1.Show();

}

private void EngineerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Engineer form1 = new Engineer();

form1.Show();

}

private void ChiefCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Chief form1 = new Chief();

form1.Show();

}

private void WorkerCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.Рабочие";

uri = 10;

cmbTab.ItemsSource = new string[] { "[Номер Рабочего]", "ФИО" };

RoleBox = "Proekt.Рабочие";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Рабочие");

dataGridView1.ItemsSource = ds.Tables["Proekt.Рабочие"].DefaultView;

}

}

private void DriverCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Driver form1 = new Driver();

form1.Show();

}

private void SupervisorCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Supervisor form1 = new Supervisor();

form1.Show();

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring form1 = new Measuring();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative form1 = new Generative();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry form1 = new Telemetry();

form1.Show();

}

private void MethodologyCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Methodology form1 = new Methodology();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket form1 = new Picket();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area form1 = new Area();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile form1 = new Profile();

form1.Show();

}

private void EquipmentCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Equipment form1 = new Equipment();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateWorker form1 = new CreateWorker();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditWorker form1 = new EditWorker();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.[Рабочие] where [Номер Рабочего] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtNumberWorker.Content = dr.Rows[0][0].ToString();

txtFIO.Content = dr.Rows[0][1].ToString();

txtCompetence.Content = dr.Rows[0][2].ToString();

txtExperience.Content = dr.Rows[0][3].ToString();

txtExperienced.Content = dr.Rows[0][4].ToString();

dateMedical.Content = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportWorker.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.[Рабочие]");

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

{

sw.WriteLine("[Номер Рабочего]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[ФИО]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Квалификация]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Опыт работы в коллективе]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Общий опыт работы по специальности]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Дата прохождения медосмотра]:" + dt.Rows[i][5].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

using WindowRBD1.FormsMain;

namespace WindowRBD1.FormsMainOperator

{

public partial class Area1 : System.Windows.Window

{ public Area1()

{

InitializeComponent();

AreaCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer1 form1 = new Cursomer1();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project1 form1 = new Project1();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract1 form1 = new Contract1();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, cmbTab.Text);

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

}

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring1 form1 = new Measuring1();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative1 form1 = new Generative1();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry1 form1 = new Telemetry1();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket1 form1 = new Picket1();

form1.Show();

}

private void AreaCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.Площади";

uri = 5;

cmbTab.ItemsSource = new string[] { "[Номер площади]", "[Наименование площади]" };

RoleBox = "Proekt.Площади";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Площади");

dataGridView1.ItemsSource = ds.Tables["Proekt.Площади"].DefaultView;

}

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile1 form1 = new Profile1();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateArea form1 = new CreateArea();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditArea form1 = new EditArea();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Close();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.Площади where [Номер площади] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

DataTable dt = new DataTable();

da.Fill(dt);

txtNumberArea.Content = dt.Rows[0][0].ToString();

txtArea.Content = dt.Rows[0][1].ToString();

txtProfile.Content = dt.Rows[0][2].ToString();

txtList.Content = dt.Rows[0][3].ToString();

txtPerimeterLength.Content = dt.Rows[0][4].ToString();

txtSizeArea.Content = dt.Rows[0][5].ToString();

dateBeginning.Content = dt.Rows[0][6].ToString();

dateEnd.Content = dt.Rows[0][7].ToString();

DateCreate.Content = dt.Rows[0][8].ToString();

DateEdit.Content = dt.Rows[0][9].ToString();

txtSupervisorData.Content = dt.Rows[0][10].ToString();

txtSupervisorOrder.Content = dt.Rows[0][11].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportArea.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.Площади");

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

{

sw.WriteLine("[Номер площади]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Наименование площади]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Номер списка углов периметра]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Величина площади окружённая периметром]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Дата и время окончания работ]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Дата и время появления записи]:" + dt.Rows[i][5].ToString());

sw.WriteLine("[Дата и время изменения записи]:" + dt.Rows[i][6].ToString());

sw.WriteLine("[Супервайзер полевых работ]:" + dt.Rows[i][7].ToString());

sw.WriteLine("[Супервайзер обработки данных]:" + dt.Rows[i][8].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

using WindowRBD1.FormsMain;

namespace WindowRBD1.FormsMainOperator

{

public partial class Contract1 : System.Windows.Window

{

public Contract1()

{

InitializeComponent();

ContractCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer1 form1 = new Cursomer1();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project1 form1 = new Project1();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ContractCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.Договор";

uri = 3;

cmbTab.ItemsSource = new string[] { "[Номер договора]", "[Наименование договора]" };

RoleBox = "Proekt.Договор";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Договор");

dataGridView1.ItemsSource = ds.Tables["Proekt.Договор"].DefaultView;

}

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, cmbTab.Text);

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

}

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring1 form1 = new Measuring1();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative1 form1 = new Generative1();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry1 form1 = new Telemetry1();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket1 form1 = new Picket1();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area1 form1 = new Area1();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile1 form1 = new Profile1();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateContract form1 = new CreateContract();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditContract form1 = new EditContract();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Close();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.Договор where [Номер договора] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

DataTable dt = new DataTable();

da.Fill(dt);

txtNumberContract.Content = dt.Rows[0][0].ToString();

txtNameContract.Content = dt.Rows[0][1].ToString();

txtBeginnings.Content = dt.Rows[0][2].ToString();

txtEndings.Content = dt.Rows[0][3].ToString();

txtCost.Content = dt.Rows[0][4].ToString();

DateCreate.Content = dt.Rows[0][5].ToString();

DateEdit.Content = dt.Rows[0][6].ToString();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportContract.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.Договор");

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

{

sw.WriteLine("[Номер договора]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Наименование договора]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Начала]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Окончания]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Стоимость]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Дата и время появления записи]:" + dt.Rows[i][5].ToString());

sw.WriteLine("[Дата и время изменения записи]:" + dt.Rows[i][5].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql)// Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

using WindowRBD1.FormsMain;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement.ListView;

using System.Security.Policy;

using System.Data.Entity.Core.Common.CommandTrees.ExpressionBuilder;

namespace WindowRBD1.FormsMainOperator

{

public partial class Cursomer1 : System.Windows.Window

{

public Cursomer1()

{

InitializeComponent();

CursomerCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.Заказчик";

cmbTab.ItemsSource = new string[] { "[Номер заказчика]", "[Название компаний]", "[Представитель]" };

RoleBox = "Proekt.Заказчик";

uri = 1;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Заказчик");

dataGridView1.ItemsSource = ds.Tables["Proekt.Заказчик"].DefaultView;

}

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project1 form1 = new Project1();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract1 form1 = new Contract1();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, cmbTab.Text);

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

}

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring1 form1 = new Measuring1();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative1 form1 = new Generative1();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry1 form1 = new Telemetry1();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket1 form1 = new Picket1();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area1 form1 = new Area1();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile1 form1 = new Profile1();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateClient form1 = new CreateClient();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditClient form1 = new EditClient();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Close();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.Заказчик where [Номер заказчика] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

DataTable dt = new DataTable();

da.Fill(dt);

NumberClient.Content = dt.Rows[0][0].ToString();

txtNameCompany.Content = dt.Rows[0][1].ToString();

txtLegalAddress.Text = dt.Rows[0][2].ToString();

txtActualAddress.Text = dt.Rows[0][3].ToString();

txtINN.Content = dt.Rows[0][4].ToString();

txtPda.Content = dt.Rows[0][5].ToString();

txtCalculated.Content = dt.Rows[0][6].ToString();

txtCorrespondent.Content = dt.Rows[0][7].ToString();

txtAgent.Content = dt.Rows[0][8].ToString();

txtPhone.Content = dt.Rows[0][9].ToString();

Email.Text = dt.Rows[0][10].ToString();

Site.Text = dt.Rows[0][11].ToString();

DateCreate.Content = dt.Rows[0][12].ToString();

DateEdit.Content = dt.Rows[0][13].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportCursomer.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.[Заказчик]");

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

{

sw.WriteLine("[Номер заказчика]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Название компаний]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Юридический адрес]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Фактический адрес]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[ИНН]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[КПК]:" + dt.Rows[i][5].ToString());

sw.WriteLine("[Расчетный счёт]:" + dt.Rows[i][6].ToString());

sw.WriteLine("[Корреспондентский счёт]:" + dt.Rows[i][7].ToString());

sw.WriteLine("[Представитель]:" + dt.Rows[i][8].ToString());

sw.WriteLine("[Телефон представителя]:" + dt.Rows[i][9].ToString());

sw.WriteLine("[Адрес электронной почты]:" + dt.Rows[i][10].ToString());

sw.WriteLine("[Сайт]:" + dt.Rows[i][11].ToString());

sw.WriteLine("[Дата и время появления записи]:" + dt.Rows[i][12].ToString());

sw.WriteLine("[Дата и время изменения записи]:" + dt.Rows[i][13].ToString());

sw.WriteLine(" ");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

using WindowRBD1.FormsMain;

namespace WindowRBD1.FormsMainOperator

{

public partial class Generative1 : System.Windows.Window

{

public Generative1()

{

InitializeComponent();

GenerativeCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer1 form1 = new Cursomer1();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project1 form1 = new Project1();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract1 form1 = new Contract1();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, cmbTab.Text);

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

}

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring1 form1 = new Measuring1();

form1.Show();

}

private void GenerativeCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.[Генеративное оборудование]";

uri = 14;

cmbTab.ItemsSource = new string[] { "[Номер генеративного оборудования]", "[Наименование]" };

RoleBox = "Proekt.[Генеративное оборудование]";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.[Генеративное оборудование]");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.[Генеративное оборудование]"].DefaultView;

}

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry1 form1 = new Telemetry1();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket1 form1 = new Picket1();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area1 form1 = new Area1();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile1 form1 = new Profile1();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateGenerative form1 = new CreateGenerative();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditGenerative form1 = new EditGenerative();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Close();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.[Генеративное оборудование] where [Номер генеративного оборудования] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtNumber.Content = dr.Rows[0][0].ToString();

txtName.Content = dr.Rows[0][1].ToString();

txtInventory.Content = dr.Rows[0][2].ToString();

dateVerification.Content = dr.Rows[0][3].ToString();

datePurchases.Content = dr.Rows[0][4].ToString();

txtCharacteristic.Text = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportGenerative.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.[Генеративное оборудование]");

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

{

sw.WriteLine("[Номер генеративного оборудования]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Наименование]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Инвентарный номер]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Дата приобретения]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Дата поверки]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Характеристики]:" + dt.Rows[i][5].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

using System.Xml.Linq;

using WindowRBD1.FormsMain;

namespace WindowRBD1.FormsMainOperator

{

public partial class Measuring1 : System.Windows.Window

{

public Measuring1()

{

InitializeComponent();

MeasuringCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer1 form1 = new Cursomer1();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project1 form1 = new Project1();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract1 form1 = new Contract1();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, cmbTab.Text);

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

}

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void MeasuringCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.[Измерительное оборудование]";

uri = 13;

cmbTab.ItemsSource = new string[] { "[Номер измерительного оборудования]", "[Наименование]" };

RoleBox = "Proekt.[Измерительное оборудование]";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.[Измерительное оборудование]");

dataGridView1.ItemsSource = ds.Tables["Proekt.[Измерительное оборудование]"].DefaultView;

}

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative1 form1 = new Generative1();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry1 form1 = new Telemetry1();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket1 form1 = new Picket1();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area1 form1 = new Area1();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile1 form1 = new Profile1();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateMeasuring form1 = new CreateMeasuring();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditMeasuring form1 = new EditMeasuring();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.[Измерительное оборудование] where [Номер измерительного оборудования] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtNumber.Content = dr.Rows[0][0].ToString();

txtName.Content = dr.Rows[0][1].ToString();

txtInventory.Content = dr.Rows[0][2].ToString();

dateVerification.Content = dr.Rows[0][3].ToString();

datePurchases.Content = dr.Rows[0][4].ToString();

txtCharacteristic.Text = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportMeasuring.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.[Измерительное оборудование]");

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

{

sw.WriteLine("[Номер измерительного оборудования]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Наименование]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Инвентарный номер]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Дата приобретения]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Дата поверки]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Характеристики]:" + dt.Rows[i][5].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

using WindowRBD1.FormsMain;

namespace WindowRBD1.FormsMainOperator

{

public partial class Picket1 : System.Windows.Window

{

public Picket1()

{

InitializeComponent();

PicketCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer1 form1 = new Cursomer1();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project1 form1 = new Project1();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract1 form1 = new Contract1();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, cmbTab.Text);

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

}

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring1 form1 = new Measuring1();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative1 form1 = new Generative1();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry1 form1 = new Telemetry1();

form1.Show();

}

private void PicketCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.Пикет";

uri = 4;

cmbTab.ItemsSource = new string[] { "[Номер пикета]", "[Наименование пикета]" };

RoleBox = "Proekt.Пикет";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Пикет");

dataGridView1.ItemsSource = ds.Tables["Proekt.Пикет"].DefaultView;

}

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area1 form1 = new Area1();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile1 form1 = new Profile1();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreatePicket form1 = new CreatePicket();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditPicket form1 = new EditPicket();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.Пикет where [Номер пикета] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtNumberPicket.Content = dr.Rows[0][0].ToString();

txtNamePicket.Content= dr.Rows[0][1].ToString();

txtCoordinate.Content = dr.Rows[0][2].ToString();

txtTypeMeasurement.Content = dr.Rows[0][3].ToString();

NumberOrder.Content = dr.Rows[0][4].ToString();

NumberMethodology.Content = dr.Rows[0][5].ToString();

NumberEquipment.Content = dr.Rows[0][6].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportPicket.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.Пикет");

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

{

sw.WriteLine("[Номер пикета]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Наименование пикета]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Координата]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Вид измерения]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Окончательный результат измерения (кривая измерения)]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Номер отряда]:" + dt.Rows[i][5].ToString());

sw.WriteLine("[Номер методики]:" + dt.Rows[i][6].ToString());

sw.WriteLine("[Номер оборудования]:" + dt.Rows[i][7].ToString());

sw.WriteLine("[Трансформанта измерения 1]:" + dt.Rows[i][8].ToString());

sw.WriteLine("[Трансформанта измерения 2]:" + dt.Rows[i][9].ToString());

sw.WriteLine("[Трансформанта измерения 3]:" + dt.Rows[i][10].ToString());

sw.WriteLine("[Промежуточные результаты измерения]:" + dt.Rows[i][11].ToString());

sw.WriteLine("[Промежуточные результаты измерения 1]:" + dt.Rows[i][12].ToString());

sw.WriteLine("[Промежуточные результаты измерения 2]:" + dt.Rows[i][13].ToString());

sw.WriteLine("[Промежуточные результаты измерения 3]:" + dt.Rows[i][14].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

using WindowRBD1.FormsMain;

namespace WindowRBD1.FormsMainOperator

{

public partial class Profile1 : System.Windows.Window

{

public Profile1()

{

InitializeComponent();

ProfileCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer1 form1 = new Cursomer1();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project1 form1 = new Project1();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract1 form1 = new Contract1();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, cmbTab.Text);

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

}

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring1 form1 = new Measuring1();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative1 form1 = new Generative1();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry1 form1 = new Telemetry1();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket1 form1 = new Picket1();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area1 form1 = new Area1();

form1.Show();

}

private void ProfileCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.Профили";

uri = 6;

cmbTab.ItemsSource = new string[] { "[Номер профиля]", "[Наименование профиля]" };

RoleBox = "Proekt.Профили";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.Профили");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.Профили"].DefaultView;

}

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateProfile form1 = new CreateProfile();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditProfile form1 = new EditProfile();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.Профили where [Номер профиля] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtNumberProfile.Content = dr.Rows[0][0].ToString();

txtNameProfile.Content = dr.Rows[0][1].ToString();

txtNumberArea.Content = dr.Rows[0][2].ToString();

txtNumberPicket.Content = dr.Rows[0][3].ToString();

txtCoordinatesBeginning.Content = dr.Rows[0][4].ToString();

txtCoordinatesFracture.Content = dr.Rows[0][5].ToString();

txtCoordinatesEnd.Content = dr.Rows[0][6].ToString();

txtLength.Content = dr.Rows[0][7].ToString();

dateBeginning.Content = dr.Rows[0][8].ToString();

dateEnd.Content = dr.Rows[0][9].ToString();

DateCreate.Content = dr.Rows[0][10].ToString();

DateEdit.Content = dr.Rows[0][11].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportProfile.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.Профили");

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

{

sw.WriteLine("[Номер профиля]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Наименование профиля]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Номер площади]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Номер пикета]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Координаты начала]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Координаты изломов]:" + dt.Rows[i][5].ToString());

sw.WriteLine("[Координаты окончания]:" + dt.Rows[i][6].ToString());

sw.WriteLine("[Длина]:" + dt.Rows[i][7].ToString());

sw.WriteLine("[Дата и время начала работ]:" + dt.Rows[i][8].ToString());

sw.WriteLine("[Дата и время окончания работ]:" + dt.Rows[i][9].ToString());

sw.WriteLine("[Дата и время появления записи]:" + dt.Rows[i][10].ToString());

sw.WriteLine("[Дата и время изменения записи]:" + dt.Rows[i][11].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

using iTextSharp.text.pdf;

using WindowRBD1.FormsMain;

namespace WindowRBD1.FormsMainOperator

{

public partial class Project1 : System.Windows.Window

{ public Project1()

{

InitializeComponent();

ProjectCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer1 form1 = new Cursomer1();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.Проект";

cmbTab.ItemsSource = new string[] { "[Номер проекта]", "[Наименование проекта]" };

RoleBox = "Proekt.Проект";

uri = 2;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, "Proekt.Проект");

dataGridView1.ItemsSource = ds.Tables["Proekt.Проект"].DefaultView;

}

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract1 form1 = new Contract1();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, cmbTab.Text);

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

}

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring1 form1 = new Measuring1();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative1 form1 = new Generative1();

form1.Show();

}

private void TelemetryCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Telemetry1 form1 = new Telemetry1();

form1.Show();

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket1 form1 = new Picket1();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area1 form1 = new Area1();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile1 form1 = new Profile1();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateProekt form1 = new CreateProekt();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditProekt form1 = new EditProekt();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.Проект where [Номер проекта] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

NumberProekt.Content = dr.Rows[0][0].ToString();

NameProekt.Content = dr.Rows[0][1].ToString();

NumberClient.Content = dr.Rows[0][2].ToString();

NumberContract.Content = dr.Rows[0][3].ToString();

NumberArea.Content = dr.Rows[0][4].ToString();

DateCreate.Content = dr.Rows[0][5].ToString();

DateEdit.Content = dr.Rows[0][6].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportProject.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.Проект");

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

{

sw.WriteLine("[Номер проекта]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Название проекта]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Номер заказчика]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Номер договора]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Номер площади]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Дата и время появления записи]:" + dt.Rows[i][5].ToString());

sw.WriteLine("[Дата и время изменения записи]:" + dt.Rows[i][6].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql) // Подсоединение к базе данных для экспорта в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

using System;

using System.Collections.ObjectModel;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Diagnostics;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Media;

using WindowRBD1.Class;

using WindowRBD1.FormsCreate;

using WindowRBD1.FormsCreate.Equipment;

using WindowRBD1.FormsCreate.Works;

using WindowRBD1.FormsEdit;

using WindowRBD1.FormsEdit.Equipment;

using WindowRBD1.FormsEdit.Works;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using DataTable = System.Data.DataTable;

using Range = Microsoft.Office.Interop.Excel.Range;

using WindowRBD1.FormsMain;

namespace WindowRBD1.FormsMainOperator

{

public partial class Telemetry1 : System.Windows.Window

{

public Telemetry1()

{

InitializeComponent();

TelemetryCommand();

}

string con = BdCon.Con;

public string RoleBox;

SqlDataAdapter dataAdapter = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dtSales = new DataTable();

string sql;

int uri;

public object openFileDialog1 { get; private set; }

private void CursomerCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Cursomer1 form1 = new Cursomer1();

form1.Show();

}

private void MeaningCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

Meaning form1 = new Meaning();

form1.Show();

}

private void ProjectCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Project1 form1 = new Project1();

form1.Show();

}

private void ContractCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Contract1 form1 = new Contract1();

form1.Show();

}

private void Button\_Click(object sender, EventArgs e, string sql) //Поисковик

{

if (txtBox1.Text == "") { sql = "select \* from " + RoleBox; }

else

{

sql = "select \* from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'";

}

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, cmbTab.Text);

dataGridView1.ItemsSource = ds.Tables[cmbTab.Text].DefaultView;

}

}

private void Button\_Click(object sender, EventArgs e) //Вызов Поисковика

{

Button\_Click(sender, e, sql);

}

private void MeasuringCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Measuring1 form1 = new Measuring1();

form1.Show();

}

private void GenerativeCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Generative1 form1 = new Generative1();

form1.Show();

}

private void TelemetryCommand() //Происходит заполнение данными datagrid

{

sql = "select \* from Proekt.[Телеметрическое оборудование]";

uri = 15;

cmbTab.ItemsSource = new string[] { "[Номер телеметрического оборудования]", "[Наименование]" };

RoleBox = "Proekt.[Телеметрическое оборудование]";

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

// Заполняем ds данными из dataAdapter:

dataAdapter.Fill(ds, "Proekt.[Телеметрическое оборудование]");

// Указываем источник данных DataSource для dataGrid1:

dataGridView1.ItemsSource = ds.Tables["Proekt.[Телеметрическое оборудование]"].DefaultView;

}

}

private void PicketCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Picket1 form1 = new Picket1();

form1.Show();

}

private void AreaCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Area1 form1 = new Area1();

form1.Show();

}

private void ProfileCommand\_Click(object sender, RoutedEventArgs e) //Происходит закрытие формы и открытие другой

{

this.Hide();

Profile1 form1 = new Profile1();

form1.Show();

}

private void AddCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

CreateTelemetry form1 = new CreateTelemetry();

form1.Show();

}

private void EditCommand\_Click(object sender, RoutedEventArgs e) //Происходит открытие формы

{

EditTelemetry form1 = new EditTelemetry();

form1.Show();

}

private void DeleteCommand\_Click(object sender, RoutedEventArgs e) //Происходит удаление записи из базы данных

{

using (SqlConnection con = new SqlConnection(BdCon.Con))

{

con.Open();

SqlCommand cmd = new SqlCommand("Delete from " + RoleBox + " where " + cmbTab.Text + "= '" + txtBox1.Text + "'", con);

cmd.ExecuteNonQuery();

System.Windows.MessageBox.Show("Запись удалена");

con.Close();

}

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void btClose\_Click(object sender, RoutedEventArgs e) // закрытие программы

{

this.Hide();

}

private void btRefresh\_Click(object sender, RoutedEventArgs e) // Обновление данными datagrid

{

ds.Clear();

sql = "select \* from " + RoleBox;

using (SqlConnection conn = new SqlConnection(con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = sql;

ds.Clear();

dataAdapter.SelectCommand = myCommand;

dataAdapter.Fill(ds, RoleBox);

dataGridView1.ItemsSource = ds.Tables[RoleBox].DefaultView;

}

}

private void dataGridView1\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e) //Происходит выделение строки и заполнение информацией элементов

{

dataGridView1.IsReadOnly = true;

DataRowView rowView = dataGridView1.SelectedValue as DataRowView;

string str = "Select \* from Proekt.[Телеметрическое оборудование] where [Номер телеметрического оборудования] = " + rowView[0].ToString();

using (SqlConnection conn = new SqlConnection(BdCon.Con))

{

conn.Open();

SqlCommand myCommand = new SqlCommand();

myCommand.Connection = conn;

myCommand.CommandText = str;

SqlDataAdapter da = new SqlDataAdapter(myCommand);

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

da.Fill(dr);

txtNumber.Content = dr.Rows[0][0].ToString();

txtName.Content = dr.Rows[0][1].ToString();

txtInventory.Content = dr.Rows[0][2].ToString();

dateVerification.Content = dr.Rows[0][3].ToString();

datePurchases.Content = dr.Rows[0][4].ToString();

txtCharacteristic.Text = dr.Rows[0][5].ToString();

da.Dispose();

conn.Close();

}

}

private void btExport\_Click(object sender, RoutedEventArgs e) // Экспорт данных из datagid в Excel

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add("C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\База Данных.xlsx");

Worksheet sheet1 = (Worksheet)workbook.Sheets[uri];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

private void ExportToTxt\_Click(object sender, RoutedEventArgs e) // Подсоединение к базе данных для экспорта в txt

{

string path = "C:\\Users\\Пользователь\\Desktop\\WindowRBD1\\ExportToTxt\\exportTelemetry.txt";

StreamWriter sw = new StreamWriter(path);

DataTable dt = Select1("Select \* from Proekt.[Телеметрическое оборудование]");

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

{

sw.WriteLine("[Номер телеметрического оборудования]:" + dt.Rows[i][0].ToString());

sw.WriteLine("[Наименование]:" + dt.Rows[i][1].ToString());

sw.WriteLine("[Инвентарный номер]:" + dt.Rows[i][2].ToString());

sw.WriteLine("[Дата приобретения]:" + dt.Rows[i][3].ToString());

sw.WriteLine("[Дата поверки]:" + dt.Rows[i][4].ToString());

sw.WriteLine("[Характеристики]:" + dt.Rows[i][5].ToString());

sw.WriteLine("");

}

sw.Close();

Process.Start("notepad.exe", path);

}

public DataTable Select1(string selectSql) // Экспорт данных из datagid в txt

{

DataTable dt = new DataTable();

SqlConnection sqlConnection = new SqlConnection(BdCon.Con);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandText = selectSql;

SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);

adapter.Fill(dt);

return dt;

}

}

}

## *Приложение Е*

{При наличии, можно привести Акт о внедрении в эксплуатацию разработанной программной системы.}