**ВВЕДЕНИЕ**

Архив – это учреждение, осуществляющее хранение, комплектование, учет и использование архивных документов. Они существуют с давних времен и были известны всем народам, достигавшим известной цивилизации.

Ведение архивов достаточно тяжелый труд, отнимающий много времени. Внимательность и аккуратность без возможности исправления неверных заметок в чистовом варианте. Также все содержимое такого места не застраховано от, например, пожара или потопа, что ведет к неминуемой гибели если не всего, то большей части архива. Но так было раньше. С появлением ЭВМ в XX веке мир начал координально меняться и на смену бумаге пришли компьютеры.

Компьютер устройство или система, способная выполнять заданную, чётко определённую, изменяемую последовательность операций. Это чаще всего операции численных расчётов и манипулирования данными, однако сюда относятся и операции ввода-вывода.

Компьютеры развились настолько, что стали главным информационным инструментом как в офисе, так и дома. Теперь почти любая работа с информацией зачастую осуществляется через компьютер — будь то набор текста или просмотр фильмов. Это относится и к хранению информации, и к её пересылке по каналам связи. Основное применение современных домашних компьютеров — навигация в Интернете и игры. Описание последовательности операций называется программой.

**Программа** – это совокупность инструкций и алгоритмов, подробно описывающих способы определенных действий. Это определение может иметь несколько разных значений, а также само слово может означать различные виды программ. Программа - это способ поэтапной организации деятельности по развертыванию какого-либо содержания, а также по организации и реализации исследовательских содержаний и процедур. Конечная цель, которую призвана реализовать программа, всегда формулируется как возможная, желательная, должная, соотносимая с каким-либо идеалом, предполагающая вариативные пути своего достижения и не содержащая изначальных гарантий своей реализуемости. В этом отношении она отличается от «сценарного» способа организации деятельности, где цель предзадаётся изначально, хотя и может варьироваться в зависимости от индивидуального мастерства социальных факторов. Программа предполагает свое переопределение, перезадание, как минимум - уточнение в ходе реализации каждого из этапов (шагов) - как в плане удержания продуцируемых новых содержаний, требующих постоянной рефлексии над собой, так и в плане отслеживания процедур деятельности и оценки их эффективности. Каждое последующее решение принимается на основе реализации предыдущих, ретроспективно видоизменяя и их. Таким образом, внутри принятой программы субъект приобретает достаточную автономность и активную позицию. В этом отношении программа противостоит «рецептурному» способу задания деятельности, предполагающему точную выверенность каждого шага и решения и запрещающему какую-либо самодеятельность субъекта, который обязан строго следовать предписанному плану.

Таким образом программа – это своеобразное поэтапное руководство для нашего вычислительного устройства (компьютера), которое определяет порядок действий всех систем (электронных компонентов), входящих в его состав и функционирующих как некая дружная команда в процессе выполнения поставленной задачи.

Из вышесказанного можно сделать вывод, что актуальной темой курсового проекта будет являться продукт машинного кода: программа для создания архива конструктивных чертежей.

**1 ПОСТАНОВКА ЗАДАЧИ**

**1.1 Исследование предметной области**

Работу над проектом следует начинать с изучения предметной области: такие как эскизных чертежей планов, разрезов, фасада, планов перекрытия, покрытия и фундаментов. Назначение эскизных чертежей, выявление и принципиальная разработка объемно-планировочного и конструктивного решения здания и его отдельных частей. Эскизирование позволяет своевременно устранить всевозможные ошибки, выявить непонятные части в решении проекта, тем самым избежать переделок и исправлений в чертежах проекта.

Задание курсового проекта, разработать программу для создания архива конструктивных чертежей, которую смогут использовать большинство инженеров различных отраслей. Программа проста в использовании и является конкурентноспособной в сети интернет, так как аналогов существует немного.

Для того, чтобы облегчить работу пользователю и свести все необходимые функции в одном проекте, было принято решение написать программу для создания архива конструктивных чертежей.

В программе необходимо реализовать следующие функции:

1. добавление, удаление и редактирование чертежей и характеристик;
2. просмотр чертежей и характеристик;
3. поиск чертежей по характеристикам и первым символам;
4. сортирование по наименованию, обозначению, среде, назначению.

**1.2 Входная и выходная информация**

Входная информация – это информация, которая вводится в программу пользователем и затем обрабатывается программой. В программе «Архив для создания конструктивных чертежей» для ввода информации используется мышь и клавиатура.

С помощью клавиатуры идет ввод информации о чертежах и их характеристиках, а также во время поиска.

С помощью мыши ввод информации происходит во время выбора какой-либо функции, нажатия на кнопку или выбор вкладки.

К выходной информации можно отнести следующее:

* вывод на экран сообщений программой о выполненных действиях;
* вывод на экран результатов поиска;
* вывод в таблицах отсортированных данных.

Одним из главных выходных и входных элементов являются изображения в базе данных, хранящиеся в поле «MediumBlob», которое позволяет записывать данные вплоть до 16 Мб. Данного размера достаточно для загрузки изображений высокого разрешения и отсутствия перегрузки сервера, что оказывает положительное влияние на быстродействие всей программы в целом. Чертежи хранятся в формате PDF, который представляет собой международный формат, что позволяет использовать программу эффективно на мировой арене. Так как в Delphi не предусмотрена встроенная возможность просмотра PDF ни при помощи библиотек, ни при помощи компонентов – была установлена специальная библиотека ActiveX. Из данной библиотеки был взят компонент TAcroPDF, благодаря которому можно просматривать изображения формата PDF.

**2 ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА**

**2.1 Выбор программных средств и его обоснование**

Для создания базы данных в программе была использована программа «Open Server».

Программа «Open Server» содержит в себе все компоненты для работы с базой данных. Для создания базы данных был использован MySQL – Server который был уже встроен в программу «Open Server». MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, XAMPP. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы. Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц. Для удобной работы с MySQL использовали панель управления «PhpMyAdmin».

Для создания программы для создания архива конструктивных чертежей использовали Delphi 10 Seattle. RAD Studio 10 Seattle – это законченное средство для быстрой разработки кроссплатформенных приложений с помощью Object Pascal и C++. Данная программа применяется для создания приложений на нескольких языках для платформ, поддерживаемых новой версией RAD Studio. Кроме того, проведенная работа позволит реализовать для всех поддерживаемых платформ и языков новые возможности и функции, обеспечивающие повышенную стабильность, качество и эффективность документации. RAD Studio 10 – это новый стандарт стабильности, качества и эффективной документации.

**2.2 Операционная система**

Для написания программы для создания архива конструктивных чертежей была выбрана операционная система Windows 7. Выбор такой операционной системы был обусловлен тем, что в ней довольно просто проектировать, разрабатывать программу и налаживать связи с таблицами базы данных.

Одним из основных преимуществ Windows 7 является то, что эта операционная система установлена на большинстве современных стационарных компьютеров, нетбуков и ноутбуков, в связи с этим намного проще выполнить восстановление Windows XP после вирусов или иные распространенные процедуры. С точки зрения пользовательского интерфейса ОС является достаточно удобной. Интерфейс пользователя версии 7, если сравнивать с Windows XP и Vista, характеризуется насыщенностью красок и стилями оформления, поэтому желающие установить видео на рабочий стол, будут порадованы очень интересной картинкой. Вдобавок здесь быстро устанавливается ПО, позволяющее искать свободные ip адреса или программа Windows запись iso. Еще одно преимущество данной системы — это стабильная работа. Но нужно сказать, что стабильная работа зависит от используемого железа, наличия обновлений, версий драйверов, наличия выхода Интернет, установленных приложений, правильной настройки и т.д.

Windows 7 предлагает огромный перечень настроек – начиная от графического интерфейса и заканчивая правами доступа и настройкой безопасности. Важным показателем является хорошая совместимость с распространенными программными продуктами. Windows 7 имеет высокую степень устойчивости к ошибочным и неосторожным действиям пользователей, а это также имеет особое значение при выборе системы.

Среда Windows 7 характеризуется следующими особенностями, отличающими её от других программ — оболочек:

1. восстановление системы в случае возникновения ошибки. Если в результате установки программы или по другой причине компьютер перестал работать, с помощью системы восстановления можно вернуть работоспособность, сохранив при этом важную информацию;
2. хорошая совместимость с прошлыми версиями Windows;
3. большие мультимедийные возможности;
4. удобный интерфейс Windows 7 имеет оригинальный и функциональный дизайн;
5. усовершенствованные возможности резервного копирования – мобильные пользователи получают возможность создавать резервные копии в сети, а не на локальных дисках;
6. обеспечение высокой производительности труда пользователей – обеспечивает высокую производительность труда конечных пользователей практически везде;
7. повышенная безопасность и контроль – основывается на системе безопасности и дает большую свободу действий, отвечающим за безопасность ПК и данных;
8. упрощенное управление ПК;
9. локальный поиск Windows – помощь в поиске файлов на локальном компьютере или в сети;
10. подключение к домену – простота и безопасность подключения к серверу по сети.

**2.3 Система программирования**

Программа для создания архива конструктивных чертежей была разработана в среде разработки «Delphi 10 Seattle». Delphi 10 Seattle – это самый быстрый способ создавать и обновлять интенсивно работающие с данными, распределенные и сильно взаимодействующие приложения с развитым и визуально привлекательным пользовательским интерфейсом для Windows 10, Mac, мобильных устройств и других платформ.

Вы сможете с легкостью обновлять приложения VCL и FMX для Windows 10, используя новые элементы управления и стили VCL для Windows и компоненты служб универсальной платформы Windows. Откройте для себя удобную среду разработки с удвоенным объемом доступной памяти для крупных проектов, расширенной поддержкой нескольких мониторов и улучшенным инспектором объектов. С помощью Delphi 10 разработчики смогут в 5 раз быстрее создавать приложения для настольных, мобильных, облачных платформ и платформ баз данных, включая 32- и 64-разрядную версии Windows 10.

Разработчики на Delphi смогут быстро обновить свои VCL- и FMX-приложения и в полной мере воспользоваться бурным ростом интереса к Windows 10. Поддерживаются компоненты Windows 10 и «родные» API и компоненты WinRT/UWP, элементы интерфейса Windows 10 VCL. Также обновлена поддержка Windows 10 FMX.

Системные требования:

* 1 GB RAM (рекомендуется 2 GB+);
* 3–9 GB свободного пространства на жестком диске в зависимости от редакции и конфигурации;
* обычный GPU с поддержкой DirectX 9.0 или лучше (Pixel Shader Level 2);
* разрешение 1024×768 или более высокое;
* intel® Pentium® или совместимый, минимум 1.6 GHz (рекомендуется 2GHz+);
* мышь или другое указывающее устройство.

Из незаметных для пользователя изменений можно отметить переработанную в данной версии архитектуру системы управления продуктами и сборками, которая позволила фактически удвоить поддерживаемый размер проектов и повысить стабильность и производительность при работе с крупными проектами, особенно для нескольких платформ. Обновление до этой версии пригодится разработчикам, которые сталкивались с ограничениями ресурсов и другими проблемами в крупных проектах.

Процедуры разработки, тестирования и составления документации были серьезно доработаны. Теперь они лучше подходят для создания приложений на нескольких языках для платформ, поддерживаемых новой версией RAD Studio. Кроме того, проведенная работа позволила реализовать для всех поддерживаемых платформ и языков новые возможности и функции, обеспечивающие повышенную стабильность, качество и эффективность документации. RAD Studio 10 Seattle – это новый стандарт стабильности, качества и эффективной документации.

RAD Studio 10 Seattle является тем самым «мультипликатором производительности» производителей, позволяя быстро и визуальными методами буквально ставить на поток создание приложений для Windows 10, Mac, .NET, Web и мобильных платформ.

Новые возможности в«Delphi 10 Seattle»:

1. поддержка параллельной компиляции C++;
2. отладка iOS 64x приложений;
3. поддерживается iOS 8.4;
4. поддержка модульного тестирования DUnitX для Android и iOS;
5. поддержка DirectX 12;
6. поддержка вызова API WinRT;
7. поддержка FireDAC для базы данных NoSQL MongoDB
8. новое поведение MultiView;
9. новые компоненты VCL;
10. новые компоненты для работы с Beacon;
11. улучшен механизм стилей;
12. улучшен диспетчер библиотек GetIt;
13. улучшены возможности IDE.

**3 ПРОЕКТИРОВАНИЕ ПРОГРАММ**

**3.1 Описание задачи в ключевых абстракциях предметной области**

Объектно-ориентированное программирование представляет собой способ программирования структурного типа, который позволяет создавать модульные программы с предоставлением данных, на определенном уровне абстракции. Оно является естественной эволюцией более ранних нововведений в разработке языков программирования. Объектно–ориентированное программирование является более структурным, чем все предыдущие разработки, касающиеся структурного программирования. Оно также является более модульным и более абстрактным, чем предыдущие попытки абстрагирования данных и переноса деталей программирования на внутренний уровень.

Основным понятием объектно-ориентированного программирования является понятие объекта. Объект – это высший уровень абстракции данных, который объединяет в себе данные и методы. Данными, то есть полями объекта могут быть любые структурные данные, в том числе и другие объекты.

Инкапсуляция – это механизм, который объединяет данные и код, манипулирующий этими данными, а также защищает и то, и другое от внешнего вмешательства или неправильного использования. В объектно-ориентированном программировании код и данные могут быть объединены вместе; в этом случае говорят, что создаётся так называемый “чёрный ящик». Когда коды и данные объединяются таким способом, создаётся объект. Другими словами, объект – это то, что поддерживает инкапсуляцию.

Наследование – это процесс, посредством которого один объект может приобретать свойство другого. Точнее, объект может наследовать основные свойства другого объекта и добавлять к ним черты, характерные только для него. Наследование является важным, поскольку оно позволяет поддерживать концепцию иерархии классов. Применение иерархии классов делает управляемыми большие потоки информации. Без использования иерархии классов, для каждого объекта пришлось бы задать все характеристики, которые бы исчерпывающи его определяли. Однако при использовании наследования можно описать объект путём определения того общего класса, к которому он относится, с теми специальными чертами, которые делают объект уникальным. Наследование играет важную роль в объектно-ориентированном программировании.

Полиморфизм – это свойство, которое позволяет одно и то же имя использовать для решения двух или более схожих, но технически разных задач. Целью полиморфизма, применительно к объектно-ориентированному программированию, является использование одного имени для задания общих для класса действий. Выполнение каждого конкретного действия будет определяться типом данных. В более общем смысле, концепцией полиморфизма является идея один интерфейс для группы близких по смыслу действий. Преимуществом полиморфизма является то, что он помогает снижать сложность программ, разрешая использование того же интерфейса для задания единого класса действий. Выбор же конкретного действия, в зависимости от ситуации, возлагается на компилятор. Программисту, не нужно делать этот выбор самому. Нужно только помнить и использовать общий интерфейс.

UML – язык графического описания для объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля, это – открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML- моделью. UML был создан для определения, визуализации, проектирования, и документирования, в основном программных систем. UML не является языком программирования, но на основании UML- моделей возможна генерация кода.

Язык UML представляет собой общецелевой язык визуального моделирования, который разработан для специфики, визуализации, проектирования и документирования компонентов программного обеспечения, бизнес – процессов и других систем. Язык UML одновременно является простым и мощным средством моделирования, которой может быть эффективно использован для построения концептуальных, логических и графических моделей сложных систем самого различного целевого назначения. Этот язык вобрал в себя наилучшее качества методов программной инженерии, которое с успехом использовались на протяжении последних лет при моделировании больших и сложных систем.

Язык UML основан на некотором числе базовых понятий, которые могут быть изучены и применены большинством программистов и разработчиков, знакомых с методами объектно-ориентированного анализа и проектирования. При этом базовые понятия могут комбинироваться и расширяться таким образом, что специалисты объектного моделирования получают возможность самостоятельно разрабатывать модели больших и сложных систем в самых различных областях приложений.

Конструктивное использование языка UML основывается на понимании общих принципов моделирования сложных систем и особенностей процесса, объектно-ориентированного проектирования в частности. Выбор выразительных средств для построения моделей сложных систем предопределяет те задачи, которые могут быть решены с использованием данных моделей. При этом одним из основных принципов построения моделей сложных систем является принцип абстрагирования, который предписывает включать в модель только те аспекты проектируемой системы, которые имеют непосредственное отношение к выполнению системой своих функций или своего целевого предназначения. При этом все второстепенные детали опускаются, чтобы чрезмерно не усложнять процесс анализа и исследования полученной модели.

Другим принципом построения моделей сложных систем является принцип многомодельности. Этот принцип представляет собой утверждение о том, что никакая единственная модель не может с достаточной степенью адекватности описывать различные аспекты сложной системы. Применительно к методологии объектно-ориентированного проектирования это означает, что достаточно полная модель сложной системы допускает некоторое число взаимосвязанных представлений (views), каждое из которых адекватно отражает некоторый аспект поведения или структуры системы. При этом наиболее общими представлениями сложной системы принято считать статическое и динамическое представления, которые в свою очередь могут подразделяться на другие более частные представления.) феномен сложной системы как раз и состоит в том, что никакое ее единственное представление не является достаточным для адекватного выражения всех особенностей моделируемой системы.

Язык [UML](http://www.znannya.org/?view=concept:1709) предназначен для решения следующих задач:

1. предоставить в распоряжение пользователей легко воспринимаемый и выразительный язык визуального моделирования, специально предназначенный для разработки и документирования моделей сложных систем самого различного целевого назначения. Речь идет о том, что важным фактором дальнейшего развития и повсеместного использования методологии ООАП является интуитивная ясность и понятность основных конструкций соответствующего языка моделирования. Язык [UML](http://www.znannya.org/?view=concept:1709) включает в себя не только абстрактные конструкции для представления метамоделей систем, но и целый ряд конкретных понятий, имеющих вполне определенную семантику. Это позволяет языку UML одновременно достичь не только универсальности представления моделей для самых различных приложений, но и возможности описания достаточно тонких деталей реализации этих моделей применительно к конкретным системам;
2. снабдить исходные понятия языка [UML](http://www.znannya.org/?view=concept:1709) возможностью расширения и специализации для более точного представления моделей систем в конкретной предметной области. Хотя язык UML является формальным языком - спецификаций, формальность его описания отличается от синтаксиса как традиционных формально-логических языков, так и известных языков программирования. Разработчики из [OMG](http://www.znannya.org/?view=concept:1776) предполагают, что язык [UML](http://www.znannya.org/?view=concept:1709) как никакой другой может быть приспособлен для конкретных предметных областей. Это становится возможным по той причине, что в самом описании языка UML заложен механизм расширения базовых понятий, который является самостоятельным элементом языка и имеет собственное описание в форме правил расширения;
3. описание языка UML должно поддерживать такую спецификацию моделей, которая не зависит от конкретных языков программирования и инструментальных средств проектирования программных систем. Речь идет о том, что ни одна из конструкций языка UML не должна зависеть от особенностей ее реализации в известных языках программирования. То есть, хотя отдельные понятия языка UML и связаны с последними очень тесно, их жесткая интерпретация в форме каких бы то ни было конструкций программирования не может быть признана корректной. Другими словами, разработчики из OMG считают необходимым свойством языка UML его контекстно-программную независимость;
4. описание языка UML должно включать в себя семантический базис для понимания общих особенностей ООАП. Говоря об этой особенности, имеют в виду самодостаточность языка UML для понимания не только его базовых конструкций, но что не менее важно - понимания общих принципов ООАП. В этой связи необходимо отметить, что поскольку язык UML не является языком программирования, а служит средством для решения задач объектно-ориентированного моделирования систем, описание языка UML должно по возможности включать в себя все необходимые понятия для ООАП. Без этого свойства язык UML может оказаться бесполезным и невостребованным большинством пользователей, которые не знакомы с проблематикой ООАП сложных систем;
5. поощрять развитие рынка объектных инструментальных средств. Более 800 ведущих производителей программных и аппаратных средств, усилия которых сосредоточены в рамках [OMG](http://www.znannya.org/?view=concept:1776), видят перспективы развития современных информационных технологий и основу своего коммерческого успеха в широком продвижении на рынок инструментальных средств, поддерживающих объектные технологии. Говоря же об объектных технологиях, разработчики из OMG имеют в виду, прежде всего, совокупность технологических решений CORBA и [UML](http://www.znannya.org/?view=concept:1709). С этой точки зрения языку UML отводится роль базового средства для описания и документирования различных объектных [компонентов](http://www.znannya.org/?view=concept:1735) CORBA;
6. способствовать распространению объектных технологий и соответствующих понятий ООАП. Эта задача тесно связана с предыдущей и имеет с ней много общего. Если исключить из рассмотрения рекламные заявления разработчиков об исключительной гибкости и мощности языка [UML](http://www.znannya.org/?view=concept:1709), а попытаться составить объективную картину возможностей этого языка, то можно прийти к следующему заключению. Следует признать, что усилия достаточно большой группы разработчиков были направлены на интеграцию в рамках языка UML многих известных техник визуального моделирования, которые успешно зарекомендовали себя на практике. Хотя это привело к усложнению языка UML по сравнению с известными нотациями структурного системного анализа, платой за сложность являются действительно высокая гибкость и изобразительные возможности уже первых версий языка UML. В свою очередь, использование языка UML для решения всевозможных практических задач будет только способствовать его дальнейшему совершенствованию, а значит и дальнейшему развитию объектных технологий и практики ООАП;
7. интегрировать в себя новейшие и наилучшие достижения практики ООАП. Язык [UML](http://www.znannya.org/?view=concept:1709) непрерывно совершенствуется разработчиками, и основой этой работы является его дальнейшая интеграция с современными модельными технологиями. При этом различные методы системного моделирования получают свое прикладное осмысление в рамках ООАП. В последующем эти методы могут быть включены в состав языка UML в форме дополнительных базовых понятий, наиболее адекватно и полно отражающие наилучшие достижения практики ООАП.

Использование UML не ограничивается моделированием программного обеспечения. Его также используют для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

В UML используются следующие виды диаграмм:

1. структурные диаграммы;
2. диаграммы кооперации;
3. диаграмма поведения;
4. диаграмма взаимодействия.

**3.2 Описание интерфейсов классов**

Для описания интерфейса классов опишем основные процедуры, реализованные в проекте:

* + 1. procedure ImgAdminClick(Sender: TObject) – открытие формы с правами админа;
    2. procedure Timer1Timer(Sender: TObject) – плавное открытие главной формы;
    3. procedure Timer2Timer(Sender: TObject) – после загрузки изображения, показывает все компоненты формы входа;
    4. procedure FormCreate(Sender: TObject) – при создании формы выгружаются изображения для плавного перехода;
    5. procedure TrackBar1Change(Sender: TObject) – происходит смена изображений;
    6. procedure ImgGuestClick(Sender: TObject) – скрывает в главном меню возможность редактирования;
    7. procedure Button2Click(Sender: TObject) – удаление изображения;
    8. procedure Button1Click(Sender: TObject) – изменение изображения;
    9. procedure TForm1.DBGrid1TitleClick(Column: TColumn) – сортировка по нажатию титула столбца, в два порядка первой таблицы;
    10. procedure TForm1.DBGrid2TitleClick(Column: TColumn) – сортировка по нажатию титула столбца, в два порядка второй таблицы;
    11. procedure TForm1.EditV\_MaxKeyPress(Column: TColumn) – запрет ввода символов кроме цифр и точки
    12. procedure TForm1.EditV\_MinKeyPress(Column: TColumn) – запрет ввода символов кроме цифр и точки;
    13. procedure TForm1.EditFastFEnter(Sender: TObject) – быстрый поиск;
    14. procedure TForm1.BtnSaveAddZapisClick(Sender: TObject) – добавление записи;
    15. procedure TForm1.BtnAddCharacClick(Sender: TObject) – добавление характеристики;
    16. procedure Timer1Timer(Sender: TObject) – переход к форме пользователей;
    17. procedure FormClose(Sender: TObject; var Action: TCloseAction) – закрытие предыдущих форм;
    18. procedure HelperClick(Sender: TObject) – выход на раздел помощи программы;
    19. procedure EditFV\_MinKeyPress(Sender: TObject; var Key: Char) – стирает запись и переводит начертание шрифта на обычное;
    20. procedure EditFV\_MaxKeyPress(Sender: TObject; var Key: Char) – стирает запись и переводит начертание шрифта на обычное;
    21. procedure EditV\_MinKeyPress(Sender: TObject; var Key: Char) – стирает запись и переводит начертание шрифта на обычное;
    22. procedure EditV\_MaxKeyPress(Sender: TObject; var Key: Char) – стирает запись и переводит начертание шрифта на обычное;
    23. procedure BtnEdZapisClick(Sender: TObject) – проявляет компоненты редактирования записи;
    24. procedure BtnEdHandZapisClick(Sender: TObject) – включает/выключает ручное редактирования записи;
    25. procedure BtnDelZapisCharacClick(Sender: TObject) – проявляет компоненты удаления записи и его характеристик;
    26. procedure BtnDelCharacClick(Sender: TObject) – проявляет компоненты удаления характеристики;
    27. procedure BtnEdCharacClick(Sender: TObject) – проявляет компоненты редактирования характеристики;
    28. procedure BtnEdHandCharacClick(Sender: TObject) – включает/выключает ручное редактирования записи;
    29. procedure MMDelClick(Sender: TObject) – проявляет компоненты удаления;
    30. procedure MMClearClick(Sender: TObject) – скрывает компоненты на вкладке редактирования;
    31. procedure MySiteClick(Sender: TObject) – открывает браузер и переходит на сайт программы;
    32. procedure BtnAddPDFClick(Sender: TObject) – открывает форму с добавлением PDF;
    33. procedure BtnDelPDFClick(Sender: TObject) – открывает форму с удалением PDF;
    34. procedure ListBox1Click(Sender: TObject) – пролистывание статей в форме помощи;
    35. procedure BtnSaveAddZapisClick(Sender: TObject) – добавление новой записи;
    36. procedure EditSymbolClick(Sender: TObject) – стирает запись и переводит начертание шрифта на обычное;
    37. procedure EditV\_MinClick(Sender: TObject) – стирает запись и переводит начертание шрифта на обычное;
    38. procedure EditV\_MaxClick(Sender: TObject) – стирает запись и переводит начертание шрифта на обычное;
    39. procedure BtnAddCharacClick(Sender: TObject) – добавление новой характеристики;
    40. procedure MMEditClick(Sender: TObject) – проявление компонентов редактирования;
    41. procedure MMAllClick(Sender: TObject) – проявление всех компонентов внесения изменений;
    42. procedure EEditIzmerenieClick(Sender: TObject) – стирает текущие значения4;
    43. procedure BtnEdEditorClick(Sender: TObject) изменяет значения дополнительных величин;

**4 ОПИСАНИЕ ПРОГРАММЫ**

Программное средство функционирует в операционной среде Windows XP и более новых версиях Windows. Общий размер всех используемых файлов составляет более 40 Мб.

Исходный текст программы для создания архива конструктивных чертежей находится в файлах со следующим разрешением:

1. PAS – это текстовый файл, содержащий описание связанным с формой модулем (процедур и функций программной логики функционирования окна приложения в синтаксисе Object Pascal). Для каждой формы, включаемой в проект, создается отдельный модуль;
2. DFM – это файл формы содержит значения свойств формы и ее компонентов. Файл формы связывает графическое представление формы с обработчиками сообщений и используется методом TApplication.CreateForm для начального создания формы;
3. RES – это двоичный файл, содержащий все необходимые для проекта ресурсы, такие как, например, пиктограммы, графические изображения, курсоры мыши или строки. Этот файл создает и модифицирует Delphi. Он не должен изменяться или создаваться пользователем;
4. DCU – это двоичный файл модуля. Имеет одноимённое название с модулем;
5. DPR – это файл, который содержит главный программный блок. Файл проекта подключает все используемые программные модули и содержит операторы для запуска приложения. Этот файл среда Delphi создает и контролирует сама;
6. IDENTCACHE – это кэшированные привязки к файлам;
7. DPROJ.LOCAL – это исходный код проекта (xml);
8. EXE – это откомпилированная программа.

**5 ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ**

**5.1 Цель испытаний**

Тестирование программного обеспечения – это процесс выявления ошибок в программном обеспечении (ПО). Существующие на сегодняшний день методы тестирования ПО не позволяют однозначно и полностью устранить все дефекты и ошибки и установить корректность функционирования анализируемой программы особенно в закрытых частных программах. Поэтому все существующие методы тестирования действуют в рамках формального процесса проверки исследуемого или разрабатываемого ПО.

Такой процесс формальной проверки или верификации может доказать, что дефекты отсутствуют, с точки зрения используемого метода. (То есть нет никакой возможности точно установить или гарантировать отсутствие дефектов в программном продукте с учётом человеческого фактора, присутствующего на всех этапах жизненного цикла ПО).

Существует множество подходов к решению задачи тестирования и верификации ПО, но эффективное тестирование сложных программных продуктов – это процесс в высшей степени творческий, не сводящийся к следованию строгим и чётким процедурам или созданию таковых.

Тестирование проводится с целью обеспечить качество разрабатываемого программного продукта. Стандарт ISO-8402, посвященный описанию систем обеспечения качества программного обеспечения, под качеством понимает «совокупность характеристик программного продукта, относящихся к его способности удовлетворять установленные и предполагаемые потребности клиента». Основным параметром качества программы является надёжность. Надёжность определяется как вероятность его работы без отказов в течении определённого периода времени, рассчитанная с учётом стоимости для пользователя каждого отказа. Отказ программного обеспечения - это проявление ошибки в нём. Отсюда тестирование ПО – это процесс выполнения программы с целью обнаружения в ней ошибок.

Тестирование программного обеспечения – это попытка определить, выполняет ли программа то, что от неё ожидают. Качество программных средств можно определить, как совокупную характеристику исследуемого программного обеспечения с учётом следующих составляющих:

1. [надёжность](http://dic.academic.ru/dic.nsf/ruwiki/7191);
2. с[опровождаемость](http://dic.academic.ru/dic.nsf/ruwiki/139790);
3. практичность;
4. эффективность;
5. мобильность;
6. функциональность.

**5.2 Функциональное тестирование**

При запуске программы появлялось окно, где написано название темы и кто разработал программу, как показано на рисунке 1.

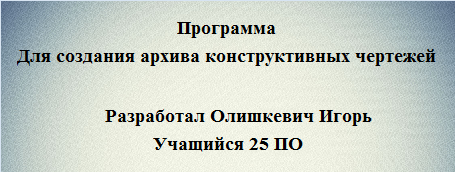


Рисунок 1 – Заставка

После того, как загрузка завершена, открылось окно авторизации, которое представлено на рисунке 2.

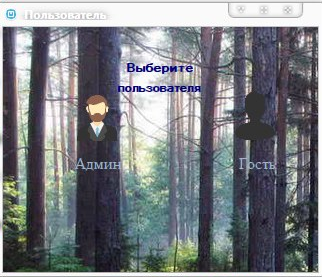


Рисунок 2 – Авторизация

При авторизации пользователь авторизовался как «Гость» или «Админ», в зависимости от пользователя выбрали, что ему приемлемо, и загрузилась форма с данными, как показано на рисунке 3.

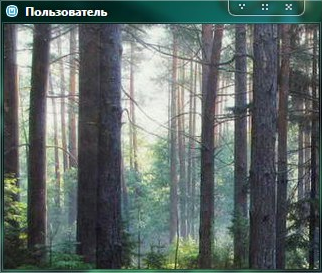


Рисунок 3 – Авторизация пользователя

При успешной авторизации программа перешла в главное меню, которое показано на рисунке 4.

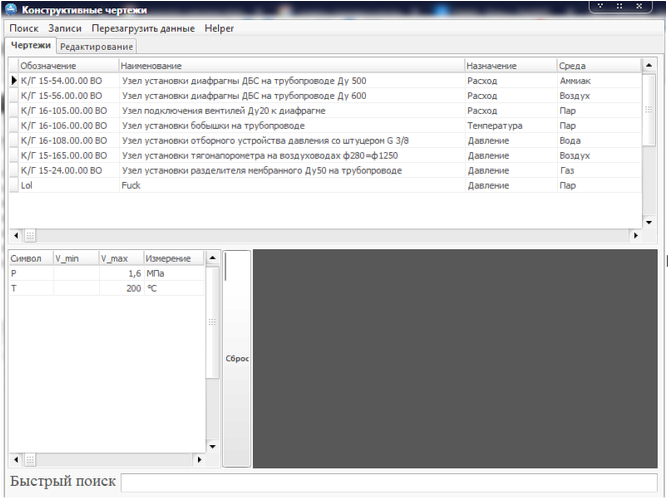


Рисунок 4 – Главное меню

Так же доступна вкладка «Редактирование», с помощью которой можно редактировать данные, то есть добавлять, удалять, изменять записи, меню редактирования показано на рисунке 5.

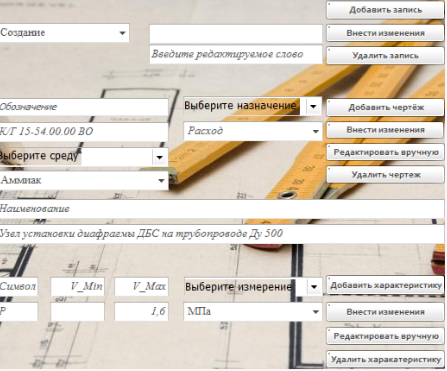


Рисунок 5 – Меню редактирования чертежей

Протестировали редактирование базы данных. Нажали на вкладку редактирование и выбрали добавление записи. В выпадающем списке выбрали один из предоставленных пунктов, в данном случае «Среда». В поле напротив ввели данные, в данном случае «Атмосфера» и нажали на кнопку правее: «Добавить запись», как показано на рисунке 6.



Рисунок 6 – Добавление записи

Следующим редактированием базы данных является добавление чертежа, для этого нажали на редактирование и выбрали «Добавить чертеж», как показано на рисунке 7.

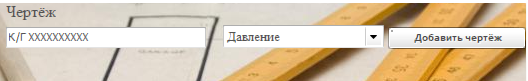


Рисунок 7 – Добавление чертежа

Так же следующим редактированием будет добавление характеристики, как показано на рисунке 8.



Рисунок 8 – Добавление характеристики

Для редактирования данных в главном меню выбрали «Записи» – «Редактировать». Чтобы изменить назначение, среду или измерение необходимо выбрать из выпадающего списка предмет изменения. В поле справа внесли новое название, и нажали на кнопку «Внести изменения», как показано на рисунке 9.

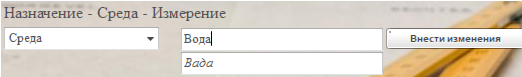


Рисунок 9 – Редактирование записей

Существуют кнопки ручного редактирования, позволяющие редактировать чертежи и характеристики по аналогичным действиям, описанным выше: в поле вносится новое значение (или повторение старого), под ним текущее название. Например, для измерения используются выпадающие списки, где концепция остается той же. Для возвращения в автоматический режим нажимаются те же кнопки (они изменят название на «Обычное редактирование»). В автоматическом редактировании будет изменятся выделенный чертеж/характеристика из соответствующей таблицы, как показано на рисунке 10.

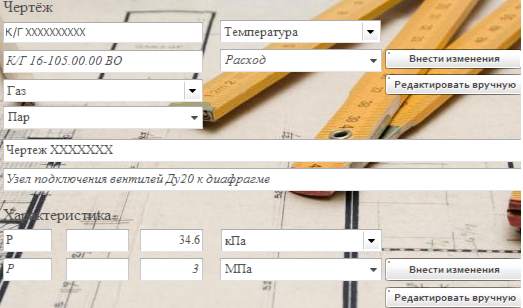


Рисунок 10 – Кнопки редактирования

Для удаления данных в главном меню выбрали «Запись» – «Удаление». Чтобы удалить запись назначения, среды или измерения следует выбрать соответствующий предмет из выпадающего списка и полностью повторить удаляемый объект в поле справа, затем нажать «Удалить запись». Для чертежей проводятся аналогичные действия, где следует повторить обозначение и нажать на кнопку «Удалить чертёж» (Внимание! Удаление чертежа также удалит все его характеристики). Похожим образом, как с записью, происходит удаление характеристики. Необходимо повторить названия полей(необязательным полем для ввода является V\_Min), затем нажать на соответствующую кнопку, как показано на рисунке 11.

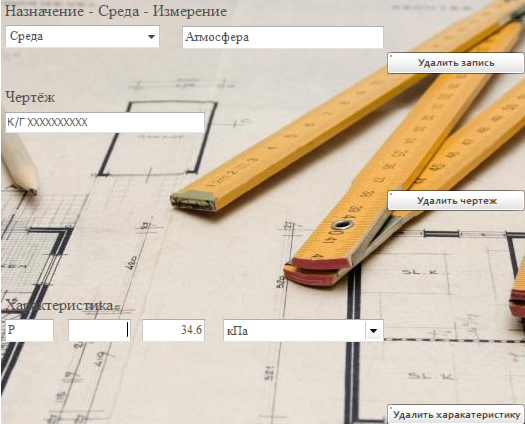


Рисунок 11 – Удаление данных

Применив редактирование в базе данных программы о создании архива конструктивных чертежей, отображается результат добавления в таблице данных о чертежах, как показано на рисунке 12.

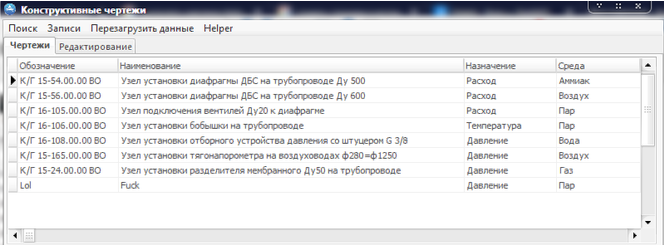


Рисунок 12 – Успешное редактирование

**5.3 Полное тестирование**

В качестве полного тестирования произвели редактирование характеристики. Для этого выбрали пользователя «Админ» и перешли на вкладку «Редактирование», в главном меню нажали «Записи» - «Редактировать» и выделили нужную характеристику при помощи фокусирования нужной записи чертежа в главной таблице, затем произвели то же действие для её величин. Если всё сделано верно, то характеристики отобразятся в нижних полях под наименованием «Характеристика». Нажали на поле с надписью «Символ», после чего надпись будет стерта и курсивное начертание сменится обычным, затем ввели значение. Повторили для поля V\_Min и V\_Max, однако в графу минимальных характеристик ввод данных необязателен. В данном случае внесли значения в графу символа «P», V\_Min оставили пустым, V\_Max задали 3. Затем нажали на стрелочку выпадающего списка с надписью «Выберите измерение», где после выбора значения текст пропал. В данном случае в выпадающем списке назначили «Мпа». Результат действий изображен на рисунке 13.

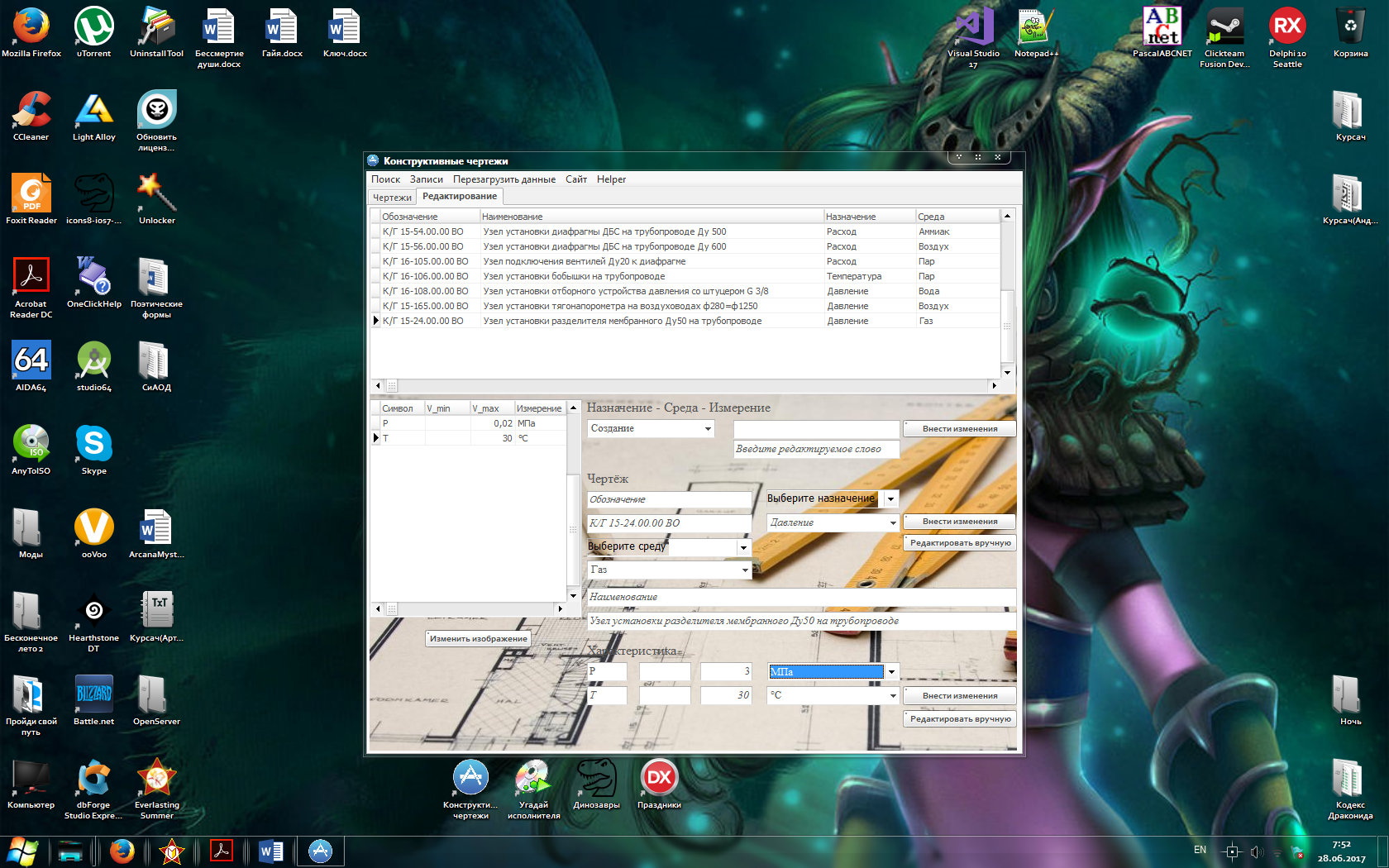


Рисунок 13 – Ввод данных редактирования характеристики

Нажали «Внести изменения». Если все выполнено верно, то в таблице характеристика сменит свои значения на новые. Измененная характеристика представлена на рисунке 14.

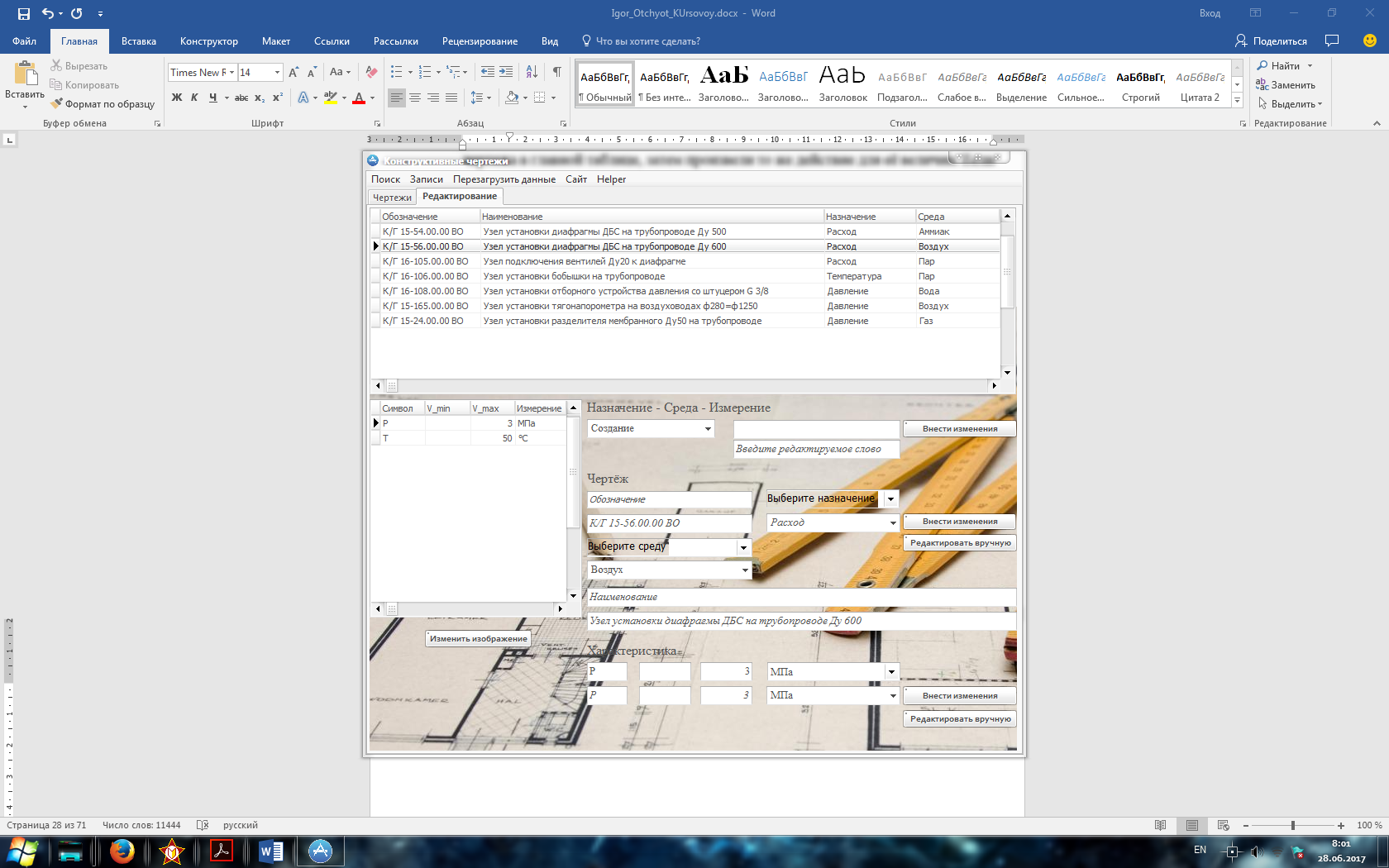


Рисунок 14 – Итог редактирования характеристики

Также существует возможность редактировать вручную по одноименной кнопке. После ее нажатия графы с изменяемой характеристикой можно заполнить вручную. Если такой характеристики не существует или его символа, то будет выдано соответствующее сообщение. Пример ошибки представлен на рисунке 15.

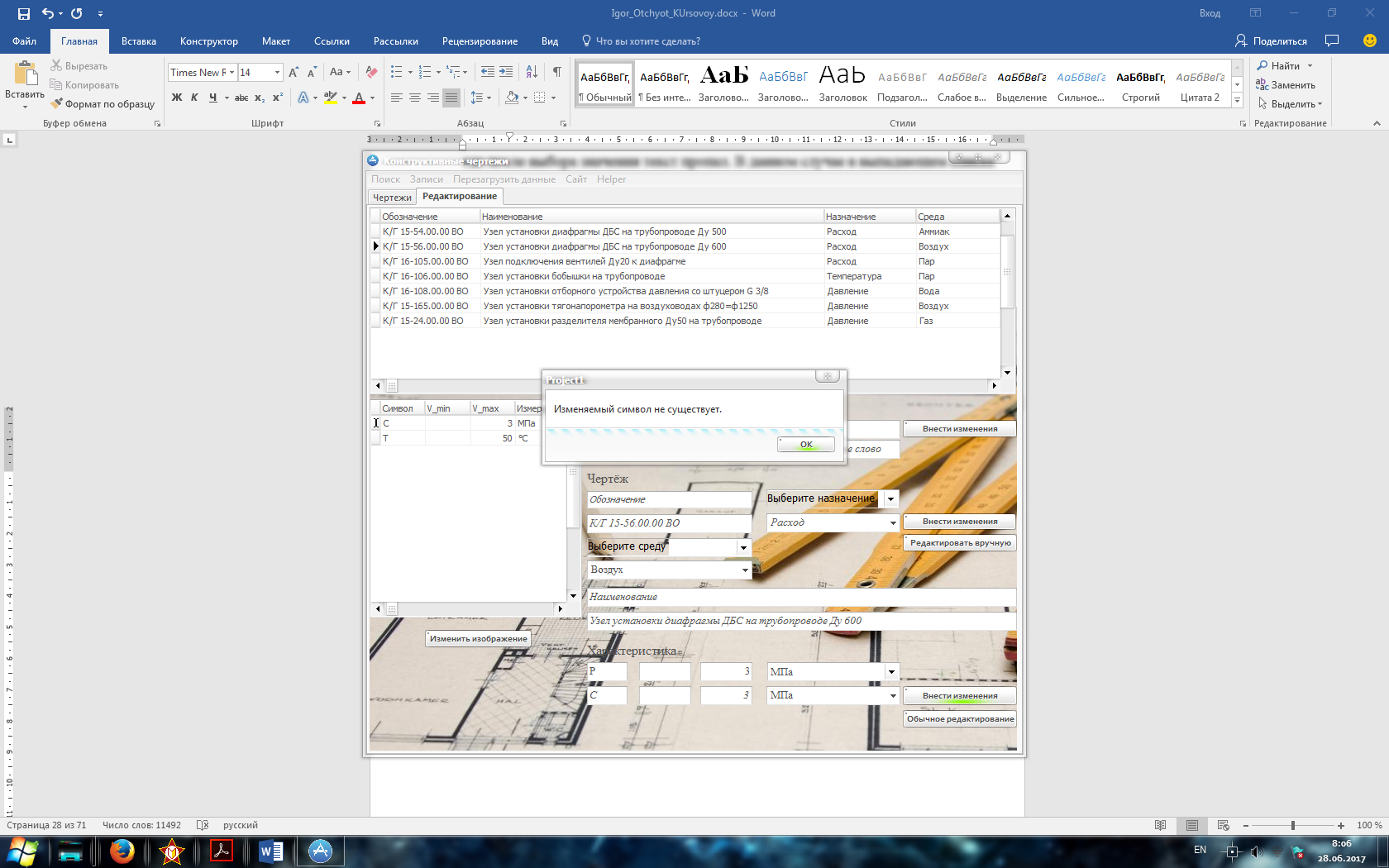


Рисунок 15 – Сообщение об ошибке

**6 ОПИСАНИЕ ПРИМЕНЕНИЯ**

Программа предусматривает собой, что если редактирование базы данных может проходить только под авторизации как «Админ», если же пользователь авторизовался как «Гость», то вкладка на редактирование проекта будет не доступна и «Гость» сможет только проводить поиск по архиву конструкторскими чертежами предложемыми данной программой как показано на рисунке 16.

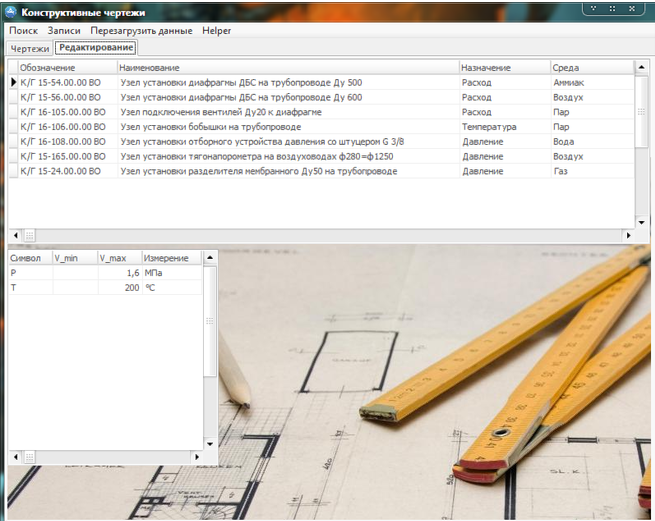


Рисунок 16 – Ограничение для пользователя

Так же «Гость» может отсортировать данные базы по:

1. обозначение;
2. наименование;
3. назначение;
4. среда.

Поиск проводиться по 1 из характеристик, например по обозначению чертежа, как показано на рисунке 17.

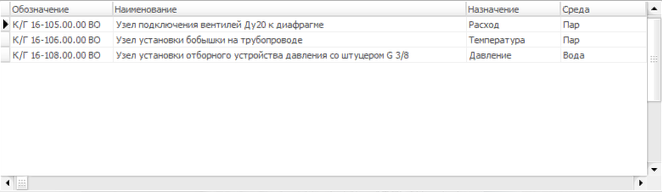


Рисунок 17 – Сортировка данных

В том числе данные можно найти при помощи «Быстрого поиска», где в соответствующую графу достаточно ввести искомые слова, как представлено на изображении 18.

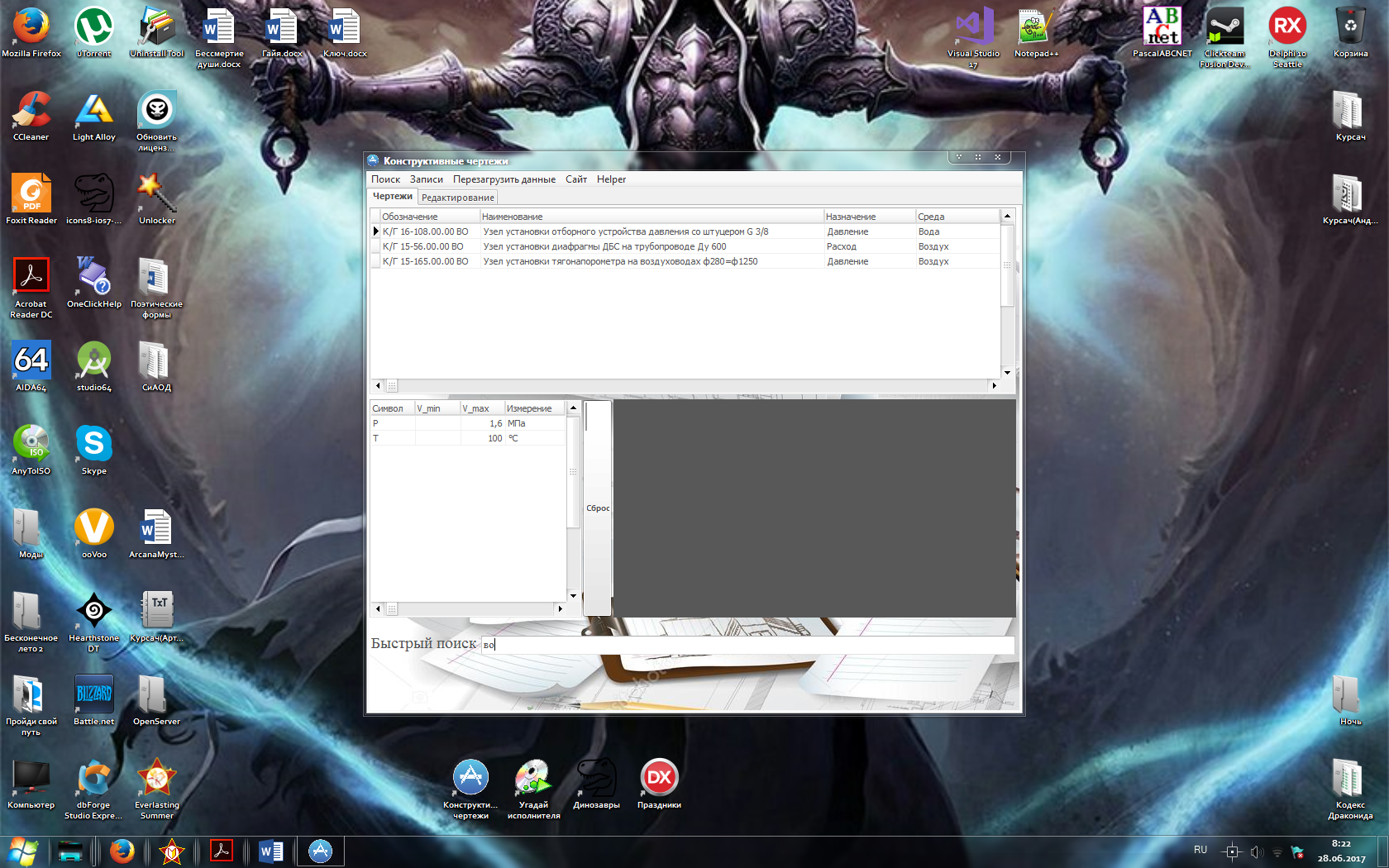


Рисунок 18 – Быстрый поиск

**6.1 Назначение и условия применения программы**

Назначение программы заключается в подборе конструктивных чертежей представленной программой, данная программа может быть использована на сайте если доработать.

**6.2 Инсталляция**

Для установки программы требуется скопировать папку с программой и базу данных на используемый ресурс, после чего настроить базу данных с помощью «Open Server», установить Adobe Acrobat Reader DC последней версии с официального сайта и после всего проделанного у нас подключённый проект.

**6.3 Выполнение программ**

Программа для создания архива конструктивных чертежей представляет собой программу, с помощью которой пользователь может просматривать, редактировать, добавлять, удалять чертежи. Находить нужные по характеристикам данные и многое другое.

**ЗАКЛЮЧЕНИЕ**

Задачей данного проекта являлась разработка программы – создание архива конструктивных чертежей.

Для реализации данного проекта была выбрана операционная система Windows 7 Ultimate. Данная ОС была выбрана из-за своей простоты и удобства использования, а также за обширные возможности. Программа была создана в среде разработки Embarcadero Delphi 10.1 Seattle на языке Delphi. Была использована реляционная СУБД MySQL v. 5.5 в программной сборке OpenSever v. 5.2.2 с включающим себя Apache v. 2.4 и PhP v. 5.5.

В ходе написания курсового проекта были разработаны следующие функции:

* добавление, удаление и редактирование чертежей и характеристик;
* просмотр чертежей и характеристик;
* поиск чертежей по характеристикам и первым символам;
* сортирование по наименованию, обозначению, среде, назначению.

Вся программа неоднократно тестировалась на правильность получаемых результатов, все возникающие ошибки были устранены в ходе работы. После проведенного тестирования можно установить, что программа удовлетворяет всем поставленным перед ней требованиям и является готовым программным средством.

В программу архива конструктивных чертежей можно добавить функции детального поиска, использование других форматов изображений.

Достоинством является, что данный проект является единственным в сфере чертежей азотной промышленности в открытом доступе.

Данный проект полезен в строительстве, машиностроении, проектировании и так далее. Живой пример пользы данной программы заключается в использовании её ведущим главным специалистом в ОАО «ГИАП», позволяющая быстро найти необходимые данные или сам чертеж, что значительно сокращает время и позволяет избежать неточностей в работе.

**ЛИТЕРАТУРА**

1. Архангельский А.Я. «Программирование в Delphi». 2003г. с.1152.
2. Бобровский С.И. «Delphi 7. Учебный курс». 2008г. с.736
3. Баженова И.Ю. «Delphi 7. Самоучитель программиста». 2003г с.448
4. Глушаков С.В., Клевцов А.Л. «Программирование в среде Delphi 7.0». 2003г. с.528
5. Емельянов и др. «Основы программирования на Delphi». 2005г. с.232
6. Керман Митчелл К. Программирование и отладка в Delphi. Учебный курс. 2004г. с.720
7. Рубенкинг Н.Д. Delphi для самоучек. 2007г. с.336
8. Тюкачев Н. и др. Программирование в Delphi для начинающих. 2007г. с.672
9. Фленов М.Е. Библия Delphi. 2011г. с.880 (3-е издание)
10. Хомоненко А., Гофман В. Самоучитель Delphi. 2005г. с.576

**ПРИЛОЖЕНИЕ**

**Текст программы**

unit Unit1;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Vcl.Grids, Vcl.DBGrids,

Data.Win.ADODB, Vcl.StdCtrls, Vcl.ComCtrls, Vcl.DBCtrls, jpeg, Vcl.OleCtnrs,

Vcl.ExtCtrls, ShellApi, Vcl.ExtDlgs, Vcl.OleCtrls, AcroPDFLib\_TLB, DBXMySQL,

Data.FMTBcd, Datasnap.DBClient, Datasnap.Provider, Data.SqlExpr, Vcl.Menus,

Vcl.Mask;

type

TForm1 = class(TForm)

OpenDialog1: TOpenDialog;

OpenPictureDialog1: TOpenPictureDialog;

AcroPDF1: TAcroPDF;

PageControl1: TPageControl;

TabSheet1: TTabSheet;

TabSheet2: TTabSheet;

Button2: TButton;

DBGrid1: TDBGrid;

DBGrid2: TDBGrid;

BtnSaveEditor: TButton;

CBCAll: TButton;

MainMenu1: TMainMenu;

N1: TMenuItem;

N3: TMenuItem;

MMEditor: TMenuItem;

N5: TMenuItem;

BtnDelEditor: TButton;

BtnFound: TButton;

EditFV\_Max: TEdit;

CBFOperMax: TComboBox;

CBC2: TButton;

CBC3: TButton;

BtnFClear: TButton;

N6: TMenuItem;

Label10: TLabel;

EditFastF: TEdit;

SQLConnection1: TSQLConnection;

SQLQuery1: TSQLQuery;

DataSetProvider1: TDataSetProvider;

ClientDataSet1: TClientDataSet;

SQLQuery2: TSQLQuery;

DataSetProvider2: TDataSetProvider;

ClientDataSet2: TClientDataSet;

DataSource2: TDataSource;

EditIzmerenie: TEdit;

EditFV\_Min: TEdit;

CBFOperMin: TComboBox;

SQLQuery3: TSQLQuery;

DataSetProvider3: TDataSetProvider;

ClientDataSet3: TClientDataSet;

DataSource3: TDataSource;

EditFoundMaxID: TEdit;

MMAdd: TMenuItem;

MMEdit: TMenuItem;

MMDel: TMenuItem;

MMAll: TMenuItem;

CBCreate: TComboBox;

LabelZapis: TLabel;

EditOboznach: TEdit;

EditNaim: TEdit;

BtnSaveAddZapis: TButton;

LabelCharac: TLabel;

EditSymbol: TEdit;

EditV\_Min: TEdit;

EditV\_Max: TEdit;

BtnAddCharac: TButton;

BtnEdEditor: TButton;

EEditIzmerenie: TEdit;

Label1: TLabel;

DBEditName: TDBEdit;

BtnEdZapis: TButton;

BtnEdHandZapis: TButton;

DBEditOboznach: TDBEdit;

DBCBNaznach: TDBComboBox;

DBEditSymbol: TDBEdit;

DBEditV\_Min: TDBEdit;

DBEditV\_Max: TDBEdit;

DBCBIzmerenie: TDBComboBox;

BtnDelZapisCharac: TButton;

BtnAddPDF: TButton;

BtnDelPDF: TButton;

DataSource4: TDataSource;

ClientDataSet4: TClientDataSet;

DataSetProvider4: TDataSetProvider;

SQLQuery4: TSQLQuery;

StaticText1: TStaticText;

CBFSymbol: TDBLookupComboBox;

DataSource5: TDataSource;

ClientDataSet5: TClientDataSet;

DataSetProvider5: TDataSetProvider;

SQLQuery5: TSQLQuery;

SQLQuery6: TSQLQuery;

DataSetProvider6: TDataSetProvider;

ClientDataSet6: TClientDataSet;

DataSource6: TDataSource;

SQLQuery7: TSQLQuery;

DataSetProvider7: TDataSetProvider;

ClientDataSet7: TClientDataSet;

DataSource7: TDataSource;

SaveDialog1: TSaveDialog;

CBIzmerenie: TDBLookupComboBox;

CBSreda: TDBLookupComboBox;

CBNaznach: TDBLookupComboBox;

DataSource1: TDataSource;

StaticText2: TStaticText;

StaticText3: TStaticText;

StaticText4: TStaticText;

MMClear: TMenuItem;

SQLQuery8: TSQLQuery;

DataSetProvider8: TDataSetProvider;

ClientDataSet8: TClientDataSet;

DataSource8: TDataSource;

DBCBSreda: TDBComboBox;

BtnEdCharac: TButton;

BtnDelCharac: TButton;

BtnEdHandCharac: TButton;

MySite: TMenuItem;

Image1: TImage;

Image2: TImage;

Helper: TMenuItem;

procedure SQLQuery1AfterScroll(DataSet: TDataSet);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure Button2Click(Sender: TObject);

procedure FormKeyPress(Sender: TObject; var Key: Char);

procedure FormShow(Sender: TObject);

procedure DBGrid1TitleClick(Column: TColumn);

procedure DBGrid2TitleClick(Column: TColumn);

procedure FormCreate(Sender: TObject);

procedure CBCAllClick(Sender: TObject);

procedure BtnSaveEditorClick(Sender: TObject);

procedure MMAddClick(Sender: TObject);

procedure N5Click(Sender: TObject);

procedure BtnDelEditorClick(Sender: TObject);

procedure N3Click(Sender: TObject);

procedure BtnFoundClick(Sender: TObject);

procedure BtnFClearClick(Sender: TObject);

procedure N6Click(Sender: TObject);

procedure EditFastFEnter(Sender: TObject);

procedure EditFV\_MaxClick(Sender: TObject);

procedure EditFV\_MinClick(Sender: TObject);

procedure ImgAddClick(Sender: TObject);

procedure EditNaimClick(Sender: TObject);

procedure EditOboznachClick(Sender: TObject);

procedure BtnSaveAddZapisClick(Sender: TObject);

procedure EditSymbolClick(Sender: TObject);

procedure EditV\_MinClick(Sender: TObject);

procedure EditV\_MaxClick(Sender: TObject);

procedure BtnAddCharacClick(Sender: TObject);

procedure MMEditClick(Sender: TObject);

procedure MMAllClick(Sender: TObject);

procedure EEditIzmerenieClick(Sender: TObject);

procedure BtnEdEditorClick(Sender: TObject);

procedure BtnAddPDFClick(Sender: TObject);

procedure BtnDelPDFClick(Sender: TObject);

procedure CBFSymbolClick(Sender: TObject);

procedure CBSredaClick(Sender: TObject);

procedure CBNaznachClick(Sender: TObject);

procedure CBIzmerenieClick(Sender: TObject);

procedure BtnEdZapisClick(Sender: TObject);

procedure BtnEdHandZapisClick(Sender: TObject);

procedure BtnDelZapisCharacClick(Sender: TObject);

procedure BtnDelCharacClick(Sender: TObject);

procedure BtnEdCharacClick(Sender: TObject);

procedure BtnEdHandCharacClick(Sender: TObject);

procedure MMDelClick(Sender: TObject);

procedure MMClearClick(Sender: TObject);

procedure MySiteClick(Sender: TObject);

procedure EditFV\_MinKeyPress(Sender: TObject; var Key: Char);

procedure EditFV\_MaxKeyPress(Sender: TObject; var Key: Char);

procedure EditV\_MinKeyPress(Sender: TObject; var Key: Char);

procedure EditV\_MaxKeyPress(Sender: TObject; var Key: Char);

procedure HelperClick(Sender: TObject);

private

procedure CreateParams(VAR Params: TCreateParams);

override;

public

{ Public declarations }

end;

var

Form1 : TForm1;

ID\_s : string;

Ref, CBI : Integer;

LUser, SQLAD, SQLFound, FoundAfterScroll : boolean;

usOb, usNa, usCaption, usCaptionC, usMin, usMax, usSymb, usV\_Min, usV\_max : boolean;

implementation

{$R \*.dfm}

uses Unit2, Unit4, Unit7;

procedure TForm1.CreateParams(VAR Params: TCreateParams);

begin

Inherited CreateParams(Params);

WITH Params DO

ExStyle := ExStyle OR WS\_EX\_APPWINDOW;

end;

procedure TForm1.DBGrid1TitleClick(Column: TColumn);

begin

case Column.Index of

1:

begin

ClientDataSet1.Active := false;

SQLQuery1.Close;

SQLQuery1.SQL.Clear;

if SQLAD = true then

begin

SQLQuery1.SQL.Add('SELECT Drawing.ID, Drawing.Designation, Drawing.Name, Conditions.Appointment, Environment.Appointment, Image FROM Drawing, Conditions, Environment ');

SQLQuery1.SQL.Add('WHERE Drawing.ID\_Conditions = Conditions.ID AND Drawing.ID\_Environment = Environment.ID ORDER BY Designation ASC');

SQLAD := false;

end

else

begin

SQLQuery1.SQL.Add('SELECT Drawing.ID, Drawing.Designation, Drawing.Name, Conditions.Appointment, Environment.Appointment, Image FROM Drawing, Conditions, Environment ');

SQLQuery1.SQL.Add('WHERE Drawing.ID\_Conditions = Conditions.ID AND Drawing.ID\_Environment = Environment.ID ORDER BY Designation DESC');

SQLAD := true;

end;

SQLQuery1.Open;

ClientDataSet1.Active := true;

DBGrid1.SetFocus;

end;

3:

begin

ClientDataSet1.Active := false;

SQLQuery1.Close;

SQLQuery1.SQL.Clear;

if SQLAD = true then

begin

SQLQuery1.SQL.Add('SELECT Drawing.ID, Drawing.Designation, Drawing.Name, Conditions.Appointment, Environment.Appointment, Image FROM Drawing, Conditions, Environment ');

SQLQuery1.SQL.Add('WHERE Drawing.ID\_Conditions = Conditions.ID AND Drawing.ID\_Environment = Environment.ID ORDER BY ID\_Conditions ASC');

SQLAD := false;

end

else

begin

SQLQuery1.SQL.Add('SELECT Drawing.ID, Drawing.Designation, Drawing.Name, Conditions.Appointment, Environment.Appointment, Image FROM Drawing, Conditions, Environment ');

SQLQuery1.SQL.Add('WHERE Drawing.ID\_Conditions = Conditions.ID AND Drawing.ID\_Environment = Environment.ID ORDER BY ID\_Conditions DESC');

SQLAD := true;

end;

SQLQuery1.Open;

ClientDataSet1.Active := true;

DBGrid1.SetFocus;

end;

4:

begin

ClientDataSet1.Active := false;

SQLQuery1.Close;

SQLQuery1.SQL.Clear;

if SQLAD = true then

begin

SQLQuery1.SQL.Add('SELECT Drawing.ID, Drawing.Designation, Drawing.Name, Conditions.Appointment, Environment.Appointment, Image FROM Drawing, Conditions, Environment ');

SQLQuery1.SQL.Add('WHERE Drawing.ID\_Conditions = Conditions.ID AND Drawing.ID\_Environment = Environment.ID ORDER BY ID\_Environment ASC');

SQLAD := false;

end

else

begin

SQLQuery1.SQL.Add('SELECT Drawing.ID, Drawing.Designation, Drawing.Name, Conditions.Appointment, Environment.Appointment, Image FROM Drawing, Conditions, Environment ');

SQLQuery1.SQL.Add('WHERE Drawing.ID\_Conditions = Conditions.ID AND Drawing.ID\_Environment = Environment.ID ORDER BY ID\_Environment DESC');

SQLAD := true;

end;

SQLQuery1.Open;

ClientDataSet1.Active := true;

DBGrid1.SetFocus;

end;

end;

end;

procedure TForm1.DBGrid2TitleClick(Column: TColumn);

begin

case Column.Index of

1:

begin

ClientDataSet2.Active := false;

SQLQuery2.Close;

SQLQuery2.SQL.Clear;

if SQLAD = true then

begin

SQLQuery2.SQL.Add('SELECT Characteristics.Id, Characteristics.Symbol, V\_min, V\_max, Designation.Designation FROM Drawing, Characteristics, Designation ');

SQLQuery2.SQL.Add('WHERE Drawing.ID\_Characteristics = Characteristics.ID AND Characteristics.ID\_Designation = Designation.ID ORDER BY ' + Column.FieldName + ' ASC');

SQLAD := false;

end

else

begin

SQLQuery2.SQL.Add('SELECT Characteristics.Id, Characteristics.Symbol, V\_min, V\_max, Designation.Designation FROM Drawing, Characteristics, Designation ');

SQLQuery2.SQL.Add('WHERE Drawing.ID\_Characteristics = Characteristics.ID AND Characteristics.ID\_Designation = Designation.ID ORDER BY ' + Column.FieldName + ' DESC');

SQLAD := true;

end;

SQLQuery2.Open;

ClientDataSet2.Active := true;

DBGrid2.SetFocus;

end;

2:

begin

ClientDataSet2.Active := false;

SQLQuery2.Close;

SQLQuery2.SQL.Clear;

if SQLAD = true then

begin

SQLQuery2.SQL.Add('SELECT Characteristics.Id, Characteristics.Symbol, V\_min, V\_max, Designation.Designation FROM Drawing, Characteristics, Designation ');

SQLQuery2.SQL.Add('WHERE Drawing.ID\_Characteristics = Characteristics.ID AND Characteristics.ID\_Designation = Designation.ID ORDER BY ' + Column.FieldName + ' ASC');

SQLAD := false;

end

else

begin

SQLQuery2.SQL.Add('SELECT Characteristics.Id, Characteristics.Symbol, V\_min, V\_max, Designation.Designation FROM Drawing, Characteristics, Designation ');

SQLQuery2.SQL.Add('WHERE Drawing.ID\_Characteristics = Characteristics.ID AND Characteristics.ID\_Designation = Designation.ID ORDER BY ' + Column.FieldName + ' DESC');

SQLAD := true;

end;

SQLQuery2.Open;

ClientDataSet2.Active := true;

DBGrid2.SetFocus;

end;

3:

begin

ClientDataSet2.Active := false;

SQLQuery2.Close;

SQLQuery2.SQL.Clear;

if SQLAD = true then

begin

SQLQuery2.SQL.Add('SELECT Characteristics.Id, Characteristics.Symbol, V\_min, V\_max, Designation.Designation FROM Drawing, Characteristics, Designation ');

SQLQuery2.SQL.Add('WHERE Drawing.ID\_Characteristics = Characteristics.ID AND Characteristics.ID\_Designation = Designation.ID ORDER BY ID\_Designation ASC');

SQLAD := false;

end

else

begin

SQLQuery2.SQL.Add('SELECT Characteristics.Id, Characteristics.Symbol, V\_min, V\_max, Designation.Designation FROM Drawing, Characteristics, Designation ');

SQLQuery2.SQL.Add('WHERE Drawing.ID\_Characteristics = Characteristics.ID AND Characteristics.ID\_Designation = Designation.ID ORDER BY ID\_Designation DESC');

SQLAD := true;

end;

SQLQuery2.Open;

ClientDataSet2.Active := true;

DBGrid2.SetFocus;

end;

4:

begin

ClientDataSet2.Active := false;

SQLQuery2.Close;

SQLQuery2.SQL.Clear;

if SQLAD = true then

begin

SQLQuery2.SQL.Add('SELECT Characteristics.Id, Symbol, V\_min, V\_max, Designation.Designation FROM Drawing, Characteristics, Designation ');

SQLQuery2.SQL.Add('WHERE Drawing.ID\_Characteristics = Characteristics.ID AND Characteristics.ID\_Designation = Designation.ID ORDER BY ' + Column.FieldName + ' DESC');

SQLAD := false;

end

else

begin

SQLQuery2.SQL.Add('SELECT Characteristics.Id, Symbol, V\_min, V\_max, Designation.Designation FROM Drawing, Characteristics, Designation ');

SQLQuery2.SQL.Add('WHERE Drawing.ID\_Characteristics = Characteristics.ID AND Characteristics.ID\_Designation = Designation.ID ORDER BY ' + Column.FieldName + ' ASC');

SQLAD := true;

end;

SQLQuery2.Open;

ClientDataSet2.Active := true;

DBGrid2.SetFocus;

end;

end;

end;

procedure TForm1.EditOboznachClick(Sender: TObject);

begin

EditOboznach.Font.Style := [];

if usOb <> true then

EditOboznach.Text := '';

usOb := true;

end;

procedure TForm1.EditSymbolClick(Sender: TObject);

begin

EditSymbol.Font.Style := [];

if usSymb <> true then

EditSymbol.Text := '';

usSymb := true;

end;

procedure TForm1.EditV\_MaxClick(Sender: TObject);

begin

EditV\_Max.Font.Style := [];

if usMax <> true then

EditV\_Max.Text := '';

usMax := true;

end;

procedure TForm1.EditV\_MaxKeyPress(Sender: TObject; var Key: Char);

begin

if not(Key in ['0'..'9']) and not(Key in ['.']) then Key:=#0;

end;

procedure TForm1.EditV\_MinClick(Sender: TObject);

begin

EditV\_Min.Font.Style := [];

if usMin <> true then

EditV\_Min.Text := '';

usMin := true;

end;

procedure TForm1.EditV\_MinKeyPress(Sender: TObject; var Key: Char);

begin

if not(Key in ['0'..'9']) and not(Key in ['.']) then Key:=#0;

end;

procedure TForm1.EEditIzmerenieClick(Sender: TObject);

begin

EEditIzmerenie.Text := '';

end;

procedure TForm1.EditFastFEnter(Sender: TObject);

Var buf : string;

begin

buf := EditFastF.Text + '%';

ClientDataSet1.Active := false;

SQLQuery1.Close;

SQLQuery1.SQL.Clear;

SQLQuery1.SQL.Add('SELECT DISTINCT Drawing.ID,Drawing.Designation,Name,Conditions.Appointment,Environment.Appointment FROM Drawing,Conditions,Characteristics,Environment ');

SQLQuery1.SQL.Add('WHERE Drawing.ID\_Conditions = Conditions.ID AND Drawing.ID\_Environment = Environment.ID AND Drawing.ID\_Characteristics = Characteristics.ID ');

SQLQuery1.SQL.Add('AND (Drawing.Designation LIKE «' + buf + '%» OR Name LIKE «' + buf + '%» OR Conditions.Appointment LIKE «' + buf + '%» OR Environment.Appointment LIKE «' + buf + '%»)');

SQLQuery1.Open;

ClientDataSet1.Active := true;

end;

procedure TForm1.EditFV\_MaxClick(Sender: TObject);

begin

if usV\_Max <> true then

begin

EditFV\_Max.Font.Style := [];

EditFV\_Max.Text := '';

end;

usV\_Max := true;

end;

procedure TForm1.EditFV\_MaxKeyPress(Sender: TObject; var Key: Char);

begin

if not(Key in ['0'..'9']) and not(Key in ['.']) then Key:=#0;

end;

procedure TForm1.EditFV\_MinClick(Sender: TObject);

begin

if usV\_Min <> true then

begin

EditFV\_Min.Font.Style := [];

EditFV\_Min.Text := '';

end;

usV\_Min := true;

end;

procedure TForm1.EditFV\_MinKeyPress(Sender: TObject; var Key: Char);

begin

if not(Key in ['0'..'9']) and not(Key in ['.']) then Key:=#0;

end;

procedure TForm1.EditNaimClick(Sender: TObject);

Var us : boolean;

begin

EditNaim.Font.Style := [];

if usNa <> true then

EditNaim.Text := '';

usNa := true;

end;

procedure TForm1.SQLQuery1AfterScroll(DataSet: TDataSet);

var FileName: string;

begin

{if FoundAfterScroll then

begin

ID\_s := DBGrid1.Fields[0].Value;

end; }

{if FoundAfterScroll = true then

begin

ClientDataSet2.Active := false;

ID\_s := DBGrid1.Fields[0].Value;

SQLQuery2.Close;

SQLQuery2.SQL.Clear;

SQLQuery2.SQL.Add('Select Symbol, V\_min, V\_max, Designation.Designation FROM Characteristics, Designation WHERE Characteristics.Designation = Designation.ID AND Characteristics.ID = ' + ID\_s);

SQLQuery2.Open;

ClientDataSet2.Active := true;

end;}

end;

procedure TForm1.BtnSaveAddZapisClick(Sender: TObject);

Var bufID : integer;

bufNaznach, bufSreda : string;

begin

if (EditOboznach.Text = '') or (EditOboznach.Text = 'Обозначение') then

begin

ShowMessage('Введите обозначение.');

exit;

end;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT Designation FROM Drawing WHERE Designation = «' + EditOboznach.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

EditFoundMaxID.Text := VarToStr(SQLQuery3.FieldValues['Designation']);

if EditFoundMaxID.Text = EditOboznach.Text then

begin

ShowMessage('Такой чертеж уже существует.');

Exit;

end;

EditFoundMaxID.Text := '';

if (EditNaim.Text = '') or (EditNaim.Text = 'Наименование') then

begin

ShowMessage('Введите наименование.');

exit;

end;

if (CBNaznach.Text = '') then

begin

ShowMessage('Выберете назначение.');

exit;

end;

if (CBSreda.Text = '') then

begin

ShowMessage('Выберете среду.');

exit;

end;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT ID FROM Conditions WHERE Appointment = «' + CBNaznach.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

bufNaznach := VarToStr(SQLQuery3.FieldValues['ID']);

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT ID FROM Environment WHERE Appointment = «' + CBSreda.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

bufSreda := VarToStr(SQLQuery3.FieldValues['ID']);

bufID := 0;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT MAX(ID) AS ID FROM Drawing');

SQLQuery3.Open;

ClientDataSet3.Active := true;

EditFoundMaxID.Text := IntToStr(SQLQuery3.FieldValues['ID']);

bufID := StrToInt(EditFoundMaxID.Text) + 1;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('INSERT INTO Drawing (ID, Designation, ID\_Characteristics, Name, ID\_Environment, ID\_Conditions) VALUES(');

SQLQuery3.SQL.Add(IntToStr(bufID) + ', «' + EditOboznach.Text + '«, ' + IntToStr(bufID) + ', «' + EditNaim.Text + '«, ');

SQLQuery3.SQL.Add(bufSreda + ', ' + bufNaznach + ')');

SQLQuery3.ExecSQL;

{if OpenDialog1.Execute Then

begin

SQLQuery3.SQL.Text := 'UPDATE Drawing SET Image = :img WHERE ID = ' + IntToStr(bufID);

SQLQuery3.ParamByName('img').LoadFromFile(OpenDialog1.FileName, ftblob);

SQLQuery3.ExecSQL;

end;}

N6Click(Self);

DBGrid1.DataSource.DataSet.Last;

end;

procedure TForm1.BtnSaveEditorClick(Sender: TObject);

Var bufID, i : integer;

begin

if CBCreate.ItemIndex < 0 then

begin

ShowMessage('Выберите создание из списка.');

exit;

end;

if EditIzmerenie.Text = '' then

begin

ShowMessage('Введите новый элемент.');

exit;

end;

bufID := 0;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

if CBCreate.ItemIndex = 0 then

begin

SQLQuery3.SQL.Add('SELECT Appointment FROM Environment WHERE Appointment = «' + EditIzmerenie.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

EditFoundMaxID.Text := VarToStr(SQLQuery3.FieldValues['Appointment']);

if EditFoundMaxID.Text = EditIzmerenie.Text then

begin

ShowMessage('Такое наименование уже существует.');

Exit;

end;

EditFoundMaxID.Text := '';

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT MAX(ID) AS ID FROM Environment');

SQLQuery3.Open;

ClientDataSet3.Active := true;

EditFoundMaxID.Text := IntToStr(SQLQuery3.FieldValues['ID']);

bufID := StrToInt(EditFoundMaxID.Text) + 1;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('INSERT INTO Environment (ID, Appointment) VALUES(' + IntToStr(bufID) + ', «' + EditIzmerenie.Text + '«)');

for i := 0 to DBCBSreda.Items.Count do

if EditIzmerenie.Text <> DBCBSreda.Items.Strings[i-1] then

begin

DBCBSreda.Items.Add(EditIzmerenie.Text);

Break;

end;

end;

if CBCreate.ItemIndex = 1 then

begin

SQLQuery3.SQL.Add('SELECT Appointment FROM Conditions WHERE Appointment = «' + EditIzmerenie.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

EditFoundMaxID.Text := VarToStr(SQLQuery3.FieldValues['Appointment']);

if EditFoundMaxID.Text = EditIzmerenie.Text then

begin

ShowMessage('Такое наименование уже существует.');

Exit;

end;

EditFoundMaxID.Text := '';

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT MAX(ID) AS ID FROM Conditions');

SQLQuery3.Open;

ClientDataSet3.Active := true;

EditFoundMaxID.Text := IntToStr(SQLQuery3.FieldValues['ID']);

bufID := StrToInt(EditFoundMaxID.Text) + 1;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('INSERT INTO Conditions (ID, Appointment) VALUES(' + IntToStr(bufID) + ', «' + EditIzmerenie.Text + '«)');

for i := 0 to DBCBNaznach.Items.Count do

if EditIzmerenie.Text <> DBCBNaznach.Items.Strings[i-1] then

begin

DBCBNaznach.Items.Add(EditIzmerenie.Text);

Break;

end;

end;

if CBCreate.ItemIndex = 2 then

begin

SQLQuery3.SQL.Add('SELECT Designation FROM Designation WHERE Designation = «' + EditIzmerenie.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

EditFoundMaxID.Text := VarToStr(SQLQuery3.FieldValues['Designation']);

if EditFoundMaxID.Text = EditIzmerenie.Text then

begin

ShowMessage('Такое наименование уже существует.');

Exit;

end;

EditFoundMaxID.Text := '';

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT MAX(ID) AS ID FROM Designation');

SQLQuery3.Open;

ClientDataSet3.Active := true;

EditFoundMaxID.Text := IntToStr(SQLQuery3.FieldValues['ID']);

bufID := StrToInt(EditFoundMaxID.Text) + 1;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('INSERT INTO Designation (ID, Designation) VALUES(' + IntToStr(bufID) + ', «' + EditIzmerenie.Text + '«)');

for i := 0 to DBCBIzmerenie.Items.Count do

if EditIzmerenie.Text <> DBCBIzmerenie.Items.Strings[i-1] then

begin

DBCBIzmerenie.Items.Add(EditIzmerenie.Text);

Break;

end;

end;

CBIzmerenie.Refresh;

DBCBIzmerenie.Refresh;

SQLQuery3.ExecSQL;

EEditIzmerenie.Text := EditIzmerenie.Text;

end;

procedure TForm1.BtnAddCharacClick(Sender: TObject);

Var bufID : integer;

bufIzmerenie : string;

dec1,dec2:Extended;

begin

if (EditV\_Min.Text = 'V\_Min') then

begin

ShowMessage('Сотрите надпись «V\_Min».');

exit;

end;

if (EditV\_Max.Text = 'V\_Max') then

begin

ShowMessage('Сотрите надпись «V\_Max» и введите величины.');

exit;

end;

if (EditV\_Max.Text = '') then

begin

ShowMessage('Введите «V\_Max»');

exit;

end;

if (EditSymbol.Text = 'Символ') then

begin

ShowMessage('Сотрите надпись «Символ» и введите величины.');

exit;

end;

if (EditSymbol.Text = '') then

begin

ShowMessage('Введите «Символ»');

exit;

end;

if (CBIzmerenie.Text = '') then

begin

ShowMessage('Выберите еденицу измерения.');

exit;

end;

ID\_s := DBGrid1.Fields[0].Value;

bufID := 0;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT MAX(ID\_) AS ID FROM Characteristics');

SQLQuery3.Open;

ClientDataSet3.Active := true;

EditFoundMaxID.Text := IntToStr(SQLQuery3.FieldValues['ID']);

bufID := StrToInt(EditFoundMaxID.Text) + 1;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT ID FROM Designation WHERE Designation = «' + CBIzmerenie.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

bufIzmerenie := VarToStr(SQLQuery3.FieldValues['ID']);

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('INSERT INTO Characteristics (Characteristics.ID\_, Characteristics.ID, Characteristics.Symbol, ');

if EditV\_Min.Text <> '' then

SQLQuery3.SQL.Add('Characteristics.V\_Min, ');

SQLQuery3.SQL.Add('Characteristics.V\_Max, Characteristics.ID\_Designation) VALUES(');

SQLQuery3.SQL.Add(IntToStr(bufID) + ', ' + ID\_s + ', «' + EditSymbol.Text + '«, ');

if EditV\_Min.Text <> '' then

SQLQuery3.SQL.Add(':dec1, ');

SQLQuery3.SQL.Add(':dec2, ' + bufIzmerenie + ')');

if EditV\_Min.Text <> '' then

SQLQuery3.ParamByName('dec1').asFloat:=StrToFloat(EditV\_Min.Text);

SQLQuery3.ParamByName('dec2').asFloat:=StrToFloat(EditV\_Max.Text);

SQLQuery3.ExecSQL;

N6Click(Self);

DBGrid1.DataSource.DataSet.Locate('ID', ID\_s, [loCaseInsensitive, loPartialKey]);

end;

procedure TForm1.BtnAddPDFClick(Sender: TObject);

Var PicWithAPDF: Tmemorystream;

begin

if FileExists('temp\buf.pdf') then

DeleteFile('temp\buf.pdf');

ID\_s := DBGrid1.Fields[0].Value;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT \* FROM Drawing WHERE ID = ' + ID\_s);

SQLQuery3.Open;

ClientDataSet3.Active := true;

PicWithAPDF := Tmemorystream.Create;

TBLOBField(Form1.SQLQuery3.FieldByName('Image')).SaveToStream(PicWithAPDF);

PicWithAPDF.Position := 0;

PicWithAPDF.SaveToFile('temp\buf.pdf');

PicWithAPDF.SaveToFile('temp\temp.pdf');

FreeAndNil(PicWithAPDF);

if FileExists('temp\buf.pdf') then

FormPDF.acropdf1.LoadFile('temp\buf.pdf');

FormPDF.Show;

FormPDF.Caption := 'Изменение изображения';

FormPDF.Button1.Caption := 'Изменить';

end;

procedure TForm1.BtnDelCharacClick(Sender: TObject);

begin

if (EditSymbol.Text = 'Символ') then

begin

ShowMessage('Сотрите надпись «Символ» и введите величины.');

exit;

end;

if (EditSymbol.Text = '') then

begin

ShowMessage('Введите «Символ»');

exit;

end;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT Symbol FROM Characteristics WHERE ID = ' + ID\_s + ' AND Symbol = «' + EditSymbol.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

EditFoundMaxID.Text := VarToStr(SQLQuery3.FieldValues['Symbol']);

if EditFoundMaxID.Text <> EditSymbol.Text then

begin

ShowMessage('Такая характеристика не существует.');

Exit;

end;

EditFoundMaxID.Text := '';

ID\_s := DBGrid1.Fields[0].Value;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('DELETE FROM Characteristics WHERE ID = ' + ID\_s + ' AND Symbol = «' + EditSymbol.Text + '«');

SQLQuery3.ExecSQL;

ClientDataSet2.Active := false; SQLQuery2.Close;

SQLQuery2.SQL.Clear;

SQLQuery2.SQL.Add('SELECT Characteristics.Id, Characteristics.Symbol, Characteristics.V\_min, Characteristics.V\_max, Designation.Designation');

SQLQuery2.SQL.Add(' FROM Characteristics, Designation WHERE Characteristics.ID\_Designation = Designation.ID');

SQLQuery2.Open;

ClientDataSet2.Active := true;

DataSource2.DataSet.Active := false;

DataSource2.DataSet.Active := true;

DBGrid2.Refresh;

DBGrid1.DataSource.DataSet.Locate('ID', ID\_s, [loCaseInsensitive, loPartialKey]);

end;

procedure TForm1.BtnDelEditorClick(Sender: TObject);

Var i : integer;

begin

if CBCreate.ItemIndex < 0 then

begin

ShowMessage('Выберите удаление из списка.');

exit;

end;

if EditIzmerenie.Text = '' then

begin

ShowMessage('Введите удаляемый элемент.');

exit;

end;

if CBCreate.ItemIndex = 0 then

begin

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT Appointment FROM Environment WHERE Appointment = «' + EEditIzmerenie.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

EditFoundMaxID.Text := VarToStr(SQLQuery3.FieldValues['Appointment']);

if EditFoundMaxID.Text <> EEditIzmerenie.Text then

begin

ShowMessage('Такое наименование не существует.');

Exit;

end;

EditFoundMaxID.Text := '';

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('DELETE FROM Environment WHERE Appointment = «' + EditIzmerenie.Text + '«');

for i := 0 to DBCBSreda.Items.Count do

if EditIzmerenie.Text = DBCBSreda.Items.Strings[i-1] then

DBCBSreda.Items.Delete(i-1);

CBNaznach.Refresh;

DBCBNaznach.Refresh;

end;

if CBCreate.ItemIndex = 1 then

begin

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT Appointment FROM Conditions WHERE Appointment = «' + EEditIzmerenie.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

EditFoundMaxID.Text := VarToStr(SQLQuery3.FieldValues['Appointment']);

if EditFoundMaxID.Text <> EEditIzmerenie.Text then

begin

ShowMessage('Такое наименование не существует.');

Exit;

end;

EditFoundMaxID.Text := '';

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('DELETE FROM Conditions WHERE Appointment = «' + EditIzmerenie.Text + '«');

for i := 0 to DBCBNaznach.Items.Count do

if EditIzmerenie.Text = DBCBNaznach.Items.Strings[i-1] then

DBCBNaznach.Items.Delete(i-1);

CBSreda.Refresh;

DBCBSreda.Refresh;

end;

if CBCreate.ItemIndex = 2 then

begin

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT Designation FROM Designation WHERE Designation = «' + EEditIzmerenie.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

EditFoundMaxID.Text := VarToStr(SQLQuery3.FieldValues['Designation']);

if EditFoundMaxID.Text <> EEditIzmerenie.Text then

begin

ShowMessage('Такое наименование не существует.');

Exit;

end;

EditFoundMaxID.Text := '';

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('DELETE FROM Designation WHERE Designation = «' + EditIzmerenie.Text + '«');

for i := 0 to DBCBIzmerenie.Items.Count do

if EditIzmerenie.Text = DBCBIzmerenie.Items.Strings[i-1] then

DBCBIzmerenie.Items.Delete(i-1);

CBIzmerenie.Refresh;

DBCBIzmerenie.Refresh;

end;

SQLQuery3.ExecSQL;

end;

procedure TForm1.BtnDelPDFClick(Sender: TObject);

Var PicWithAPDF: Tmemorystream;

begin

if FileExists('temp\buf.pdf') then

DeleteFile('temp\buf.pdf');

ID\_s := DBGrid1.Fields[0].Value;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT \* FROM Drawing WHERE ID = ' + ID\_s);

SQLQuery3.Open;

ClientDataSet3.Active := true;

PicWithAPDF := Tmemorystream.Create;

TBLOBField(Form1.SQLQuery3.FieldByName('Image')).SaveToStream(PicWithAPDF);

PicWithAPDF.Position := 0;

PicWithAPDF.SaveToFile('temp\buf.pdf');

PicWithAPDF.SaveToFile('temp\temp.pdf');

FreeAndNil(PicWithAPDF);

if FileExists('temp\buf.pdf') then

FormPDF.acropdf1.LoadFile('temp\buf.pdf');

FormPDF.Show;

FormPDF.Caption := 'Удаление изображения';

FormPDF.Button1.Caption := 'Удалить';

end;

procedure TForm1.BtnFClearClick(Sender: TObject);

begin

EditFV\_Min.Text := '';

EditFV\_Max.Text := '';

StaticText1.Visible := true;

end;

procedure TForm1.BtnFoundClick(Sender: TObject);

begin

ClientDataSet1.Active := false;

SQLQuery1.Close;

SQLQuery1.SQL.Clear;

SQLQuery1.SQL.Add('SELECT DISTINCT Drawing.ID, Drawing.Designation, Drawing.Name, Conditions.Appointment, Environment.Appointment, Image FROM Drawing, Conditions, Environment, Characteristics');

SQLQuery1.SQL.Add(' WHERE Drawing.ID\_Conditions=Conditions.ID AND Drawing.ID\_Environment = Environment.ID AND drawing.ID\_Characteristics = Characteristics.ID');

if (CBFOperMax.ItemIndex > 0) And (EditFV\_Max.Text <> '') And (CBFSymbol.Text <> '') then

SQLQuery1.SQL.Add(' AND Characteristics.Symbol = «' + CBFSymbol.Text +'« AND Characteristics.V\_Max ' + CBFOperMax.Text + ' ' + EditFV\_Max.Text);

if (CBFOperMin.ItemIndex > 0) And (EditFV\_Min.Text <> '') And (CBFSymbol.Text <> '') then

SQLQuery1.SQL.Add(' AND Characteristics.Symbol = «' + CBFSymbol.Text +'« AND Characteristics.V\_Min ' + CBFOperMin.Text + ' ' + EditFV\_Min.Text);

SQLQuery1.Open;

ClientDataSet1.Active := true;

ClientDataSet2.Active := false;

SQLQuery2.Close;

SQLQuery2.SQL.Clear;

SQLQuery2.SQL.Add('SELECT DISTINCT Drawing.ID, Drawing.Name, Conditions.Appointment, Environment.Appointment, Image, ');

SQLQuery2.SQL.Add('Characteristics.Id, Characteristics.Symbol, Characteristics.V\_min, Characteristics.V\_max, Designation.Designation');

SQLQuery2.SQL.Add(' FROM Drawing, Conditions, Environment, Characteristics, Designation WHERE Characteristics.ID\_Designation = Designation.ID');

SQLQuery2.SQL.Add(' AND Drawing.ID\_Conditions=Conditions.ID AND Drawing.ID\_Environment = Environment.ID AND drawing.ID\_Characteristics = Characteristics.ID');

if (CBFOperMax.ItemIndex > 0) And (EditFV\_Max.Text <> '') And (CBFSymbol.Text <> '') then

SQLQuery2.SQL.Add(' AND Characteristics.Symbol = «' + CBFSymbol.Text +'« AND Characteristics.V\_Max ' + CBFOperMax.Text + ' ' + EditFV\_Max.Text);

if (CBFOperMin.ItemIndex > 0) And (EditFV\_Min.Text <> '') And (CBFSymbol.Text <> '') then

SQLQuery2.SQL.Add(' AND Characteristics.Symbol = «' + CBFSymbol.Text +'« AND Characteristics.V\_Min ' + CBFOperMin.Text + ' ' + EditFV\_Min.Text);

SQLQuery2.Open;

ClientDataSet2.Active := true;

end;

procedure TForm1.BtnEdCharacClick(Sender: TObject);

Var i, bufID : integer;

ShMB : boolean;

bufSymbol, bufIzmerenie, bufOboznach, bufName: string;

bufDBOboznach : string;

begin

if (EditSymbol.Text = '') or (EditSymbol.Text = 'Символ') then

begin

ShowMessage('Введите новый символ.');

exit;

end;

if (DBEditOboznach.Text = '') then

begin

ShowMessage('Введите изменяемый символ.');

exit;

end;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT Symbol FROM Characteristics WHERE Symbol = «' + DBEditSymbol.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

bufSymbol := VarToStr(SQLQuery3.FieldValues['Symbol']);

if bufSymbol <> DBEditSymbol.Text then

begin

ShowMessage('Изменяемый символ не существует.');

Exit;

end;

if (EditV\_Min.Text = 'V\_Min') then

begin

ShowMessage('Введите новый V\_Min.');

exit;

end;

if (DBEditV\_Min.Text = '') then

begin

ShowMessage('Введите изменяемый V\_Min.');

exit;

end;

if (EditV\_Max.Text = 'V\_Max') then

begin

ShowMessage('Введите новый V\_Max.');

exit;

end;

if (DBEditV\_Min.Text = '') then

begin

ShowMessage('Введите изменяемый V\_Max.');

exit;

end;

if (CBIzmerenie.Text = '') then

begin

ShowMessage('Выберите новую еденицу измерения.');

exit;

end;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT ID FROM Designation WHERE Designation = «' + DBCBIzmerenie.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

bufIzmerenie := VarToStr(SQLQuery3.FieldValues['ID']);

ID\_s := DBGrid1.Fields[0].Value;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT ID\_ FROM Characteristics WHERE ID\_Designation = ' + bufIzmerenie + ' AND Symbol = «' + DBEditSymbol.Text + '« ');

SQLQuery3.SQL.Add('And V\_Min = ' + DBEditV\_Min.Text + ' AND V\_Max = ' + DBEditV\_Max.Text + ' AND ID = ' + ID\_s);

SQLQuery3.Open;

ClientDataSet3.Active := true;

EditFoundMaxID.Text := '';

bufID := 0;

EditFoundMaxID.Text := VarToStr(SQLQuery3.FieldValues['ID\_']);

if EditFoundMaxID.Text = '' then

begin

ShowMessage('Характеристика не найдена.');

exit;

end;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT ID FROM Designation WHERE Designation = «' + CBIzmerenie.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

bufIzmerenie := VarToStr(SQLQuery3.FieldValues['ID']);

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('UPDATE Characteristics SET ID\_Designation = ' + bufIzmerenie + ', Symbol = «' + EditSymbol.Text + '«, ');

if EditV\_Min.Text = '' then

SQLQuery3.SQL.Add('V\_Min = NULL, V\_Max = ' + EditV\_Max.Text + ' WHERE ID\_ = ' + EditFoundMaxID.Text)

else

SQLQuery3.SQL.Add('V\_Min = ' + EditV\_Min.Text + ', V\_Max = ' + EditV\_Max.Text + ' WHERE ID\_ = ' + EditFoundMaxID.Text);

SQLQuery3.ExecSQL;

N6Click(Self);

DBGrid1.DataSource.DataSet.Locate('ID', ID\_s, [loCaseInsensitive, loPartialKey]);

end;

procedure TForm1.BtnEdEditorClick(Sender: TObject);

Var i : integer;

ShMB : boolean;

begin

if CBCreate.ItemIndex < 0 then

begin

ShowMessage('Выберите редактирование из списка.');

exit;

end;

if EEditIzmerenie.Text = '' then

begin

ShowMessage('Введите редактируемый элемент.');

exit;

end;

if EditIzmerenie.Text = '' then

begin

ShowMessage('Введите измененный элемент.');

exit;

end;

ShMB := false;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

if CBCreate.ItemIndex = 0 then

begin

SQLQuery3.SQL.Add('SELECT Appointment FROM Environment WHERE Appointment = «' + EEditIzmerenie.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

EditFoundMaxID.Text := VarToStr(SQLQuery3.FieldValues['Appointment']);

if EditFoundMaxID.Text <> EEditIzmerenie.Text then

begin

ShowMessage('Такое наименование не существует.');

Exit;

end;

EditFoundMaxID.Text := '';

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('UPDATE Environment SETAppointment = «' + EditIzmerenie.Text + '« ');

SQLQuery3.SQL.Add('WHERE Appointment = «' + EEditIzmerenie.Text + '«');

for i := 0 to DBCBSreda.Items.Count do

if EEditIzmerenie.Text = DBCBSreda.Items.Strings[i-1] then

begin

DBCBSreda.Items.Delete(i-1);

ShMB := false;

break;

end

else

ShMB := true;

if ShMB = true then

begin

ShowMessage('Такого элемента нет.');

exit;

end;

for i := 0 to DBCBSreda.Items.Count do

if EditIzmerenie.Text <> DBCBSreda.Items.Strings[i-1] then

begin

DBCBSreda.Items.Add(EditIzmerenie.Text);

Break;

end;

CBSreda.Refresh;

DBCBSreda.Refresh;

EEditIzmerenie.Text := EditIzmerenie.Text;

end;

if CBCreate.ItemIndex = 1 then

begin

SQLQuery3.SQL.Add('SELECT Appointment FROM Conditions WHERE Appointment = «' + EEditIzmerenie.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

EditFoundMaxID.Text := VarToStr(SQLQuery3.FieldValues['Appointment']);

if EditFoundMaxID.Text <> EEditIzmerenie.Text then

begin

ShowMessage('Такое наименование не существует.');

Exit;

end;

EditFoundMaxID.Text := '';

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('UPDATE Conditions SETAppointment = «' + EditIzmerenie.Text + '« ');

SQLQuery3.SQL.Add('WHERE Appointment = «' + EEditIzmerenie.Text + '«');

for i := 0 to DBCBNaznach.Items.Count do

if EEditIzmerenie.Text = DBCBNaznach.Items.Strings[i-1] then

begin

DBCBNaznach.Items.Delete(i-1);

ShMB := false;

break;

end

else

ShMB := true;

if ShMB = true then

begin

ShowMessage('Такого элемента нет.');

exit;

end;

for i := 0 to DBCBNaznach.Items.Count do

if EditIzmerenie.Text <> DBCBNaznach.Items.Strings[i-1] then

begin

DBCBNaznach.Items.Add(EditIzmerenie.Text);

Break;

end;

CBNaznach.Refresh;

DBCBNaznach.Refresh;

EEditIzmerenie.Text := EditIzmerenie.Text;

end;

if CBCreate.ItemIndex = 2 then

begin

SQLQuery3.SQL.Add('SELECT Designation FROM Designation WHERE Designation = «' + EEditIzmerenie.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

EditFoundMaxID.Text := VarToStr(SQLQuery3.FieldValues['Designation']);

if EditFoundMaxID.Text <> EEditIzmerenie.Text then

begin

ShowMessage('Такое наименование не существует.');

Exit;

end;

EditFoundMaxID.Text := '';

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('UPDATE Designation SETDesignation = «' + EditIzmerenie.Text + '« ');

SQLQuery3.SQL.Add('WHERE Designation = «' + EEditIzmerenie.Text + '«');

for i := 0 to DBCBIzmerenie.Items.Count do

if EEditIzmerenie.Text = DBCBIzmerenie.Items.Strings[i-1] then

begin

DBCBIzmerenie.Items.Delete(i-1);

ShMB := false;

break;

end

else

ShMB := true;

if ShMB = true then

begin

ShowMessage('Такого элемента нет.');

exit;

end;

for i := 0 to DBCBIzmerenie.Items.Count do

if EditIzmerenie.Text <> DBCBNaznach.Items.Strings[i-1] then

begin

DBCBIzmerenie.Items.Add(EditIzmerenie.Text);

Break;

end;

CBIzmerenie.Refresh;

DBCBIzmerenie.Refresh;

EEditIzmerenie.Text := EditIzmerenie.Text;

end;

SQLQuery3.ExecSQL;

end;

procedure TForm1.BtnEdHandCharacClick(Sender: TObject);

begin

if usCaptionC = false then

begin

BtnEdHandCharac.Caption := 'Обычное редактирование';

usCaptionC := true;

DBEditSymbol.ReadOnly := false;

DBEditV\_Min.ReadOnly := false;

DBEditV\_Max.ReadOnly := false;

DBCBIzmerenie.ReadOnly := false;

end

else

begin

BtnEdHandCharac.Caption := 'Редактировать вручную';

usCaptionC := false;

DBEditSymbol.ReadOnly := true;

DBEditV\_Min.ReadOnly := true;

DBEditV\_Max.ReadOnly := true;

DBCBIzmerenie.ReadOnly := true;

end;

end;

procedure TForm1.BtnEdHandZapisClick(Sender: TObject);

begin

if usCaption = false then

begin

BtnEdHandZapis.Caption := 'Обычное редактирование';

usCaption := true;

DBEditOboznach.ReadOnly := false;

DBEditName.ReadOnly := false;

DBCBNaznach.ReadOnly := false;

DBCBSreda.ReadOnly := false;

end

else

begin

BtnEdHandZapis.Caption := 'Редактировать вручную';

usCaption := false;

DBEditOboznach.ReadOnly := true;

DBEditOboznach.DataField := 'Designation';

DBEditOboznach.DataSource := DataSource1;

DBEditName.ReadOnly := true;

DBEditName.DataField := 'Name';

DBEditName.DataSource := DataSource1;

DBCBNaznach.ReadOnly := true;

DBCBNaznach.DataField := 'Appointment';

DBCBNaznach.DataSource := DataSource1;

DBCBSreda.ReadOnly := true;

DBCBSreda.DataField := 'Appointment\_1';

DBCBSreda.DataSource := DataSource1;

end;

end;

procedure TForm1.BtnEdZapisClick(Sender: TObject);

Var i, bufID : integer;

ShMB : boolean;

bufNaznach, bufSreda, bufOboznach, bufName: string;

bufDBOboznach : string;

begin

if (EditOboznach.Text = '') or (EditOboznach.Text = 'Обозначение') then

begin

ShowMessage('Введите новое обозначение.');

exit;

end;

if (DBEditOboznach.Text = '') then

begin

ShowMessage('Введите изменяемое обозначение.');

exit;

end;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT Designation FROM Drawing WHERE Designation = «' + DBEditOboznach.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

bufOboznach := VarToStr(SQLQuery3.FieldValues['Designation']);

if bufOboznach <> DBEditOboznach.Text then

begin

ShowMessage('Изменяемое обозначение не существует.');

Exit;

end;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT Designation FROM Drawing WHERE Designation = «' + EditOboznach.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

bufOboznach := VarToStr(SQLQuery3.FieldValues['Designation']);

if (bufOboznach = EditOboznach.Text) and (EditOboznach.Text <> DBEditOboznach.Text) then

begin

ShowMessage('Такое обозначение уже существует.');

Exit;

end;

if (EditNaim.Text = '') or (EditNaim.Text = 'Наименование') then

begin

ShowMessage('Введите новое наименование.');

exit;

end;

if (DBEditName.Text = '') then

begin

ShowMessage('Введите изменяемое наименование.');

exit;

end;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT Name FROM Drawing WHERE Name = «' + DBEditName.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

bufName := VarToStr(SQLQuery3.FieldValues['Name']);

if bufName <> DBEditName.Text then

begin

ShowMessage('Изменяемое наименование не существует.');

Exit;

end;

if (CBNaznach.Text = '') then

begin

ShowMessage('Выберите новое назначение.');

exit;

end;

if (CBSreda.Text = '') then

begin

ShowMessage('Выберите новую среду.');

exit;

end;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT ID FROM Conditions WHERE Appointment = «' + DBCBNaznach.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

bufNaznach := VarToStr(SQLQuery3.FieldValues['ID']);

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT ID FROM Environment WHERE Appointment = «' + DBCBSreda.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

bufSreda := VarToStr(SQLQuery3.FieldValues['ID']);

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT ID FROM Drawing WHERE Designation = «' + DBEditOboznach.Text + '« AND Name = «' + DBEditName.Text + '« ');

SQLQuery3.SQL.Add('And ID\_Environment = ' + bufSreda + ' AND ID\_Conditions = ' + bufNaznach);

SQLQuery3.Open;

ClientDataSet3.Active := true;

EditFoundMaxID.Text := '';

bufID := 0;

EditFoundMaxID.Text := IntToStr(SQLQuery3.FieldValues['ID']);

if EditFoundMaxID.Text = '' then

begin

ShowMessage('Чертеж не найден.');

exit;

end;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT ID FROM Conditions WHERE Appointment = «' + CBNaznach.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

bufNaznach := VarToStr(SQLQuery3.FieldValues['ID']);

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT ID FROM Environment WHERE Appointment = «' + CBSreda.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

bufSreda := VarToStr(SQLQuery3.FieldValues['ID']);

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('UPDATE Drawing SET Designation = «' + EditOboznach.Text + '«, Name = «' + EditNaim.Text + '«, ');

SQLQuery3.SQL.Add('ID\_Environment = ' + bufSreda + ', ID\_Conditions = ' + bufNaznach);

SQLQuery3.SQL.Add('WHERE ID = ' + EditFoundMaxID.Text);

SQLQuery3.ExecSQL;

N6Click(Self);

DBGrid1.DataSource.DataSet.Locate('ID', EditFoundMaxID.Text, [loCaseInsensitive, loPartialKey]);

end;

procedure TForm1.BtnDelZapisCharacClick(Sender: TObject);

Var bufDesign : string;

begin

if (EditOboznach.Text = '') or (EditOboznach.Text = 'Обозначение') then

begin

ShowMessage('Введите удаляемое обозначение.');

exit;

end;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT Designation FROM Drawing WHERE Designation = «' + EditOboznach.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

EditFoundMaxID.Text := VarToStr(SQLQuery3.FieldValues['Designation']);

if EditFoundMaxID.Text <> EditOboznach.Text then

begin

ShowMessage('Такой чертеж не существует.');

Exit;

end;

EditFoundMaxID.Text := '';

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT ID\_Characteristics FROM Drawing WHERE Designation = «' + EditOboznach.Text + '«');

SQLQuery3.Open;

ClientDataSet3.Active := true;

EditFoundMaxID.Text := VarToStr(SQLQuery3.FieldValues['ID\_Characteristics']);

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('DELETE FROM Characteristics WHERE ID = ' + EditFoundMaxID.Text);

SQLQuery3.ExecSQL;

EditFoundMaxID.Text := '';

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('DELETE FROM Drawing WHERE Designation = «' + EditOboznach.Text + '« ');

SQLQuery3.ExecSQL;

N6Click(Self);

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

acropdf1.LoadFile('DONTEXISTS.pdf');

DBGrid1.SetFocus;

end;

procedure TForm1.CBCAllClick(Sender: TObject);

begin

ClientDataSet1.Active := false;

SQLQuery1.Close;

SQLQuery1.SQL.Clear;

SQLQuery1.SQL.Add('SELECT Drawing.ID, Drawing.Designation, Drawing.Name, Conditions.Appointment, Environment.Appointment, Image FROM Drawing, Conditions, Environment ');

SQLQuery1.SQL.Add('WHERE Drawing.ID\_Conditions = Conditions.ID AND Drawing.ID\_Environment = Environment.ID ORDER BY Designation ASC');

SQLQuery1.Open;

ClientDataSet1.Active := true;

DBGrid1.SetFocus;

end;

procedure TForm1.CBFSymbolClick(Sender: TObject);

begin

StaticText1.Visible := false;

end;

procedure TForm1.CBIzmerenieClick(Sender: TObject);

begin

StaticText4.Visible := false;

end;

procedure TForm1.CBNaznachClick(Sender: TObject);

begin

StaticText3.Visible := false;

end;

procedure TForm1.CBSredaClick(Sender: TObject);

begin

StaticText2.Visible := false;

end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);

begin

if FileExists('temp\temp.pdf') then

DeleteFile('temp\temp.pdf');

if FileExists('temp\buf.pdf') then

DeleteFile('temp\buf.pdf');

Ref := AcroPdf1.ControlInterface.\_AddRef;

AcroPdf1.Src := '';

FreeAndNil(AcroPdf1);

Ref := FormPDF.AcroPdf1.ControlInterface.\_AddRef;

FormPDF.AcroPdf1.Src := '';

FreeAndNil(FormPDF.AcroPdf1);

Form2.Close;

FormPDF.Close;

end;

procedure TForm1.FormCreate(Sender: TObject);

var

I,II :Integer;

begin

II := PageControl1.PageCount;

If II > 0 then

For I := 0 To II - 1 do

PageControl1.Pages[I].TabVisible := False;

PageControl1.ActivePageIndex := 0;

end;

procedure TForm1.FormKeyPress(Sender: TObject; var Key: Char);

Var PicWithAPDF, B: Tmemorystream;

begin

if key = #13 then

begin

if FileExists('temp\temp.pdf') then

DeleteFile('temp\temp.pdf');

ID\_s := DBGrid1.Fields[0].Value;

ClientDataSet3.Active := false;

SQLQuery3.Close;

SQLQuery3.SQL.Clear;

SQLQuery3.SQL.Add('SELECT \* FROM Drawing WHERE ID = ' + ID\_s);

SQLQuery3.Open;

ClientDataSet3.Active := true;

PicWithAPDF := Tmemorystream.Create;

TBLOBField(Form1.SQLQuery3.FieldByName('Image')).SaveToStream(PicWithAPDF);

PicWithAPDF.Position := 0;

PicWithAPDF.SaveToFile('temp\temp.pdf');

FreeAndNil(PicWithAPDF);

acropdf1.LoadFile('temp\temp.pdf');

end;

end;

procedure TForm1.FormShow(Sender: TObject);

begin

SetWindowLong(AcroPDF1.Handle, GWL\_STYLE, GetWindowLong(AcroPDF1.Handle, GWL\_STYLE) or WS\_THICKFRAME);

SetWindowLong(AcroPDF1.Handle, GWL\_EXSTYLE, GetWindowLong(AcroPDF1.Handle, GWL\_EXSTYLE) and not WS\_EX\_CLIENTEDGE);

end;

procedure TForm1.HelperClick(Sender: TObject);

begin

FormHelp.Show;

end;

procedure TForm1.MySiteClick(Sender: TObject);

begin

ShellExecute (Form1.Handle, nil, 'iexplore', 'https://projectstructuraldrawinghelper.nethouse.ru/', nil, SW\_RESTORE);

end;

procedure TForm1.ImgAddClick(Sender: TObject);

begin

DbGrid1.DataSource.DataSet.Insert;

end;

procedure TForm1.MMAddClick(Sender: TObject);

begin

CBCreate.Visible := true;

EditIzmerenie.Visible := true;

BtnSaveEditor.Visible := true;

Label1.Visible := true;

LabelZapis.Visible := true;

EditOboznach.Visible := true;

EditNaim.Visible := true;

StaticText3.Visible := true;

CBNaznach.Visible := true;

StaticText2.Visible := true;

CBSreda.Visible := true;

BtnSaveAddZapis.Visible := true;

LabelCharac.Visible := true;

BtnAddCharac.Visible := true;

StaticText4.Visible := true;

CBIzmerenie.Visible := true;

EditV\_Max.Visible := true;

EditV\_Min.Visible := true;

EditSymbol.Visible := true;

BtnAddPDF.Visible := true;

EEditIzmerenie.Visible := false;

BtnEdEditor.Visible := false;

DBEditOboznach.Visible := false;

DBEditName.Visible := false;

DBCBNaznach.Visible := false;

DBCBSreda.Visible := false;

BtnEdZapis.Visible := false;

BtnEdCharac.Visible := false;

DBCBIzmerenie.Visible := false;

DBEditV\_Max.Visible := false;

DBEditV\_Min.Visible := false;

DBEditSymbol.Visible := false;

BtnEdHandZapis.Visible := false;

BtnEdHandCharac.Visible := false;

BtnDelEditor.Visible := false;

BtnDelZapisCharac.Visible := false;

BtnDelCharac.Visible := false;

BtnDelPDF.Visible := false;

ClientHeight := 685;

end;

procedure TForm1.MMEditClick(Sender: TObject);

begin

DBGrid2.Options := DBGrid2.Options + [dgIndicator];

CBCreate.Visible := true;

EditIzmerenie.Visible := true;

BtnSaveEditor.Visible := true;

Label1.Visible := true;

LabelZapis.Visible := true;

EditOboznach.Visible := true;

EditNaim.Visible := true;

StaticText3.Visible := true;

CBNaznach.Visible := true;

StaticText2.Visible := true;

CBSreda.Visible := true;

BtnSaveAddZapis.Visible := true;

LabelCharac.Visible := true;

BtnAddCharac.Visible := true;

StaticText4.Visible := true;

CBIzmerenie.Visible := true;

EditV\_Max.Visible := true;

EditV\_Min.Visible := true;

EditSymbol.Visible := true;

EEditIzmerenie.Visible := true;

BtnEdEditor.Visible := true;

DBEditOboznach.Visible := true;

DBEditName.Visible := true;

DBCBNaznach.Visible := true;

DBCBSreda.Visible := true;

BtnEdZapis.Visible := true;

BtnEdCharac.Visible := true;

DBCBIzmerenie.Visible := true;

DBEditV\_Max.Visible := true;

DBEditV\_Min.Visible := true;

DBEditSymbol.Visible := true;

BtnEdHandZapis.Visible := true;

BtnEdHandCharac.Visible := true;

BtnAddPDF.Visible := true;

BtnSaveEditor.Visible := false;

BtnDelEditor.Visible := false;

BtnSaveAddZapis.Visible := false;

BtnDelZapisCharac.Visible := false;

BtnAddCharac.Visible := false;

BtnDelCharac.Visible := false;

BtnDelPDF.Visible := false;

ClientHeight := 685;

end;

procedure TForm1.N3Click(Sender: TObject);

begin

StaticText1.Visible := true;

SQLFound := true;

CBFOperMin.Visible := true;

CBFOperMax.Visible := true;

EditFV\_Min.Visible := true;

EditFV\_Max.Visible := true;

CBFSymbol.Visible := true;

BtnFClear.Visible := true;

BtnFound.Visible := true;

ClientHeight := 650;

end;

procedure TForm1.N5Click(Sender: TObject);

begin

StaticText1.Visible := false;

CBFSymbol.Visible := false;

CBFOperMin.Visible := false;

CBFOperMax.Visible := false;

EditFV\_Min.Visible := false;

EditFV\_Max.Visible := false;

BtnFound.Visible := false;

BtnFClear.Visible := false;

EditFV\_Max.Visible := false;

EditFV\_Min.Visible := false;

ClientHeight := 590;

end;

procedure TForm1.N6Click(Sender: TObject);

begin

ClientDataSet1.Active := false;

SQLQuery1.Close;

SQLQuery1.SQL.Clear;

SQLQuery1.SQL.Add('SELECT DISTINCT Drawing.ID, Drawing.Designation, Drawing.Name, Conditions.Appointment, Environment.Appointment, Image FROM Drawing, Conditions, Environment, Characteristics');

SQLQuery1.SQL.Add(' WHERE Drawing.ID\_Conditions=Conditions.ID AND Drawing.ID\_Environment = Environment.ID AND drawing.ID\_Characteristics = Characteristics.ID');

SQLQuery1.Open;

ClientDataSet1.Active := true;

ClientDataSet2.Active := false;

SQLQuery2.Close;

SQLQuery2.SQL.Clear;

SQLQuery2.SQL.Add('SELECT Characteristics.Id, Characteristics.Symbol, Characteristics.V\_min, Characteristics.V\_max, Designation.Designation');

SQLQuery2.SQL.Add(' FROM Characteristics, Designation WHERE Characteristics.ID\_Designation = Designation.ID');

SQLQuery2.Open;

ClientDataSet2.Active := true;

DataSource1.DataSet.Active := false;

DataSource1.DataSet.Active := true;

DBGrid1.Refresh;

DataSource2.DataSet.Active := false;

DataSource2.DataSet.Active := true;

DBGrid2.Refresh;

end;

procedure TForm1.MMDelClick(Sender: TObject);

begin

CBCreate.Visible := true;

EditIzmerenie.Visible := true;

BtnDelEditor.Visible := true;

Label1.Visible := true;

LabelZapis.Visible := true;

EditOboznach.Visible := true;

BtnDelZapisCharac.Visible := true;

LabelCharac.Visible := true;

BtnDelCharac.Visible := true;

StaticText4.Visible := false;

CBIzmerenie.Visible := true;

EditV\_Max.Visible := true;

EditV\_Min.Visible := true;

EditSymbol.Visible := true;

BtnDelPDF.Visible := true;

EEditIzmerenie.Visible := false;

BtnEdEditor.Visible := false;

DBEditOboznach.Visible := false;

DBEditName.Visible := false;

DBCBNaznach.Visible := false;

DBCBSreda.Visible := false;

BtnEdZapis.Visible := false;

BtnEdCharac.Visible := false;

DBCBIzmerenie.Visible := false;

DBEditV\_Max.Visible := false;

DBEditV\_Min.Visible := false;

DBEditSymbol.Visible := false;

BtnEdHandZapis.Visible := false;

BtnEdHandCharac.Visible := false;

BtnSaveEditor.Visible := false;

BtnSaveAddZapis.Visible := false;

BtnAddCharac.Visible := false;

BtnAddPDF.Visible := false;

CBSreda.Visible := false;

StaticText3.Visible := false;

StaticText2.Visible := false;

CBNaznach.Visible := false;

EditNaim.Visible := false;

ClientHeight := 685;

end;

procedure TForm1.MMAllClick(Sender: TObject);

begin

DBGrid2.Options := DBGrid2.Options + [dgIndicator];

CBCreate.Visible := true;

EditIzmerenie.Visible := true;

BtnSaveEditor.Visible := true;

Label1.Visible := true;

LabelZapis.Visible := true;

EditOboznach.Visible := true;

EditNaim.Visible := true;

StaticText3.Visible := true;

CBNaznach.Visible := true;

StaticText2.Visible := true;

CBSreda.Visible := true;

BtnSaveAddZapis.Visible := true;

LabelCharac.Visible := true;

BtnAddCharac.Visible := true;

StaticText4.Visible := true;

CBIzmerenie.Visible := true;

EditV\_Max.Visible := true;

EditV\_Min.Visible := true;

EditSymbol.Visible := true;

EEditIzmerenie.Visible := true;

BtnEdEditor.Visible := true;

DBEditOboznach.Visible := true;

DBEditName.Visible := true;

DBCBNaznach.Visible := true;

DBCBSreda.Visible := true;

BtnEdZapis.Visible := true;

BtnEdCharac.Visible := true;

DBCBIzmerenie.Visible := true;

DBEditV\_Max.Visible := true;

DBEditV\_Min.Visible := true;

DBEditSymbol.Visible := true;

BtnEdHandZapis.Visible := true;

BtnEdHandCharac.Visible := true;

BtnSaveEditor.Visible := true;

BtnDelEditor.Visible := true;

BtnSaveAddZapis.Visible := true;

BtnDelZapisCharac.Visible := true;

BtnAddCharac.Visible := true;

BtnDelCharac.Visible := true;

BtnAddPDF.Visible := true;

BtnDelPDF.Visible := true;

ClientHeight := 685;

end;

procedure TForm1.MMClearClick(Sender: TObject);

begin

DBGrid2.Options := DBGrid2.Options - [dgIndicator];

CBCreate.Visible := false;

EditIzmerenie.Visible := false;

BtnSaveEditor.Visible := false;

Label1.Visible := false;

LabelZapis.Visible := false;

EditOboznach.Visible := false;

EditNaim.Visible := false;

StaticText3.Visible := false;

CBNaznach.Visible := false;

StaticText2.Visible := false;

CBSreda.Visible := false;

BtnSaveAddZapis.Visible := false;

LabelCharac.Visible := false;

BtnAddCharac.Visible := false;

StaticText4.Visible := false;

CBIzmerenie.Visible := false;

EditV\_Max.Visible := false;

EditV\_Min.Visible := false;

EditSymbol.Visible := false;

EEditIzmerenie.Visible := false;

BtnEdEditor.Visible := false;

DBEditOboznach.Visible := false;

DBEditName.Visible := false;

DBCBNaznach.Visible := false;

DBCBSreda.Visible := false;

BtnEdZapis.Visible := false;

BtnEdCharac.Visible := false;

DBCBIzmerenie.Visible := false;

DBEditV\_Max.Visible := false;

DBEditV\_Min.Visible := false;

DBEditSymbol.Visible := false;

BtnEdHandZapis.Visible := false;

BtnEdHandCharac.Visible := false;

BtnSaveEditor.Visible := false;

BtnDelEditor.Visible := false;

BtnSaveAddZapis.Visible := false;

BtnDelZapisCharac.Visible := false;

BtnAddCharac.Visible := false;

BtnDelCharac.Visible := false;

BtnAddPDF.Visible := false;

BtnDelPDF.Visible := false;

ClientHeight := 590;

end;

end.

unit Unit2;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Imaging.pngimage,

Vcl.Imaging.jpeg, Vcl.ExtCtrls, Vcl.ComCtrls;

type

TForm2 = class(TForm)

ImgAdmin: TImage;

LabAdmin: TLabel;

ImgGuest: TImage;

LabGuest: TLabel;

Timer1: TTimer;

Image1: TImage;

Timer2: TTimer;

TrackBar1: TTrackBar;

Label3: TLabel;

Label4: TLabel;

procedure ImgAdminClick(Sender: TObject);

procedure Timer1Timer(Sender: TObject);

procedure Timer2Timer(Sender: TObject);

procedure TrackBar1Change(Sender: TObject);

procedure ImgGuestClick(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure FormShow(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

{ Private declarations }

public

tm: Real;

img1, img2: TBitmap;

function GetNewImage(const map1, map2: TBitmap; const factor: Real): TBitmap;

end;

var

Form2: TForm2;

implementation

{$R \*.dfm}

uses Unit1, Unit5;

procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction);

begin

FormStart.Close;

end;

procedure TForm2.FormCreate(Sender: TObject);

begin

img1 := TBitmap.Create;

img1.LoadFromFile('img1.bmp');

img1.PixelFormat := pf32bit;

img2 := TBitmap.Create;

img2.LoadFromFile('img2.bmp');

img2.PixelFormat := pf32bit;

end;

procedure TForm2.FormShow(Sender: TObject);

begin

TrackBar1Change(TrackBar1);

end;

function TForm2.GetNewImage(const map1, map2: TBitmap; const factor: Real): TBitmap;

type

TMyColor = record

A, B, G, R: Byte;

end;

pRGBA = ^TMyColor;

var

i, j, m, n: Integer;

p, p1, p2: pRGBA;

begin

m := map1.Width;

n := map1.Height;

result := TBitmap.Create;

result.PixelFormat := pf32bit;

result.Width := map1.Width;

result.Height := map1.Height;

for j := 0 to n - 1 do

begin

p := result.ScanLine[j];

p1 := map1.ScanLine[j];

p2 := map2.ScanLine[j];

for i := 0 to m - 1 do

begin

p^.B := Round( ((1 - factor) \* p1^.B) + (factor \* p2^.B) );

p^.G := Round( ((1 - factor) \* p1^.G) + (factor \* p2^.G) );

p^.R := Round( ((1 - factor) \* p1^.R) + (factor \* p2^.R) );

p^.A := Round( ((1 - factor) \* p1^.A) + (factor \* p2^.A) );

inc(p);

inc(p1);

inc(p2);

end;

end;

end;

procedure TForm2.Timer2Timer(Sender: TObject);

begin

trackbar1.Position := trackbar1.Position + 1;

if trackbar1.Position >= 100 then

begin

labAdmin.Visible := true;

labGuest.Visible := true;

label3.Visible := true;

label4.Visible := true;

ImgAdmin.Visible := true;

ImgGuest.Visible := true;

timer2.Enabled := false;

end;

end;

procedure TForm2.TrackBar1Change(Sender: TObject);

var

map: TBitmap;

tm1, tm2: Cardinal;

begin

tm1 := GetTiCkCount;

map := GetNewImage(img1, img2, (TrackBar1.Position / 100));

Image1.Canvas.Draw(0, 0, map);

map.Free;

tm2 := GetTiCkCount;

If tm = 0.0 then

tm := tm2 - tm1

else

tm := (tm + tm2 - tm1) / 2;

end;

procedure TForm2.ImgAdminClick(Sender: TObject);

begin

Form1.AlphaBlend := true;

Form1.AlphaBlendValue := 0;

Timer1.Enabled := true;

Form1.Show;

Unit1.LUser := true;

Form2.Hide;

Form1.DBGrid1.SetFocus;

end;

procedure TForm2.ImgGuestClick(Sender: TObject);

begin

Form1.MMEditor.Visible := false;

Form1.AlphaBlend := true;

Form1.AlphaBlendValue := 0;

Timer1.Enabled := true;

Form1.Show;

Unit1.LUser := false;

Form2.Hide;

Form1.DBGrid1.SetFocus;

Form1.FormCreate(Form1);

end;

procedure TForm2.Timer1Timer(Sender: TObject);

begin

if Form1.AlphaBlendValue < 255 then

Form1.AlphaBlendValue := Form1.AlphaBlendValue + 3

else

Timer1.Enabled := false;

end;

end.

unit Unit4;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.OleCtrls,

AcroPDFLib\_TLB, DBXMySQL, Data.FMTBcd, Data.DB, Datasnap.DBClient,

Datasnap.Provider, Data.SqlExpr;

type

TFormPDF = class(TForm)

AcroPDF1: TAcroPDF;

Button1: TButton;

Button2: TButton;

OpenDialog1: TOpenDialog;

SQLConnection1: TSQLConnection;

SQLQuery1: TSQLQuery;

DataSetProvider1: TDataSetProvider;

ClientDataSet1: TClientDataSet;

DataSource1: TDataSource;

procedure Button2Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

FormPDF: TFormPDF;

implementation

{$R \*.dfm}

uses Unit1;

procedure TFormPDF.Button1Click(Sender: TObject);

Var ID\_s : string;

PicWithAPDF: Tmemorystream;

begin

ID\_s := Form1.DBGrid1.Fields[0].Value;

if Button1.Caption = 'Изменить' then

begin

if OpenDialog1.Execute Then

begin

SQLQuery1.SQL.Text := 'UPDATE Drawing SET Image = :img WHERE ID = ' + ID\_s;

SQLQuery1.ParamByName('img').LoadFromFile(OpenDialog1.FileName, ftblob);

SQLQuery1.ExecSQL;

end;

end

else

begin

SQLQuery1.SQL.Text := 'UPDATE Drawing SET Image = NULL WHERE ID = ' + ID\_s;

SQLQuery1.ExecSQL;

end;

if FileExists('temp\buf.pdf') then

DeleteFile('temp\buf.pdf');

ID\_s := Form1.DBGrid1.Fields[0].Value;

Form1.ClientDataSet3.Active := false;

Form1.SQLQuery3.Close;

Form1.SQLQuery3.SQL.Clear;

Form1.SQLQuery3.SQL.Add('SELECT \* FROM Drawing WHERE ID = ' + ID\_s);

Form1.SQLQuery3.Open;

Form1.ClientDataSet3.Active := true;

PicWithAPDF := Tmemorystream.Create;

TBLOBField(Form1.SQLQuery3.FieldByName('Image')).SaveToStream(PicWithAPDF);

PicWithAPDF.Position := 0;

PicWithAPDF.SaveToFile('temp\buf.pdf');

FreeAndNil(PicWithAPDF);

if FileExists('temp\buf.pdf') then

FormPDF.acropdf1.LoadFile('temp\buf.pdf');

end;

procedure TFormPDF.Button2Click(Sender: TObject);

Var PicWithAPDF: Tmemorystream;

begin

ID\_s := Form1.DBGrid1.Fields[0].Value;

begin

SQLQuery1.SQL.Text := 'UPDATE Drawing SET Image = :img WHERE ID = ' + ID\_s;

SQLQuery1.ParamByName('img').LoadFromFile('temp\temp.pdf', ftblob);

SQLQuery1.ExecSQL;

end;

FormPDF.Hide;

end;

end.

unit Unit5;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.ComCtrls, Vcl.ExtCtrls,

Vcl.Imaging.jpeg, Vcl.StdCtrls;

type

TFormStart = class(TForm)

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Image1: TImage;

Timer1: TTimer;

ProgressBar1: TProgressBar;

procedure Timer1Timer(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

FormStart: TFormStart;

implementation

{$R \*.dfm}

uses Unit2;

procedure TFormStart.Timer1Timer(Sender: TObject);

begin

ProgressBar1.Position := ProgressBar1.Position + 14;

if ProgressBar1.Position >= 100 then

begin

FormStart.Hide;

Form2.Show;

Form2.Timer2.Enabled := true;

Timer1.Enabled := false;

end;

end;

end.

unit Unit7;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Imaging.jpeg,

Vcl.ExtCtrls, Vcl.Imaging.pngimage;

type

TFormHelp = class(TForm)

ListBox1: TListBox;

ScrollBox1: TScrollBox;

Image1: TImage;

Image3: TImage;

Memo3: TMemo;

Memo1: TMemo;

ScrollBox2: TScrollBox;

Label7: TLabel;

Image6: TImage;

Image7: TImage;

Memo2: TMemo;

Memo4: TMemo;

ScrollBox3: TScrollBox;

Label11: TLabel;

Image9: TImage;

Memo5: TMemo;

Memo6: TMemo;

Image2: TImage;

Memo7: TMemo;

Image4: TImage;

ScrollBox4: TScrollBox;

Label14: TLabel;

Image10: TImage;

Memo8: TMemo;

ScrollBox5: TScrollBox;

Label16: TLabel;

Image11: TImage;

Memo9: TMemo;

Image5: TImage;

Memo10: TMemo;

Image8: TImage;

Memo11: TMemo;

ScrollBox6: TScrollBox;

Label20: TLabel;

Image12: TImage;

Memo12: TMemo;

ScrollBox7: TScrollBox;

Label21: TLabel;

Image13: TImage;

Memo13: TMemo;

Image14: TImage;

ScrollBox8: TScrollBox;

Label1: TLabel;

Image15: TImage;

Memo14: TMemo;

ScrollBox9: TScrollBox;

Label22: TLabel;

Memo15: TMemo;

ScrollBox10: TScrollBox;

Memo16: TMemo;

procedure ListBox1Click(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

private

{ Private declarations }

public

{ Public declarations }

end;

var

FormHelp: TFormHelp;

implementation

{$R \*.dfm}

uses Unit1;

procedure TFormHelp.FormClose(Sender: TObject; var Action: TCloseAction);

begin

Form1.Close;

end;

procedure TFormHelp.ListBox1Click(Sender: TObject);

begin

case ListBox1.ItemIndex of

0: begin

formHelp.ScrollBox1.Visible := true;

formHelp.ScrollBox2.Visible := false;

formHelp.ScrollBox3.Visible := false;

formHelp.ScrollBox4.Visible := false;

formHelp.ScrollBox5.Visible := false;

formHelp.ScrollBox6.Visible := false;

formHelp.ScrollBox7.Visible := false;

formHelp.ScrollBox8.Visible := false;

formHelp.ScrollBox9.Visible := false;

formHelp.ScrollBox10.Visible := false;

end;

1: begin

formHelp.ScrollBox1.Visible := false;

formHelp.ScrollBox2.Visible := true;

formHelp.ScrollBox3.Visible := false;

formHelp.ScrollBox4.Visible := false;

formHelp.ScrollBox5.Visible := false;

formHelp.ScrollBox6.Visible := false;

formHelp.ScrollBox7.Visible := false;

formHelp.ScrollBox8.Visible := false;

formHelp.ScrollBox9.Visible := false;

formHelp.ScrollBox10.Visible := false;

end;

2: begin

formHelp.ScrollBox1.Visible := false;

formHelp.ScrollBox2.Visible := false;

formHelp.ScrollBox3.Visible := true;

formHelp.ScrollBox4.Visible := false;

formHelp.ScrollBox5.Visible := false;

formHelp.ScrollBox6.Visible := false;

formHelp.ScrollBox7.Visible := false;

formHelp.ScrollBox8.Visible := false;

formHelp.ScrollBox9.Visible := false;

formHelp.ScrollBox10.Visible := false;

end;

3: begin

formHelp.ScrollBox1.Visible := false;

formHelp.ScrollBox2.Visible := false;

formHelp.ScrollBox3.Visible := false;

formHelp.ScrollBox4.Visible := true;

formHelp.ScrollBox5.Visible := false;

formHelp.ScrollBox6.Visible := false;

formHelp.ScrollBox7.Visible := false;

formHelp.ScrollBox8.Visible := false;

formHelp.ScrollBox9.Visible := false;

formHelp.ScrollBox10.Visible := false;

end;

4: begin

formHelp.ScrollBox1.Visible := false;

formHelp.ScrollBox2.Visible := false;

formHelp.ScrollBox3.Visible := false;

formHelp.ScrollBox4.Visible := false;

formHelp.ScrollBox5.Visible := true;

formHelp.ScrollBox6.Visible := false;

formHelp.ScrollBox7.Visible := false;

formHelp.ScrollBox8.Visible := false;

formHelp.ScrollBox9.Visible := false;

formHelp.ScrollBox10.Visible := false;

end;

5: begin

formHelp.ScrollBox1.Visible := false;

formHelp.ScrollBox2.Visible := false;

formHelp.ScrollBox3.Visible := false;

formHelp.ScrollBox4.Visible := false;

formHelp.ScrollBox5.Visible := false;

formHelp.ScrollBox6.Visible := true;

formHelp.ScrollBox7.Visible := false;

formHelp.ScrollBox8.Visible := false;

formHelp.ScrollBox9.Visible := false;

formHelp.ScrollBox10.Visible := false;

end;

6: begin

formHelp.ScrollBox1.Visible := false;

formHelp.ScrollBox2.Visible := false;

formHelp.ScrollBox3.Visible := false;

formHelp.ScrollBox4.Visible := false;

formHelp.ScrollBox5.Visible := false;

formHelp.ScrollBox6.Visible := false;

formHelp.ScrollBox7.Visible := true;

formHelp.ScrollBox8.Visible := false;

formHelp.ScrollBox9.Visible := false;

formHelp.ScrollBox10.Visible := false;

end;

7: begin

formHelp.ScrollBox1.Visible := false;

formHelp.ScrollBox2.Visible := false;

formHelp.ScrollBox3.Visible := false;

formHelp.ScrollBox4.Visible := false;

formHelp.ScrollBox5.Visible := false;

formHelp.ScrollBox6.Visible := false;

formHelp.ScrollBox7.Visible := false;

formHelp.ScrollBox8.Visible := true;

formHelp.ScrollBox9.Visible := false;

formHelp.ScrollBox10.Visible := false;

end;

8: begin

formHelp.ScrollBox1.Visible := false;

formHelp.ScrollBox2.Visible := false;

formHelp.ScrollBox3.Visible := false;

formHelp.ScrollBox4.Visible := false;

formHelp.ScrollBox5.Visible := false;

formHelp.ScrollBox6.Visible := false;

formHelp.ScrollBox7.Visible := false;

formHelp.ScrollBox8.Visible := false;

formHelp.ScrollBox9.Visible := true;

formHelp.ScrollBox10.Visible := false;

end;

end;

end;

end.