ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК

КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ

Разработка программного обеспечения формирования и ведения базы данных на основе PostgreSQL для системы недропользования

Выпускная квалификационная работа
обучающегося по направлению подготовки
02.03.02 «Фундаментальная информатика и информационные технологии»
очной формы обучения,
группы 07001301
Никулина Владислава Ивановича

Научный руководитель
Кандидат технических наук,
Доцент Васильев П.В.

БЕЛГОРОД 2017

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Данная выпускная квалификационная работа посвящена созданию базы данных для системы недропользования.

С началом освоения богатств земных недр возникает самый распространенный вид пользования недрами — добыча полезных ископаемых. Добыче полезных ископаемых предшествуют их поиски и разведка (геологическое изучение), необходимые для определения запасов, качества руд, выбора способов разработки месторождений. Использование недр Земли для строительства и эксплуатации подземных сооружений (в том числе образования особо охраняемых геологических объектов, сбора геологических коллекционных материалов и другое) также является видом пользования недрами.

Термин «недропользование» используется всеми учеными, занимающимися проблемами изучения недр. Однако само понятие этого термина трактуется по-разному. С.А. Дзейтов определяет недропользование как деятельность недропользователей в рамках полученной лицензии или заключенного соглашения о разделе продукции. С.В. Гудков определяет недропользование как предусмотренную и защищаемую законом деятельность пользователя недр, осуществляемую на территории Российской Федерации или на территориях, находящихся под ее юрисдикцией, направленную в соответствии с целевым назначением вида недропользования на использование полезных свойств конкретного участка недр для изучения, разведки, добычи или использования иным образом содержащихся в них ресурсов, включая полезные ископаемые. [21]

Целью выпускной квалификационной работы является создание программного обеспечения формирования и ведения базы данных на основе СУБД PostgreSQL.

В ходе выполнения выпускной квалификационной работы необходимо решить следующие задачи:

1. Рассмотреть предметную область.
2. Изучить особенности СУБД PostgreSQL.
3. Спроектировать и разработать программное обеспечение формирования и ведения базы данных для системы недропользования.
4. Проверить работоспособность приложения.

Структура выпускной квалификационной работы. Выпускная квалификационная работа состоит из введения, 3 глав, заключения и приложения, в которое входит список используемой литературы и листинги программных средств.

В первой главе дается теоретический обзор предметной области, обоснование выбора СУБД, сравнение выбранной СУБД с действующими аналогами, обзор выбранных средств разработки.

Во второй главе рассматривается проектирование базы данных.

В третьей главе проводится разработка и тестирование программного приложения на основе СУБД PostgreSQL.

В заключении подводятся итоги по результатам работы.

Выпускная квалификационная работа состоит из 30 листов, включает в себя 2 таблицы, 18 рисунков, 21 источников литературы и 1 приложение.

ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.

1.1 Общие сведения о СУБД PostgreSQL.

PostgreSQL — это объектно-реляционная система управления базами данных (ОРСУБД, ORDBMS), основанная на POSTGRES, Version 4.2 — программе, разработанной на факультете компьютерных наук Калифорнийского университета в Беркли. В POSTGRES появилось множество новшеств, которые были реализованы в некоторых коммерческих СУБД гораздо позднее. [13]

Объектно-реляционная система управления базами данных, именуемая сегодня PostgreSQL, произошла от пакета POSTGRES, написанного в Беркли, Калифорнийском университете. После двух десятилетий разработки PostgreSQL стал самой развитой СУБД с открытым исходным кодом.

Проект POSTGRES, возглавляемый профессором Майклом Стоунбрейкером, спонсировали агентство DARPA при Минобороны США, Управление военных исследований (ARO), Национальный Научный Фонд (NSF) и компания ESL, Inc. Реализация POSTGRES началась в 1986 г.

Первоначальные концепции системы были представлены в документе ston86, а описание первой модели данных появилось в rowe87. Проект системы правил тогда был представлен в ston87a. Суть и архитектура менеджера хранилища были расписаны в ston87b.

С тех пор POSTGRES прошёл несколько этапов развития. Первая «демоверсия» заработала в 1987 и была показана в 1988 на конференции ACM-SIGMOD. Версия 1, описанная в ston90a, была выпущена для нескольких внешних пользователей в июне 1989. В ответ на критику первой системы правил (ston89), она была переделана (ston90b), и в версии 2, выпущенной в июне 1990, была уже новая система правил. В 1991 вышла версия 3, в которой

появилась поддержка различных менеджеров хранилища, улучшенный исполнитель запросов и переписанная система правил.

Последующие выпуски до Postgres95 в основном были направлены на улучшение портируемости и надёжности.

В 1994 Эндри Ю и Джолли Чен добавили в POSTGRES интерпретатор языка SQL. Уже с новым именем Postgres95 был опубликован в Интернете и начал свой путь как потомок разработанного в Беркли POSTGRES, с открытым исходным кодом. Postgres95 версии 1.0.x работал примерно на 30-50% быстрее POSTGRES версии 4.2 (по тестам Wisconsin Benchmark). Помимо исправления ошибок, произошли следующие изменения:

• На смену языку запросов PostQUEL пришёл SQL (реализованный в сервере). (Интерфейсная библиотека libpq унаследовала своё имя от PostQUEL.) Подзапросы не поддерживались до выхода PostgreSQL, хотя их можно было имитировать в Postgres95 с помощью пользовательских функций SQL. Были заново реализованы агрегатные функции. Также появилась поддержка предложения

• Был усовершенствован интерфейс для работы с большими объектами. Единственным механизмом хранения таких данных стали инверсионные объекты. (Инверсионная файловая система была удалена.)

• Удалена система правил на уровне экземпляров; перезаписывающие правила сохранились.

• С исходным кодом стали распространяться краткие описания возможностей стандартного SQL, а также самого Postgres95.

• Для сборки использовался GNU make (вместо BSD make). Кроме того, стало возможно скомпилировать Postgres95 с немодифицированной версией GCC.

В 1996 г. стало понятно, что имя «Postgres95» не выдержит испытание временем. Было выбрано новое имя, PostgreSQL, отражающее связь между оригинальным POSTGRES и более поздними версиями с поддержкой SQL

В процессе разработки Postgres95 основными задачами были поиск и понимание существующих проблем в серверном коде. С переходом к PostgreSQL акценты сместились к реализации новых функций и возможностей, хотя работа продолжается во всех направлениях. [19]

1.2 Сравнение СУБД PostgreSQL с аналогами.

СУБД PostgreSQL предоставляет большой объем как для самой базы данных, так и для ее элементов. Ограничение объемов таблиц и их элементов в различных СУБД представлено ниже в таблице 1.1. [3]

Таблица 1.1 Ограничение объемов таблиц и их элементов

СУБД	Макс. объем БД	Макс. объем таблицы	Макс. размер строки	Макс. кол-во столбцов	Макс Blob/Clob размер
DB2	512 TB	512 TB	32,677 B	1012	2 GB
Firebird	Неограниченно	32 TB	65,536 B	Зависит от исп. типов.	2 GB
Microsoft Access	2 GB	2 GB	16 MB	255	64 KB
MySQL	Неограниченно	2 GB	64 KB	4096	4 GB
Oracle	Неограниченно	4 GB	Неограниченно	1000	Неограниченно
PostgreSQL	Неограниченно	32 TB	1.6 TB	250	1 GB
SQLite	32 TB	-	-	2000	1 GB

Также PostgreSQL может похвастаться значительным объемом, выделенным под основные типы данных. Для сравнения, в таблице 1.2 приведено ограничение длин типов данных в различных СУБД. [11]

Таблица 1.2 Ограничение длин типов данных

СУБД	Макс. длина типа CHAR	Макс длина типа NUMBER	Мин. значение типа DATE	Макс. значение типа DATE
DB2	32 кб	64 бита	0001	9999
Firebird	32,767 б	64 бита	100	32768
Microsoft Access	255 б	32 бита	-	-
MySQL	64 кб	64 бита	1000	9999
Oracle	4000 б	126 бита	-4712	9999
PostgreSQL	1 гб	Неограниченно	-4713	5874897
SQLite	1 гб	64 бита	Нет типа «Дата»	Нет типа «Дата»

По функциональности СУБД PostgreSQL является одной из лучших в мире. PostgreSQL имеет следующие характеристики:

1. Поддержка обширного списка типов данных, включая как стандартные (числовые, текстовые, с плавающей точкой, логические), так и более редкие, например, денежные, геометрические, xml, json и другие.
2. Возможность использования многомерных массивов. Так как PostgreSQL – объектно-реляционная система управления базами данных, она позволяет осуществлять хранение многомерных массивов. [13]
3. Возможность создания нового типа данных с помощью встроенных команд.
4. PostgreSQL соответствует требованиям ACID (атомарность, согласованность, изолированность, надежность). [12]
5. В PostgreSQL осуществляется поддержка пользовательских функций и временных таблиц.

1.3 Выбранные средства разработки

Причины выбора СУБД PostgreSQL. PostgreSQL – объектно-реляционная СУБД, что дает ей определенные преимущества по сравнению с другими СУБД с открытым кодом, такими как MySQL или Firebird.

Отличительная особенность объектно-реляционной базы данных — это возможность поддержки пользовательских объектов и их поведения, включая типы данных, функции, операции, домены и индексы. Эти характеристики делают PostgreSQL более гибким и надежным. Также, в PostgreSQL есть возможность хранить и извлекать сложные структуры данных.

PostgreSQL поддерживает множество типов данных. Помимо integer, float, text, bool и других стандартных типов данных (а также множества их вариаций), PostgreSQL также осуществляет поддержку таких типов, как uuid, денежного, перечисляемого, геометрического, бинарного типов, сетевых адресов, битовых строк, текстового поиска, xml, json, массивов, композитных типов и диапазонов, а также некоторых внутренних типов для идентификации объектов и местоположения логов. [19]

Причины выбора операционной системы Windows 10. Windows 10 – новейшая на данный момент операционная система от компании Microsoft, релиз которой состоялся 29 июля 2015 года. Windows 10 создана чтобы стать единой платформой для всех устройств: для настольных компьютеров, планшетов, смартфонов, консоли Xbox. Интересной особенностью данного продукта является то, что он последний в линейке Windows. На нём нумерация прекращается, а обновления будут устанавливаться только для конкретных программ. [4]

Основными плюсами Windows 10 являются:

– возможность бесплатного перехода с Windows 7, Windows 8 в течение года после официального запуска.

– в интерфейс операционной системы содержит наилучшие достижения предыдущих выпусков Windows, поскольку обладает функционалом Windows 8 и удобством Windows 7.

– существует поддержка нескольких рабочих столов

– Windows 10 может использоваться как на ПК, так и на переносных устройствах, при этом все официальные приложения устанавливаются на любое устройство с поддержкой Windows 10.

– низкие системные требования новой операционной системы позволяют устанавливать ее даже на устаревших устройствах.

– пользователь имеет возможность вернуться на предыдущие версии Windows, если сохранена резервная копия на жестком диске. [9]

Однако Windows 10 имеет ряд минусов:

– операционная система обновляется автоматически, не предупредив пользователя;

– проблемы с приватностью. Лицензионное соглашение Windows 10 дает Microsoft право на сбор вашей персональной информации и статистики – данных о местоположении и истории браузера.

Причины выбора Embarcadero Rad Studio 10.2. 24 марта 2017 года Embarcadero Technologies объявила о выпуске RAD Studio 10.2. Релиз включает в себя: первый Linux-компилятор для RAD Studio, улучшенную систему меню в IDE для быстрой навигации, множество обновлений и новые функции для компонентов FireMonkey, новые возможности TDataSet, поддержку multi-tenancy в RAD Server, обновленный FireDAC, ряд расширений в RTL, улучшенную поддержку SOAP, существенное увеличение производительности компилятора C++ . Linux-компилятор Delphi дает возможность пользователям использовать Linux–серверы для имеющихся серверных Windows-приложений. После добавления Linux разработчики RAD Studio получили возможность кросс-компиляции на платформы — Windows, Linux, macOS, iOS и Android.

Rad Studio 10.2 обладает большими преимуществами и новыми возможностям по сравнению с прошлыми версиями:

1. Поддержка Direct2D, Windows 7 API и ввода с помощью мультисенсорных систем.
2. Поддержка прикасаний и жестов для Windows.
3. Мгновенный доступ к любой функции, компоненту или параметру с помощью IDE Insight.
4. Большое количество усовершенствований, предназначенных для повышения уровня производительности; [20]
5. Встроенная поддержка Firebird dbExpress.
6. Традиционный интерфейс языков Delphi 7, C++ Builder 6.

Также Rad Studio 10.2 содержит широкий спектр усовершенствований для технологии FireMonkey:

1. Поддержка многопоточности для TBitmap, TCanvas, TContext3D.
2. Ускоренную работу интерфейса на Android с поддержкой многопоточности.
3. Большое разнообразие средств взаимодействия с пользователем в TMultiView.
4. Новые стили FMX для MacOC и Android.

Новый проект Embarcadero включает в себя новые возможности и усовершенствованную работу с базами данных:

1. Встроенная поддержка MariaDB v5.5.
2. Возможность доступа к GUID в TField.
3. Поддержка MySQL v5.7.
4. Сокращен размер используемой памяти на клиенте при потоковой работе с BLOB.
5. Встроенная поддержка графических форматов JPEG и PNG в VCL.

С помощью инструментов Embarcadero RAD Studio 10.2 ускоряется создание различных приложений для компьютеров, сенсорных дисплеев, рабочих станций, информационных терминалов и Интернета. С учетом большого числа поддерживаемых платформ, а также типов программного обеспечения, архитектур и пользователей программный пакет Embarcadero

RAD Studio 10.2 является оптимальным, так как обладает выдающейся функциональностью, гибкостью и управляемостью.

Причины выбора языка программирования Delphi. Система программирования Delphi фирмы Embarcadero предоставляет наиболее широкие возможности для программирования приложений ОС Windows.

Deplhi – это продукт Borland International для быстрого создания приложений. RAD – среды, называемые также визуальными средами разработки, позволяют видеть как будут выглядеть рабочие и диалоговые окна программы на стадии проектирования. [15]

Высокопроизводительный инструмент визуального построения приложений включает в себя настоящий компилятор кода и предоставляет средства визуального программирования, несколько похожие на те, что можно обнаружить в Microsoft Visual Basic, которая не является RAD-системой или в других инструментах визуального проектирования. В основе Delphi лежит язык Object Pascal, который является расширением объектно-ориентированного языка Pascal. В Delphi также входят локальный SQL-сервер, генераторы отчетов, библиотеки визуальных компонентов, и прочее, необходимое для того, чтобы чувствовать себя совершенно уверенным при профессиональной разработке информационных систем или просто программ для Windows-среды. [17]

Прежде всего Delphi предназначен для профессиональных разработчиков, желающих очень быстро разрабатывать приложения в архитектуре клиент-сервер. Delphi производит небольшие по размерам высокоэффективные исполняемые модули (.exe и .dll), поэтому в Delphi должны быть, прежде всего, заинтересованы те, кто разрабатывает продукты на продажу. С другой стороны, небольшие по размерам и быстро исполняемые модули означают, что требования к клиентским рабочим местам существенно снижаются – это имеет немаловажное значение и для конечных пользователей.

Преимущества Delphi по сравнению с аналогичными программными продуктами.

– быстрота разработки приложения (RAD);
– высокая производительность разработанного приложения;
– низкие требования разработанного приложения к ресурсам компьютера;
– наращиваемость за счет добавления новых компонентов и инструментов в среду Delphi; [18]
– возможность разработки новых компонентов собственными средствами Delphi;
– высокая проработка иерархии объектов. [16]

Система программирования Delphi рассчитана на программирование различных приложений и предоставляет большое количество компонентов для этого. К тому же Delphi предоставляет скорость и качество создания программ, а эти характеристики может обеспечить только среда визуального проектирования, способная взять на себя значительные объемы рутинной работы по подготовке приложений, а также согласовать деятельность группы постановщиков, кодировщиков, тестеров и технических писателей.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

2.1 Проектирование логической модели базы данных

Логическое проектирование БД - это процесс конструирования общей информационной модели предприятия на основе отдельных моделей данных пользователей, которая является независимой от особенностей реально используемой СУБД и других физических условий.

Логическое проектирование состоит из нескольких этапов. [5]

Этап 1:

1. Преобразование локальной концептуальной модели данных в локальную логическую модель. (Удаление связей M: H, сложных связей, рекурсивных связей, связей с атрибутами, удаление множественных атрибутов.)
2. Определение комплекса отношений исходя из структуры модели данных.
3. Проверка модели с на соответствие правилам нормализации.
4. Проверка модели в отношении транзакций пользователей.
5. Реализация диаграммы сущность-связь.
6. Соответствие условиям поддержки целостности, включая целостность обязательных данных, ограничения для доменов атрибутов, целостность сущностей.
7. Утверждение разработанных логических моделей данных с пользователями системы.

Этап 2:

1. Анализ сущностей и связей и последующее слияние локальных моделей в единую глобальную модель данных.
2. Нормализация логической модели данных.
3. Создание конечного варианта диаграммы ER. [6]

Логическая модель данных описывает факты и объекты, которые необходимо зарегистрировать в базе данных. Основными элементами этой модели являются сущности, атрибуты и связи между сущностями. Физическим аналогом сущности в базе является таблица, а физическим аналогом атрибута — поле таблицы. С логической точки зрения сущность – это совокупность однотипных объектов, которые именуются экземплярами этой сущности. Физическим аналогом экземпляра является запись в таблице базы данных. Как и записи в таблице реляционной СУБД, экземпляры сущности должны быть уникальными, то есть набор значений их атрибутов не должен повторятся. Так же, как поля в таблице, атрибуты делятся на ключевые и неключевые. Помимо этого, на этапе логического проектирования для каждого атрибута необходимо определить тип данных.

Логическая модель данных является прототипом будущей базы данных. Она строится в рамках информационных единиц, но без привязки к конкретной СУБД. Также, логическая модель данных необязательно должна соответствовать средствам реляционной модели данных. Основным методом разработки логической модели данных являются варианты диаграммы сущность-связь(ER-диаграммы). Одну и ту же ER-модель можно преобразовать как в реляционную модель данных, так и в модель данных для иерархических и сетевых СУБД, или в постреляционную модель данных.

Решения, принятые на этапе создания модели предметной области, задают границы, в рамках которых можно развивать логическую модель, в пределах же этих границ можно принимать различные решения.

В модели на основе записей база данных состоит из нескольких записей фиксированного формата, которые могут иметь разные типы. Каждый тип записи определяет фиксированное количество полей, каждое из которых имеет фиксированную длину. Существует три основных типа логических моделей данных на основе записей:

– иерархическая;

– сетевая;

– реляционная.

В сетевой модели данные существуют в виде коллекции записей, а связи - в виде наборов. В отличие от реляционной модели, связи здесь моделируются наборами, которые задаются при помощи указателей. Сетевую модель можно выразить графом с записями в виде узлов графа и наборами в виде его ребер.

Иерархическая модель является подтипом сетевой модели. В ней данные представлены в виде коллекции записей, а связи - наборами. Однако в иерархической модели узел может иметь только одного родителя. Иерархическая модель может быть представлена как древовидный граф с записями в виде узлов (которые также называются сегментами) и множествами в виде ребер.

Реляционная модель данных**.** Основная идея реляционной модели данных заключается в том, чтобы представить любой набор данных в виде двумерного массива – таблицы. В простейшем случае реляционная модель описывает единственную двумерную таблицу, но чаще эта модель описывает структуру и взаимоотношения между несколькими различными таблицами. Реляционные модели данных являются основным способом проектирования и организации информационных систем.

Базовые понятия ER-диаграмм:

– сущность - это множество экземпляров объектов, обладающих общими атрибутами и характеристиками. Любой объект системы может быть представлен только одной сущностью, которая должна быть уникально идентифицирована. Каждая сущность может обладать любым количеством связей с другими сущностями модели.

– связь - ассоциация между двумя сущностями, при которой каждый экземпляр одной сущности ассоциирован с любым количеством экземпляров второй сущности, и наоборот.

Связи по признаку множественности могут быть трех типов:

1) один-к-одному(1:1);

2) один-ко-многим(1:M);

3) многие-ко-многим(M:M).

– атрибут - характеристика сущности, существенная для рассматриваемой предметной области и предназначенная для классификации или выражения состояния сущности. Атрибут определяет тип характеристик объектов. Экземпляр атрибута — это определение отдельного элемента множества атрибутов. На ER- диаграмме, атрибуты должны ассоциироваться с конкретными сущностями.

Модель одной из баз данных, используемых в программе, представлена на рисунке 2.1. и рисунке 2.2.

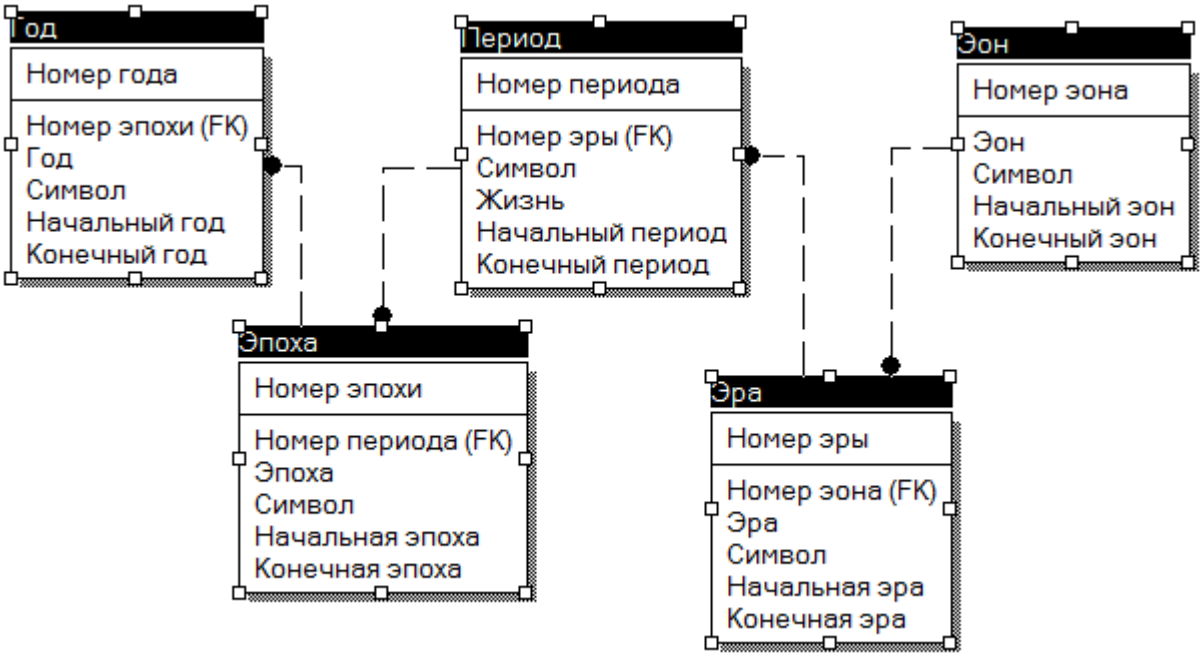

Рис. 2.1 Логическая модель базы данных Reference

На рисунке 2.2. представлены не связанные между собой таблицы этой базы данных, которые используются лишь для непосредственного хранения данных.

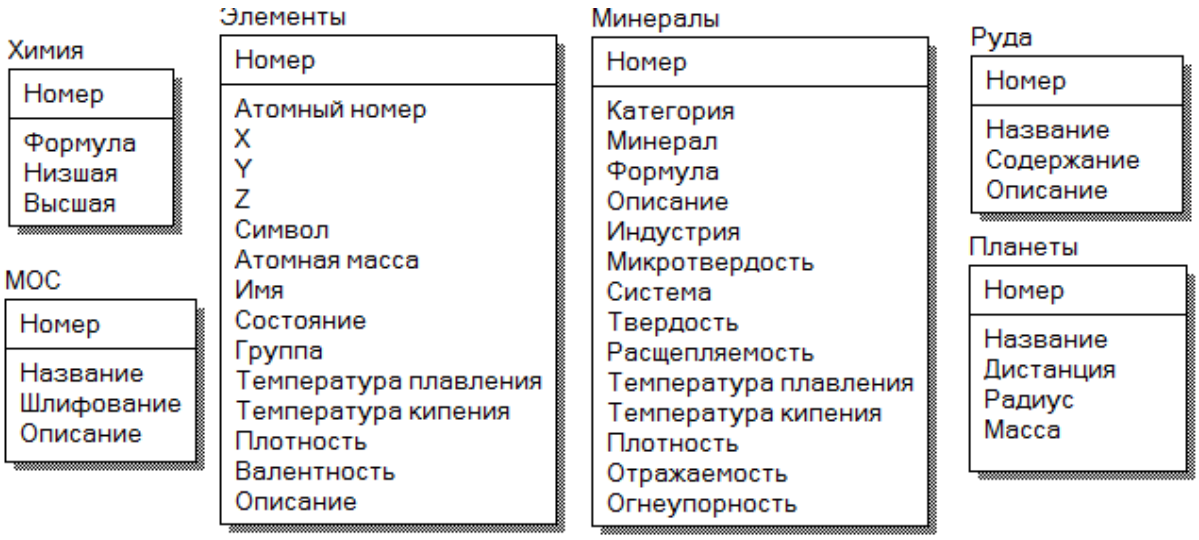

Рис.2.2. Логическая модель базы данных Reference

2.2 Проектирование физической модели базы данных

Цель физического проектирования базы данных – описание конкретной реализации базы данных, размещаемой во внешней памяти компьютера. Оно включает в себя описание структуры хранения данных и эффективных методов доступа к данным базы. При логическом проектировании отвечают на вопрос – что надо сделать, а при физическом – выбирается способ, как это сделать. [6]

Процедуры физического проектирования следующие:

1. Проектирование таблиц базы данных средствами выбранной СУБД. Осуществляется выбор реляционной СУБД, которая будет использоваться для создания базы данных, размещаемой на машинных носителях. Глубоко изучаются ее функциональные возможности по проектированию таблиц. Затем выполняется проектирование таблиц и схемы их связи в среде СУБД. Подготовленный проект базы данных описывается в сопровождаемой документации. [7]
2. Реализация бизнес-правил в среде выбранной СУБД. Обновление информации в таблицах может быть ограничено бизнес-правилами. Способ их реализации зависит от выбранной СУБД. Одни системы для реализации требований предметной области предлагают больше

возможностей, другие - меньше. В некоторых системах вообще отсутствует поддержка реализации бизнес-правил. В таком случае разрабатываются приложения для реализации их ограничений. Все решения, принятые в связи с реализацией бизнес-правил предметной области, подробно описываются в сопроводительной документации.

3. Проектирование физической организации базы данных. На этом шаге выбирается оптимальная файловая организация для таблиц. Задаются транзакции, которые будут выполняться в проектируемой базе данных, и выделяются наиболее важные из них. Анализируется пропускная способность транзакций - количество транзакций, которые могут быть обработаны за определенный интервал времени, и время ответа - промежуток времени, необходимый для выполнения одной транзакции. На основании указанных показателей принимаются решения об оптимизации производительности базы данных путем определения индексов в таблицах, ускоряющих выборку данных из базы, или снижения требований к уровню нормализации таблиц. Проводится оценка дискового объема памяти, необходимого для размещения создаваемой базы данных.

4. Разработка стратегии защиты базы данных. База данных представляет собой ценный корпоративный ресурс, и организации ее защиты уделяется большое внимание. Для этого разработчики базы данных должны иметь полное и ясное представление обо всех средствах защиты, предоставляемых в выбранной СУБД.

5. Организация мониторинга функционирования базы данных и ее настройка. После создания физического проекта базы данных организуется непрерывное слежение за ее функционированием. Полученные сведения об уровне производительности базы данных используются для ее настройки. Для этого привлекаются средства выбранной СУБД.

Физическая модель одной из базы данных, используемых в программе, изображена на рисунках 2.3 и 2.4.

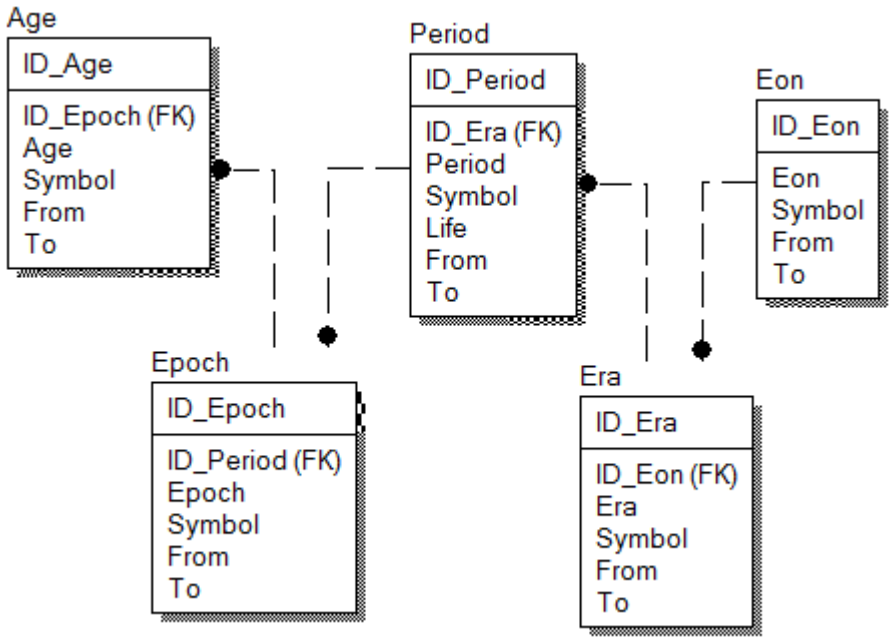

Рис. 2.3 Физическая модель базы данных Reference

На рисунке 2.4. представлены не связанные между собой таблицы этой базы данных, которые используются лишь для непосредственного хранения данных.

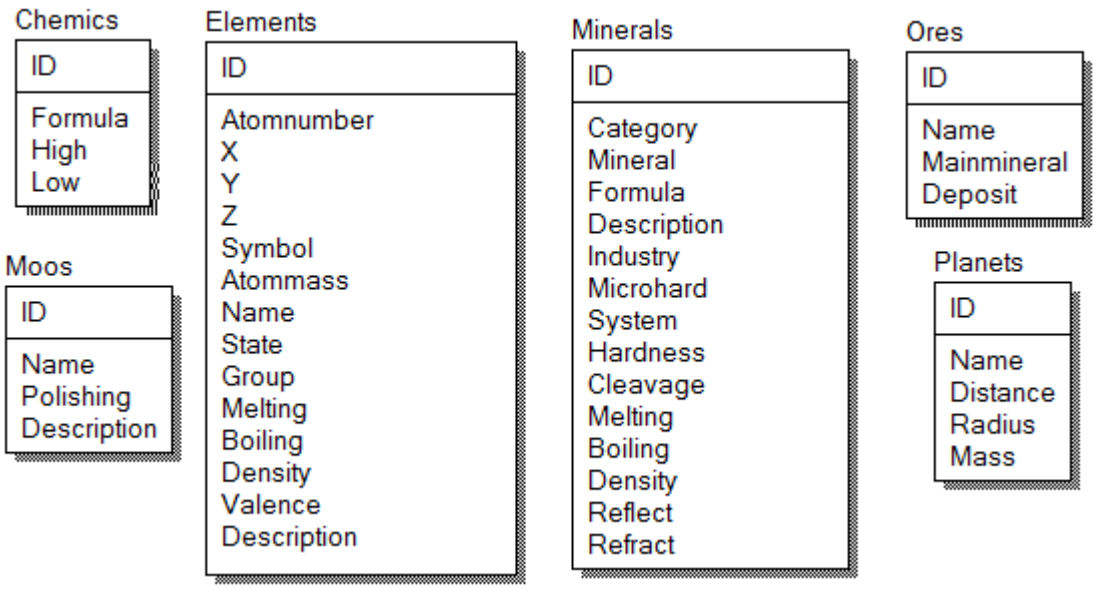

Рисунок 2.4 Физическая модель базы данных Reference

ГЛАВА 3. РАЗРАБОТКА И ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.1 Создание интерфейса программы

Для создания приложения был использован программный продукт Embarcadero Rad Studio 10.2 Tokyo. С его помощью было создано основное окно приложения, содержащее главное меню, ссылки на модули приложения и браузер базы данных.

Основное окно приложения изображено на рисунке 3.1.

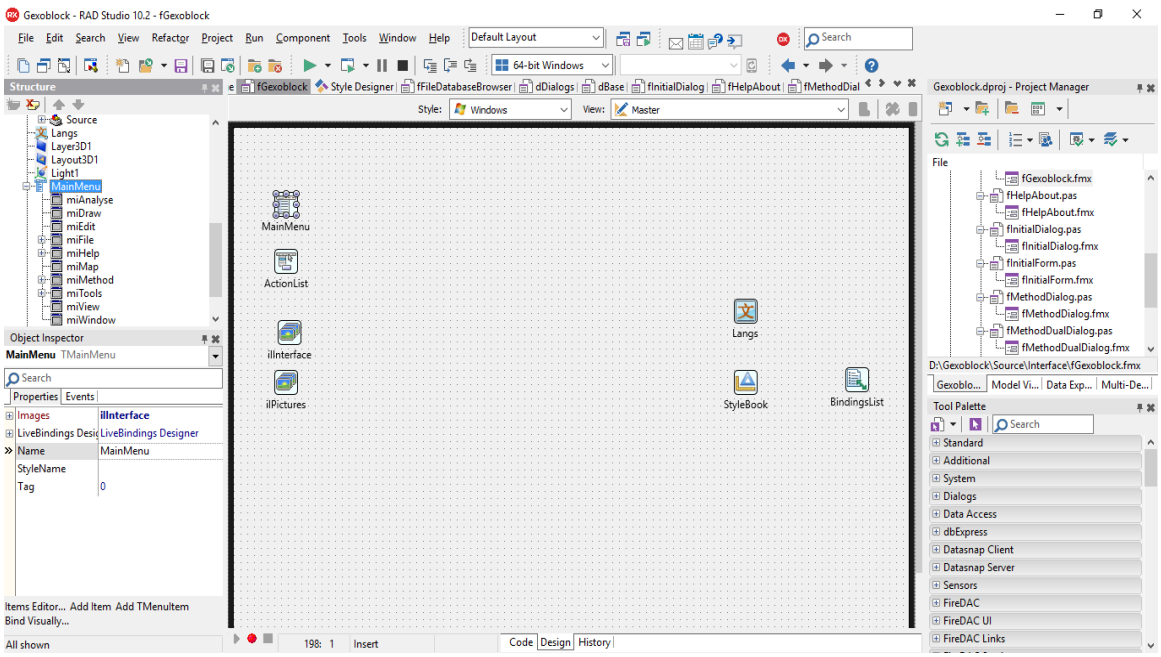

Рис 3.1. Основное окно

Переход между модулями приложения осуществляется с помощью компонента MainMenu, который служит для хранения ссылок на модули приложения. Рисунок 3.2. содержит изображение элементов компонента MainMenu.

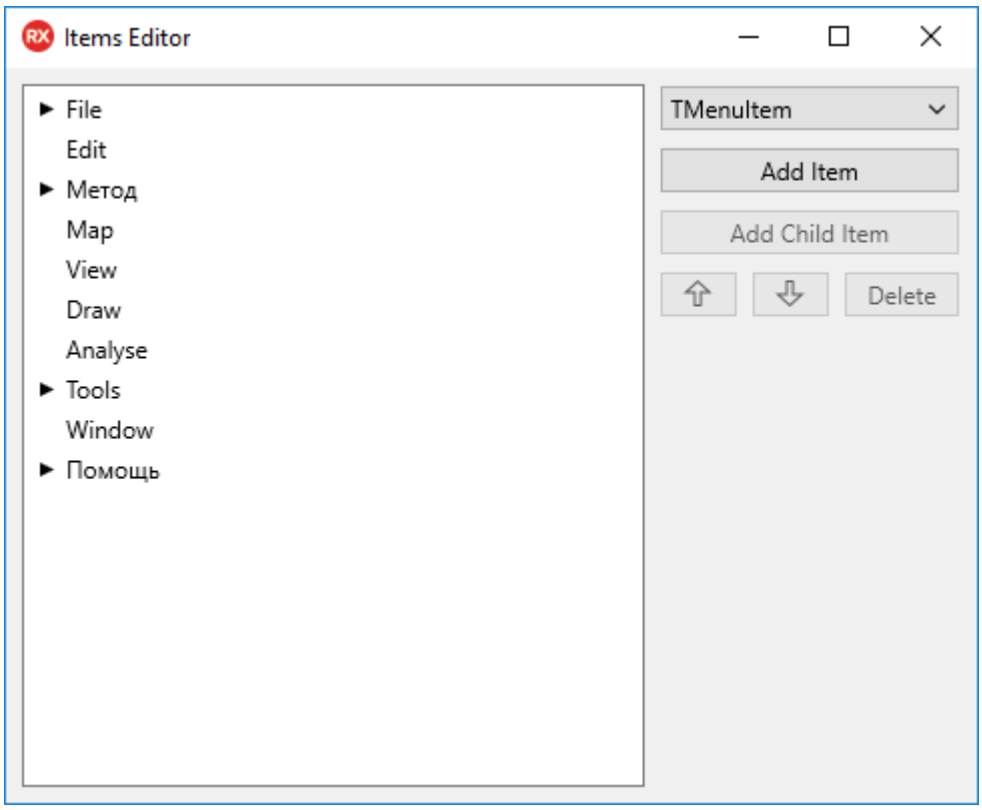

Рис.3.2. Компонент MainMenu

Непосредственный переход между модулями приложения происходит с помощью компонента ActionList, который изображен на рисунке 3.3.

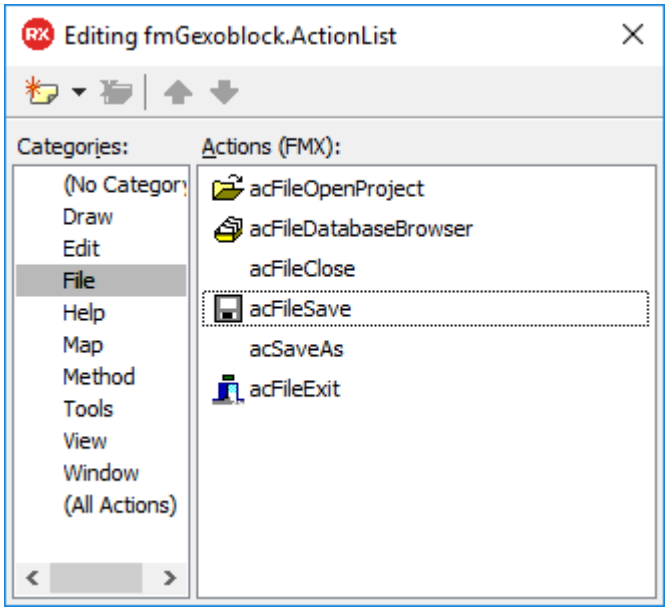

Рис.3.3. Компонент ActionList

Просмотр баз данных выполняется в браузере баз данных, который содержится в модуле fFileDatabaseBrowser. Модуль fFileDatabaseBrowser изображен на рисунке 3.4.

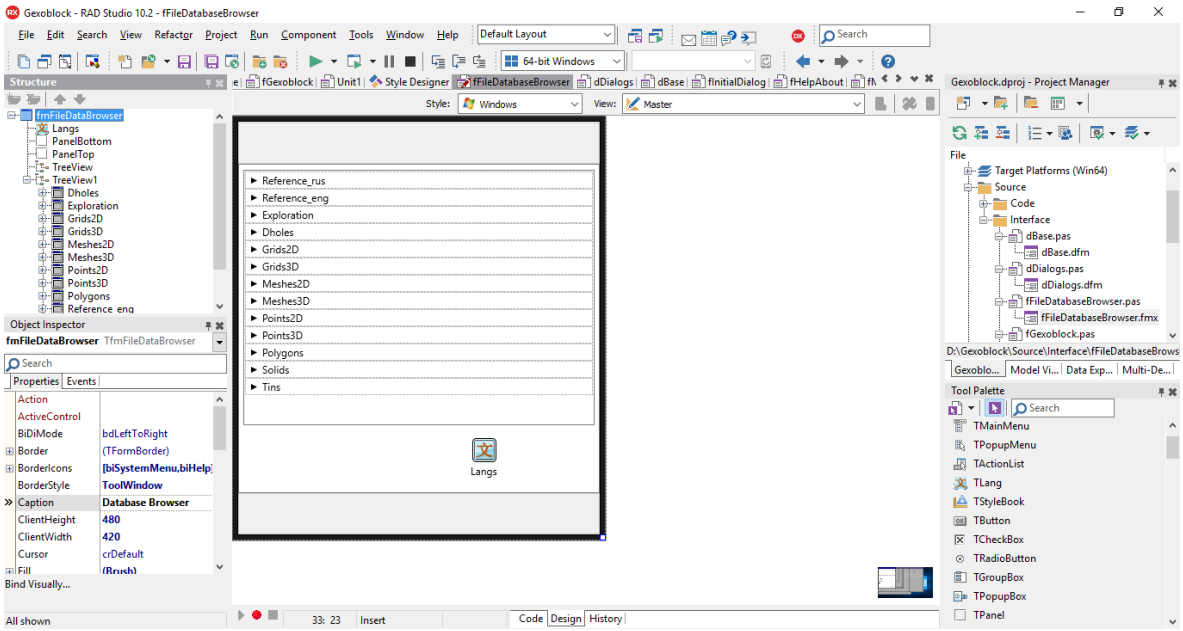

Рисунок 3.4. Модуль fFileDatabaseBrowser.

Просмотр различных баз данных происходит при помощи компонента TreeView, изображенном на рисунке 3.5.

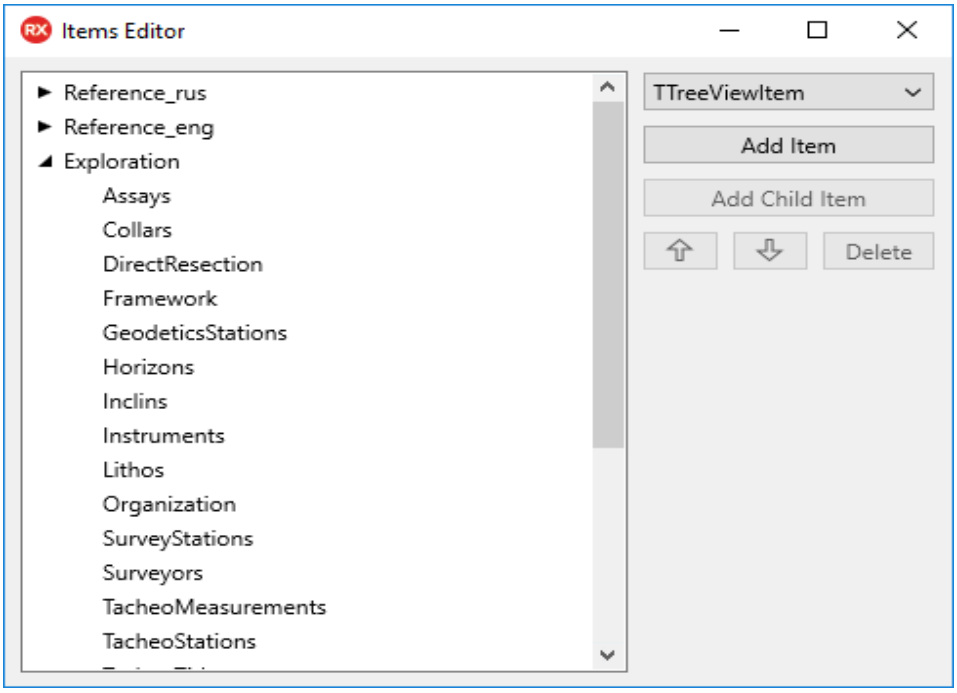

Рисунок 3.5. Компонент TreeView

Вывод требуемой информации из таблиц баз данных происходит в модуле Browser. Модуль Browser изображен на рисунке 3.6.

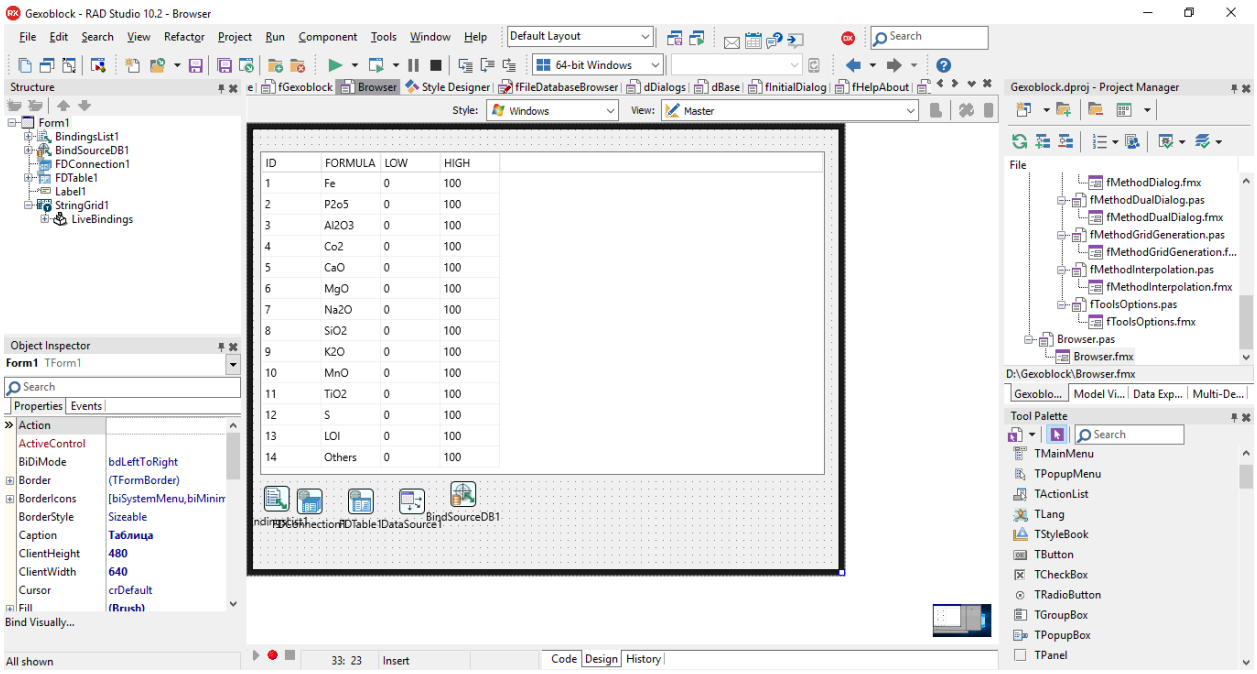

Рисунок 3.6. Модуль Browser

3.2 Создание программного приложения формирования и ведения базы данных

Следующим шагом в разработке приложения идет программная реализация модулей приложения.

В первую очередь необходимо провести соединение с файлом баз данных. Для этого в модуле приложения dBase создаем компонент FDManager и подключаем файл баз данных, как изображено на рисунке 3.7.

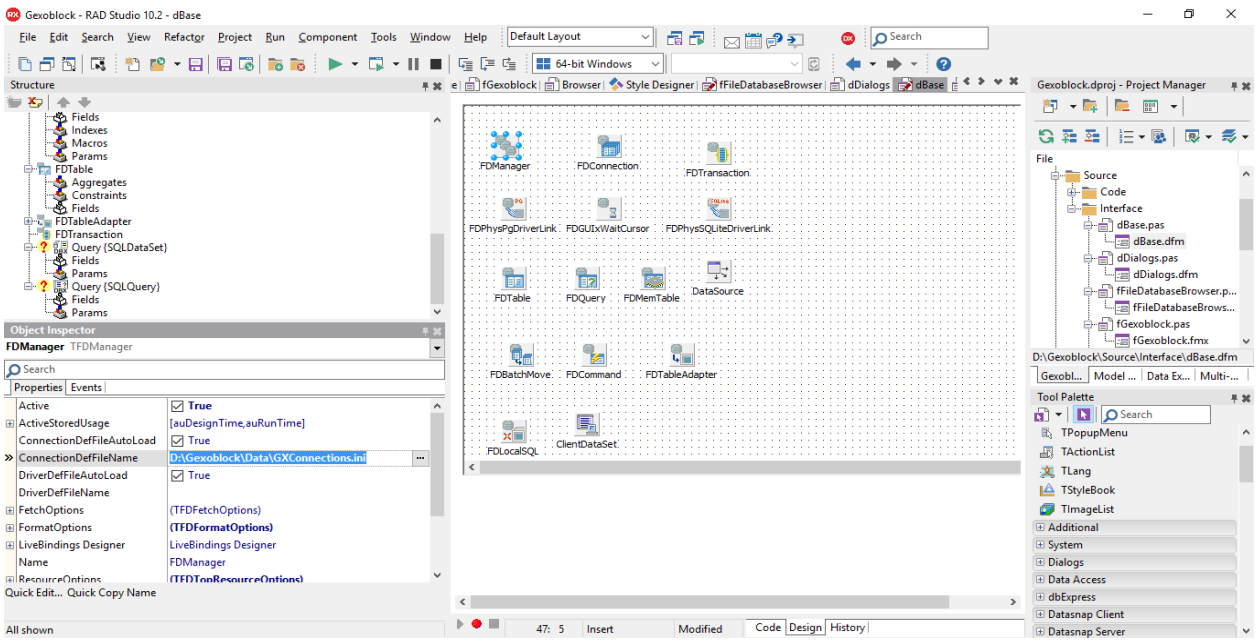

Рисунок 3.7. Подключение файла базы данных

Затем создаем компонент FDConnection и проверяем успешно ли установлено соединение. Проверка соединения изображена на рисунке 3.8.

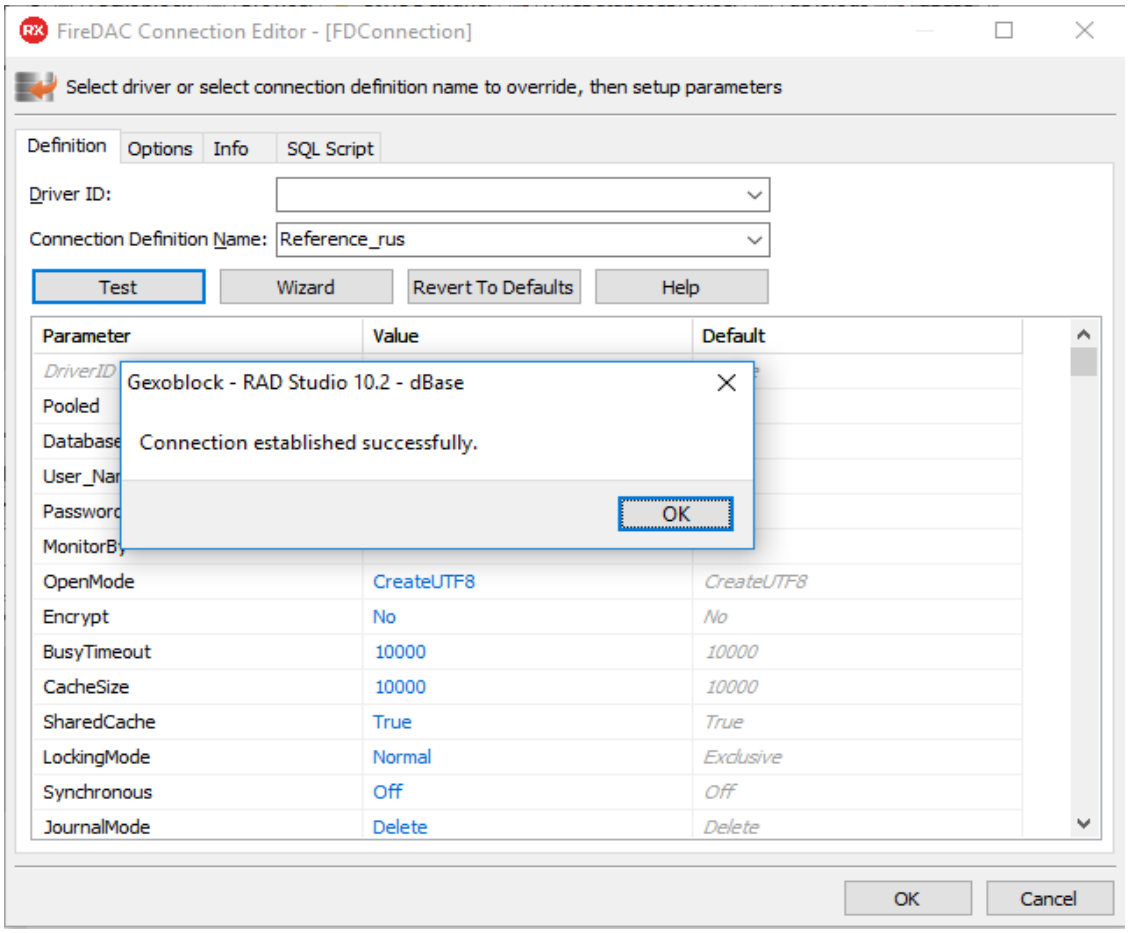

Рисунок 3.8. Проверка соединения с базой данных

Программный код открытия базы данных представлен в листинге 3.1.

Листинг 3.1. Открытие базы данных

```
procedure OpenBase(database, table: string);
begin
Form1.FDConnection1.Connected := false;
Form1.FDTable1.Active := false;
Form1.FDConnection1.Params.Database := database;
Form1.FDTable1.TableName := table;
Form1.FDConnection1.Connected := true;
Form1.FDTable1.Active := true;
Form1.Show;
end;
```

После успешной настройки соединения необходимо осуществить вывод таблиц базы данных. Данное приложение использует GUI-фреймворк FireMonkey, что несколько осложняет вывод данных. В отличии от VCL, FireMonkey не имеет управляемых данными элементов. Однако, компоненты LiveBindings позволяют соединять поля базы данных со стандартными элементами.

Для настройки вывода данных из БД, необходимо сначала создать компоненты TClientDataSet и TDataSource. Затем проводим соединение TDataSource с TClientDataSet в свойствах компонента, как изображено на рисунке 3.9.

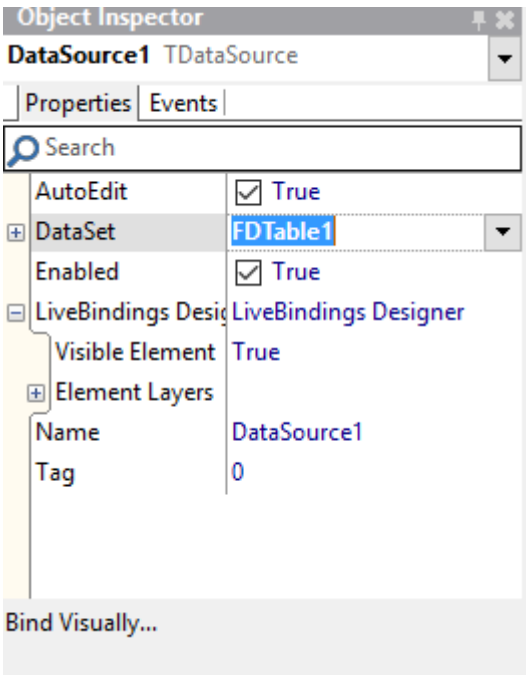

Рисунок 3.9. Соединение TDataSource с TClientDataSet

После этого помещаем компонент TStingGrid и начинаем процесс привязывания. Помещаем TBindingList и TBindSource, проводим соединение с компонентом TDataSource. Соединение изображено на рисунке 3.10.

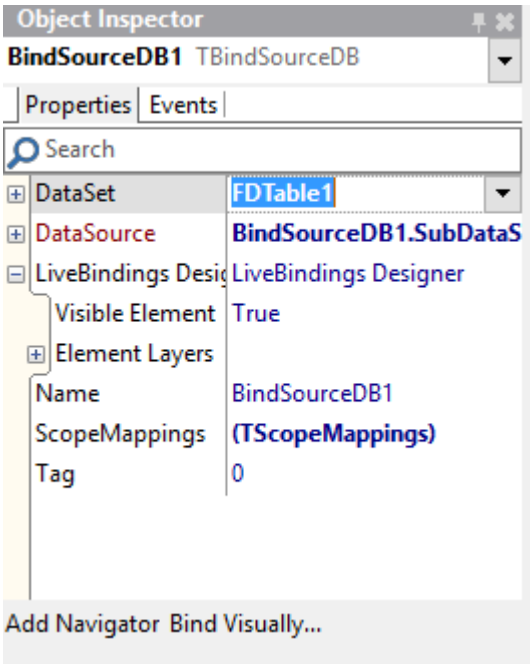

Рисунок 3.10. Соединение TBindSource с компонентом TDataSource

Наконец, можно привязать базу данных к StingGrid, используя параметр LiveBindings (рис.3.11).

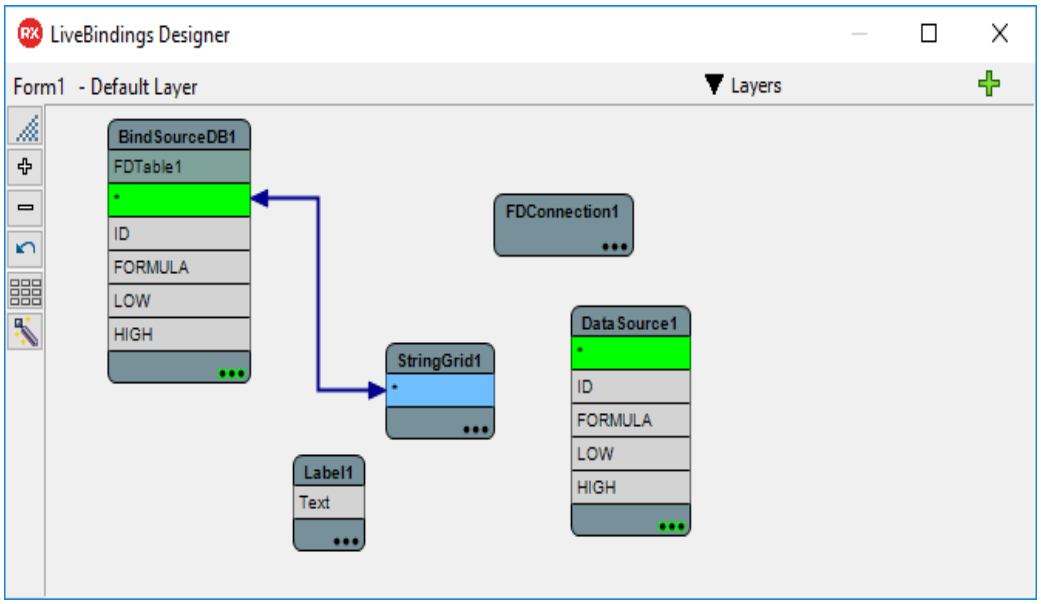

Рисунок 3.11. Привязка базы данных

Для автоматизации открытия каждой таблицы базы данных создадим процедуру для всех таблиц. Процедура открытия таблицы представлена в листинге 3.2.

Листинг 3.2. Процедура открытия таблицы

```
procedure TfmFileDataBrowser.TreeViewItem27Click(Sender: TObject);
begin
inherited;
OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','age');
end;
```

3.3 Тестирование приложения

После завершения разработки, необходимо провести тестовый запуск программы, используя .exe файл. Главное окно программы содержит ссылки на все модули приложения - изображено на рисунке 3.12.

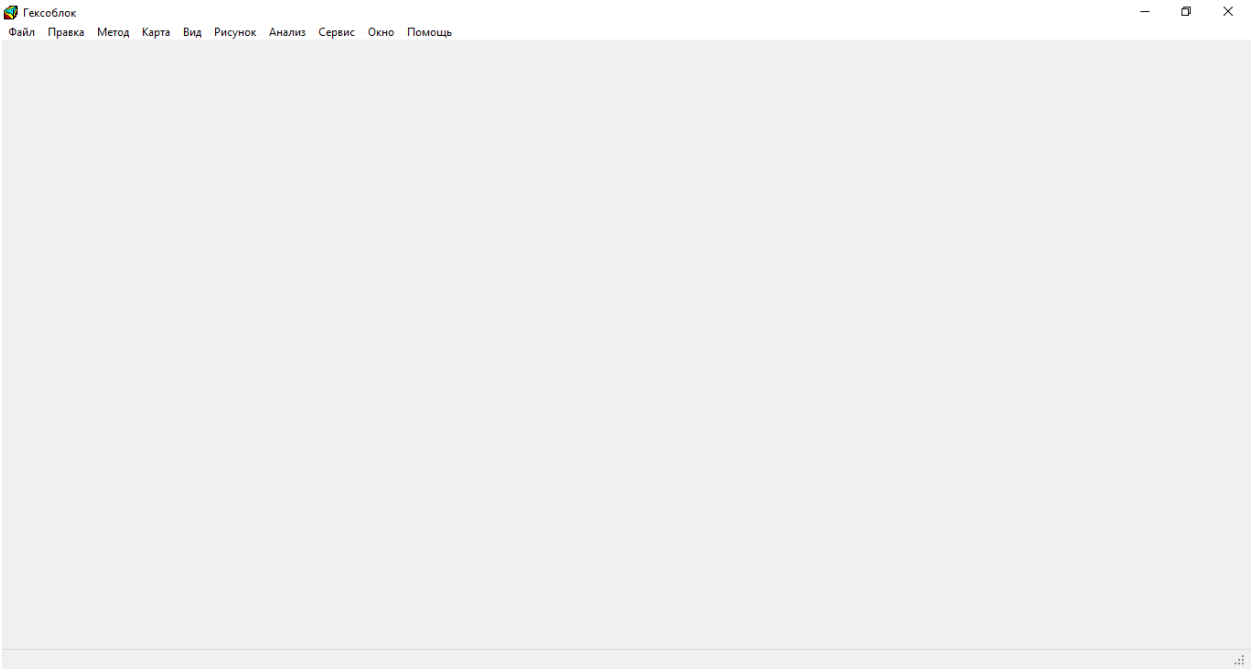

Рисунок 3.12. Главное окно программы

Основным модулем программы является модуль «Браузер базы данных». В нем содержатся данные из БД, а также ссылки на таблицы базы данных.

На рисунке 3.13. изображен модуль «Браузер базы данных».

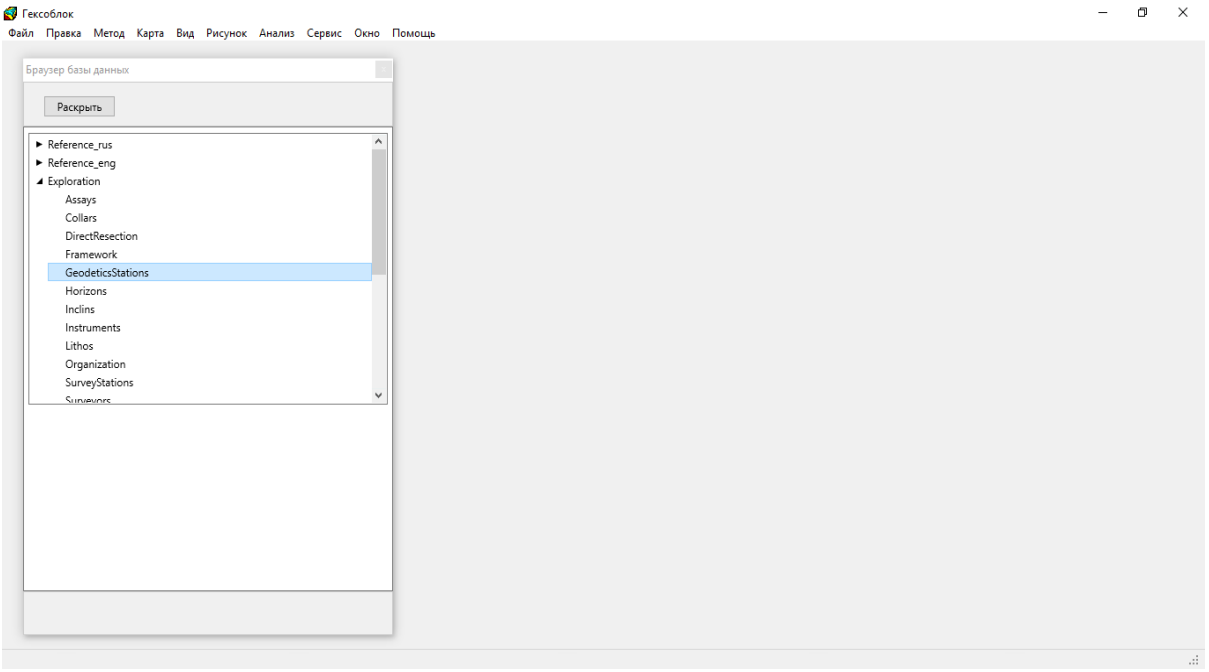

Рисунок 3.13. «Модуль Браузер базы данных»

С помощью этого модуля можно просмотреть данные из любой таблицы. Для этого необходимо щелкнуть по интересующей таблицы и в новом окне откроется таблица, содержащая нужные данные. На рисунке 3.14. показана таблица GeodeticStations из базы данных Exploration.

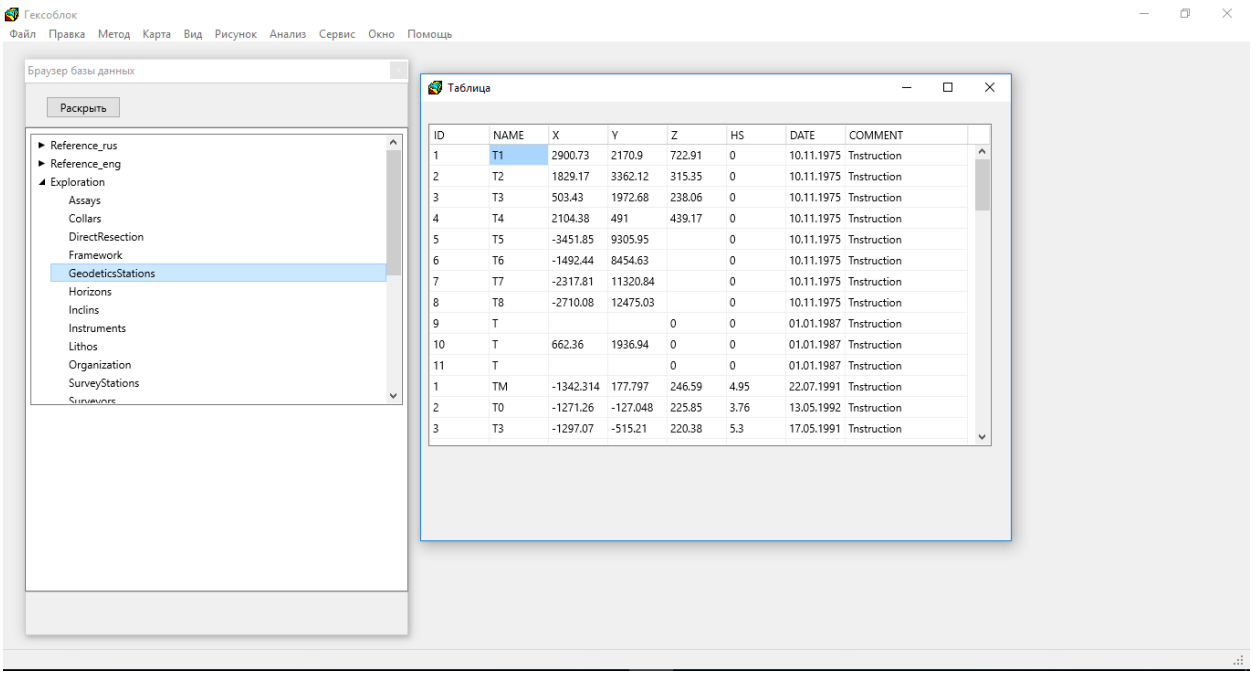

Рисунок 3.14. Таблица GeodeticStations

Заключение

В ходе написания данной выпускной квалификационной работы было спроектировано и разработано программное обеспечения формирования и ведения базы данных.

При решении этой задачи была изучена предметная область недропользования, обоснована необходимость разработки программного продукта, изучены особенности СУБД PostgreSQL.

В проектировании и разработки программного приложения были использованы следующие средства: язык программирования Delphi, среда разработки Embarcadero Rad Studio 10.2 Tokyo и СУБД PostgreSQL.

Разработанное приложение позволяет эффективно выполнять задачу формирования и ведения базы данных.

Список используемых источников

1. ГОСТ 2.105. Общие требования к текстовым документам, 2014 г. – 2 с.

2. ГОСТ 7.32. Отчет о научно-исследовательской работе. Структура и правила оформления, 2014 г. -22 с.

3. Агальцов, В. П. Базы данных. Распределенные и удаленные базы данных: Учебник/В. П. Агальцов. – Москва: ИД ФОРУМ: НИЦ Инфра, 2013. – 272 с.

4. Леонтьев, В. С. "Windows 10 Новейший самоучитель"/В. С. Леонтьев. - Эксмо, 2015 год, 528 с.

5. Логическое проектирование [Электронный ресурс]. – Режим доступа: http://life-prog.ru/1_5589_lektsiya-.html(дата обращения 05.06.2017г).

6. Логическое проектирование баз данных [Электронный ресурс]. – Режим доступа: http://www.studfiles.ru/preview/2152676/page:7/ (дата обращения 05.06.2017г).

7. Физическое проектирование базы данных [Электронный ресурс]. – Режим доступа: http://www.bourabai.kz/dbt/dbms/03.htm (дата обращения 01.06.2017г).

8. Физическое моделирование. Особенности построения физической модели базы данных [Электронный ресурс]. – Режим доступа: http://studme.org/77262/informatika/izicheskoe_modelirovanie (дата обращения 05.06.2017г).

9. Windows 10 [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Windows_10 (дата обращения 05.06.2017г).

10. Ахаян, Р. Эффективная работа с СУБД / Р. Ахаян, А. Горев, С Макашарипов. – СПб.: Питер, 1997. – 105 с.

11. Кузнецов, С.Д. Базы данных. Модели и языки / С.Д. Кузнецов. – Бином, 2008. – 720 с.

12. Хомоненко, А.Д. Базы данных / А.Д. Хомоненко, М.Г. Мальцев. – Корона Принт, 2007. – 736 с.

13. Перминов, Г.И. Система управления базами данных / Г.И. Перминов. – М.: ВШЭ, 1998. – 288с.

14. Диго, С.М. Проектирование баз данных / С.М. Диго. – М.: ЦентрЮрИнфоР, 2001. – 276 с.

15. Фараонов, В. В. Система программирования Delphi / Фараонов, В. В. – СПб.: БХВ - Петербург, 2012. – 912 с.: ил.

16. Фаронов, В. В. Delphi. Программирование на языке высокого уровня: Учебник для вузов / Фаронов, В. В. – СПб.: Петербург, 2010. – 640 с.

17. Емельянов В.И. Основы программирования на Delphi / Емельянов В.И. - М.: Высшая школа, 2005. – 226с.

18. Зубов А. Программирование на Delphi / Зубов А. - СПб.: Питер, 2005. – 397с.

19. Официальная документация к PostgreSQL 9.6.3 [Электронный ресурс]. - Режим доступа: https://postgrespro.ru/docs/postgresql/9.6/ (дата обращения 05.06.2017)

20. Официальная документация к Embarcadero Rad Studio 10.2. Tokyo [Электронный ресурс]. – Режим доступа: http://docs.embarcadero.com/products/rad_studio/ (дата обращения 02.06.2017)

21.Волков А.М. Недропользование / Волков А.М. - Вестник РУДН. Серия: Юридические науки. 2011. – 325с.

ПРИЛОЖЕНИЕ

Модуль FGexoblock

```
unit fGexoblock;
interface
uses
  System.SysUtils,
  System.Types,
  System.UITypes,
  System.Classes,
  System.Variants,
  System.SysConst,
  System.Bindings.Outputs,
  System.Rtti,
  System.IniFiles,
  System.Math.Vectors,
  System.Actions,
  System.ImageList,
  Data.Bind.EngExt,
  Data.Bind.Components,
  FMX.Types,
  FMX.StdCtrls,
  FMX.Controls,
  FMX.Controls3D,
  FMX.Forms,
  FMX.Dialogs,
  FMX.Ani,
  FMX.Menus,
  FMX.Bind.DBEngExt,
  FMX.Bind.Editors,
  FMX.Types3D,
  FMX.Layers3D,
  FMX.Controls.Presentation,
  FMX.StdActns,
  FMX.ActnList,
  FMX.ImgList,
  dBase,
  dDialogs,
  fInitialForm,
  fFileDatabaseBrowser,
  fMethodGridGeneration,
  fMethodInterpolation,
  uGlobals;
type
  TfmGexoblock = class(TfmInitialForm)
    StatusBar1: TStatusBar;
    Layer3D1: TLayer3D;
    Layout3D1: TLayout3D;
```

```
Light1: TLight;
BindingsList: TBindingsList;
BindExpressionmiFileNew1: TBindExpression;
StyleBook: TStyleBook;
ActionList: TActionList;
acFileOpenProject: TAction;
acFileClose: TAction;
acView: TViewAction;
acWindowClose: TWindowClose;
acFileExit: TFileExit;
acFileDatabaseBrowser: TAction;
acSaveAs: TAction;
acMethodInterpolation: TAction;
acMethodGridGeneration: TAction;
acMethodTriangulation: TAction;
acMethodSetOperations: TAction;
MainMenu: TMainMenu;
miFile: TMenuItem;
miMethod: TMenuItem;
miHelp: TMenuItem;
miFileOpenProject: TMenuItem;
miFileDatabaseBrowser: TMenuItem;
MenuItemFileSep: TMenuItem;
MenuItemExit: TMenuItem;
MenuItem3: TMenuItem;
MenuItemHelpSep: TMenuItem;
miHelpAbout: TMenuItem;
MenuMethodInterpolation: TMenuItem;
miMap: TMenuItem;
miView: TMenuItem;
miDraw: TMenuItem;
miAnalyse: TMenuItem;
miTools: TMenuItem;
miWindow: TMenuItem;
miEdit: TMenuItem;
MenuItemToolsOptions: TMenuItem;
Action1: TAction;
acEditUndo: TAction;
Action2: TAction;
Action3: TAction;
acToolsOptions: TAction;
acHelpAbout: TAction;
acContent: TAction;
Action6: TAction;
acHelpSep: TAction;
MenuMethodGridGeneration: TMenuItem;
miMethodTriangulation: TMenuItem;
MenuItem16: TMenuItem;
ilInterface: TImageList;
ilPictures: TImageList;
miHelpGlossary: TMenuItem;
Langs: TLang;
```

```
    acFileSave: TAction;
    miFileSave: TMenuItem;
    procedure FormCreate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure acFileExitCanActionExec(Sender: TCustomAction;
      var CanExec: Boolean);
    procedure acFileDatabaseBrowserExecute(Sender: TObject);
    procedure acSaveAsExecute(Sender: TObject);
    procedure acMethodInterpolationExecute(Sender: TObject);
    procedure acHelpAboutExecute(Sender: TObject);
    procedure acToolsOptionsExecute(Sender: TObject);
    procedure acMethodGridGenerationExecute(Sender: TObject);
    procedure FormShow(Sender: TObject);
  public
    BinPath : TFileName;
    IniFile: TIniFile;
    procedure ReadIniFile; override;
    procedure WriteIniFile; override;
  private
    procedure TranslateMainMenu;
    procedure DefaultLayout;
  end;
var
  fmGexoblock: TfmGexoblock;

uses
  fHelpAbout,
  fToolsOptions;

procedure TfmGexoblock.TranslateMainMenu;
var
  I : Integer;
begin
  //LoadLangFromStrings(Langs.LangStr[CurLang]); to switch in runtime
  Langs.Lang := CurLang;
  Caption := Translate(Caption);

  miFile.Text := Translate(miFile.Text);
  for I := 0 to miFile.ItemsCount - 1 do
    miFile.Items[I].Text := Translate(miFile.Items[I].Text);

  miEdit.Text := Translate(miEdit.Text);
  for I := 0 to miEdit.ItemsCount - 1 do
    miEdit.Items[I].Text := Translate(miEdit.Items[I].Text);

  miMethod.Text := Translate(miMethod.Text);
  for I := 0 to miMethod.ItemsCount - 1 do
    miMethod.Items[I].Text := Translate(miMethod.Items[I].Text);

  miDraw.Text := Translate(miDraw.Text);
  for I := 0 to miDraw.ItemsCount - 1 do
    miDraw.Items[I].Text := Translate(miDraw.Items[I].Text);
```

```
  miMap.Text := Translate(miMap.Text);
  for I := 0 to miMap.ItemsCount - 1 do
    miMap.Items[I].Text := Translate(miMap.Items[I].Text);

  miView.Text := Translate(miView.Text);
  for I := 0 to miView.ItemsCount - 1 do
    miView.Items[I].Text := Translate(miView.Items[I].Text);

  miAnalyse.Text := Translate(miAnalyse.Text);
  for I := 0 to miAnalyse.ItemsCount - 1 do
    miAnalyse.Items[I].Text := Translate(miAnalyse.Items[I].Text);

  miTools.Text := Translate(miTools.Text);
  for I := 0 to miTools.ItemsCount - 1 do
    miTools.Items[I].Text := Translate(miTools.Items[I].Text);

  miWindow.Text := Translate(miWindow.Text);
  for I := 0 to miWindow.ItemsCount - 1 do
    miTools.Items[I].Text := Translate(miWindow.Items[I].Text);

  miHelp.Text := Translate(miHelp.Text);
  for I := 0 to miHelp.ItemsCount - 1 do
    miHelp.Items[I].Text := Translate(miHelp.Items[I].Text);
end;
procedure TfmGexoblock.DefaultLayout;
begin
  fmFileDataBrowser.Left := fmGexoblock.Left + 10;
  fmFileDataBrowser.Top := fmGexoblock.Top + 80;
  fmFileDataBrowser.Height := fmGexoblock.Height - 110;
end;
procedure TfmGexoblock.FormCreate(Sender: TObject);
begin
  inherited;
  ReadIniFile;
  TranslateMainMenu;
procedure TfmGexoblock.FormShow(Sender: TObject);
begin
  inherited;
  DefaultLayout;
  fmFileDataBrowser.Show;
end;
procedure TfmGexoblock.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  try
    //fmViewProjectManager.SaveToFile(ExpandPath(DirProject) + 'Gexoblock.prj');
  except
  end;
  WriteIniFile;
  inherited;
end;
```

```
procedure TfmGexoblock.acFileDatabaseBrowserExecute(Sender: TObject);
begin
  fmFileDataBrowser.Show;
end;
procedure TfmGexoblock.acFileExitCanActionExec(Sender: TCustomAction;
  var CanExec: Boolean);
begin
  Close;
end;
procedure TfmGexoblock.acMethodGridGenerationExecute(Sender: TObject);
begin
  with TfmMethodGridGeneration.Create(Self) do
  try
    ShowModal;
  finally
    Free;
  end;
end;
procedure TfmGexoblock.acMethodInterpolationExecute(Sender: TObject);
begin
  with TfmMethodInterpolation.Create(Self) do
  try
    ShowModal;
  finally
    Free;
  end;
end;
procedure TfmGexoblock.acSaveAsExecute(Sender: TObject);
begin
  //Save As
end;
procedure TfmGexoblock.acToolsOptionsExecute(Sender: TObject);
begin
  with TfmToolsOptions.Create(Self) do
  try
    ShowModal;
  finally
    Free;
  end;
end;
procedure TfmGexoblock.acHelpAboutExecute(Sender: TObject);
begin
  with TfmHelpAbout.Create(Self) do
    try
      ShowModal;
    finally
      Free;
    end;
end;
procedure TfmGexoblock.ReadIniFile;
var
  StyleID: integer;
```

```
begin
  inherited;
  BinPath := ExtractFilePath(ParamStr(0));
  SetCurrentDir(BinPath);
  IniFile := TIniFile.Create(BinPath + 'Gexoblock.ini');
  with IniFile do
    try
      Top := ReadInteger(Name, 'Top', 100);
      Left := ReadInteger(Name, 'Left', 200);
      if ReadBool(Name, 'InitMax', False) then
        // WindowState = wsMaximized
      else
        // WindowState = wsNormal;
        StyleID := ReadInteger(Name, 'StyleID', 0);
      if StyleID = 0 then
      begin
        // ActionManager.Style := StandardStyle;
        // ToolsStandardStyle.Checked := True;
      end
      else
      begin
        // ActionManager.Style  := XPStyle;
        // ToolsXPStyle.Checked := True;
      end
    finally
      IniFile.Free;
    end;
end;
procedure TfmGexoblock.WriteIniFile;
begin
  BinPath := ExtractFilePath(ParamStr(0));
  SetCurrentDir(BinPath);
  IniFile := TIniFile.Create(BinPath + 'Gexoblock.ini');
  with IniFile do
    try
      WriteInteger(Name, 'Top', Top);
      WriteInteger(Name, 'Left', Left);
      /// in VCL      WriteBool(Name, 'InitMax', WindowState = wsMaximized);
      {
        if ActionManager.Style = StandardStyle then
        WriteInteger(Name, 'StyleID', 0)
        else
        WriteInteger(Name, 'StyleID', 1);
      }
    finally
      IniFile.Free;
    end;
  inherited;
end;

end.
```

Модуль fFileDatabaseBrowser

```
unit fFileDatabaseBrowser;
interface
uses
  System.SysUtils,
  System.Types,
  System.UITypes,
  System.Classes,
  System.Variants,
  FMX.Types,
  FMX.Graphics,
  FMX.Controls,
  FMX.Forms,
  FMX.Dialogs,
  FMX.StdCtrls,
  fInitialForm,
  FMX.Controls.Presentation,
  FMX.Layouts,
  FMX.TreeView,
  FMX.Menus,

  uGlobals, Browser, FMX.ListBox;

type
  TfmFileDataBrowser = class(TfmInitialForm)
    TreeView: TTreeView;
    PanelBottom: TPanel;
    PanelTop: TPanel;
    btnExpand: TButton;
    Langs: TLang;
    TreeView1: TTreeView;
    TreeViewItem1: TTreeViewItem;
    TreeViewItem2: TTreeViewItem;
    TreeViewItem3: TTreeViewItem;
    TreeViewItem4: TTreeViewItem;
    TreeViewItem5: TTreeViewItem;
    TreeViewItem6: TTreeViewItem;
    TreeViewItem7: TTreeViewItem;
    TreeViewItem8: TTreeViewItem;
    TreeViewItem9: TTreeViewItem;
    TreeViewItem10: TTreeViewItem;
    TreeViewItem11: TTreeViewItem;
    TreeViewItem12: TTreeViewItem;
    TreeViewItem13: TTreeViewItem;
    TreeViewItem14: TTreeViewItem;
    TreeViewItem15: TTreeViewItem;
    TreeViewItem16: TTreeViewItem;
    TreeViewItem17: TTreeViewItem;
    TreeViewItem18: TTreeViewItem;
    TreeViewItem19: TTreeViewItem;
```

```
TreeViewItem20: TTreeViewItem;
TreeViewItem21: TTreeViewItem;
TreeViewItem22: TTreeViewItem;
TreeViewItem23: TTreeViewItem;
TreeViewItem24: TTreeViewItem;
TreeViewItem25: TTreeViewItem;
TreeViewItem26: TTreeViewItem;
TreeViewItem27: TTreeViewItem;
TreeViewItem28: TTreeViewItem;
TreeViewItem29: TTreeViewItem;
TreeViewItem30: TTreeViewItem;
TreeViewItem31: TTreeViewItem;
TreeViewItem32: TTreeViewItem;
TreeViewItem33: TTreeViewItem;
TreeViewItem34: TTreeViewItem;
TreeViewItem35: TTreeViewItem;
TreeViewItem36: TTreeViewItem;
TreeViewItem37: TTreeViewItem;
TreeViewItem38: TTreeViewItem;
TreeViewItem39: TTreeViewItem;
TreeViewItem40: TTreeViewItem;
TreeViewItem41: TTreeViewItem;
TreeViewItem42: TTreeViewItem;
TreeViewItem43: TTreeViewItem;
TreeViewItem44: TTreeViewItem;
TreeViewItem45: TTreeViewItem;
TreeViewItem46: TTreeViewItem;
TreeViewItem47: TTreeViewItem;
TreeViewItem48: TTreeViewItem;
TreeViewItem49: TTreeViewItem;
TreeViewItem50: TTreeViewItem;
TreeViewItem51: TTreeViewItem;
TreeViewItem52: TTreeViewItem;
TreeViewItem53: TTreeViewItem;
TreeViewItem54: TTreeViewItem;
TreeViewItem55: TTreeViewItem;
TreeViewItem56: TTreeViewItem;
TreeViewItem57: TTreeViewItem;
TreeViewItem58: TTreeViewItem;
TreeViewItem59: TTreeViewItem;
TreeViewItem60: TTreeViewItem;
TreeViewItem61: TTreeViewItem;
TreeViewItem62: TTreeViewItem;
TreeViewItem63: TTreeViewItem;
TreeViewItem64: TTreeViewItem;
TreeViewItem65: TTreeViewItem;
TreeViewItem66: TTreeViewItem;
TreeViewItem67: TTreeViewItem;
TreeViewItem68: TTreeViewItem;
TreeViewItem69: TTreeViewItem;
TreeViewItem70: TTreeViewItem;
TreeViewItem71: TTreeViewItem;
```

```
TreeViewItem72: TTreeViewItem;
TreeViewItem73: TTreeViewItem;
TreeViewItem74: TTreeViewItem;
Meshes2D: TTreeViewItem;
Pit_f: TTreeViewItem;
TreeViewItem77: TTreeViewItem;
TreeViewItem78: TTreeViewItem;
TreeViewItem79: TTreeViewItem;
TreeViewItem75: TTreeViewItem;
TreeViewItem76: TTreeViewItem;
TreeViewItem80: TTreeViewItem;
TreeViewItem81: TTreeViewItem;
TreeViewItem82: TTreeViewItem;
TreeViewItem83: TTreeViewItem;
TreeViewItem84: TTreeViewItem;
TreeViewItem85: TTreeViewItem;
TreeViewItem86: TTreeViewItem;
TreeViewItem87: TTreeViewItem;
TreeViewItem88: TTreeViewItem;
TreeViewItem89: TTreeViewItem;
TreeViewItem90: TTreeViewItem;
TreeViewItem91: TTreeViewItem;
TreeViewItem92: TTreeViewItem;
TreeViewItem93: TTreeViewItem;
TreeViewItem94: TTreeViewItem;
TreeViewItem95: TTreeViewItem;
TreeViewItem96: TTreeViewItem;
TreeViewItem97: TTreeViewItem;
TreeViewItem98: TTreeViewItem;
TreeViewItem99: TTreeViewItem;
TreeViewItem100: TTreeViewItem;
TreeViewItem101: TTreeViewItem;
TreeViewItem102: TTreeViewItem;
TreeViewItem103: TTreeViewItem;
TreeViewItem104: TTreeViewItem;
TreeViewItem105: TTreeViewItem;
TreeViewItem106: TTreeViewItem;
TreeViewItem107: TTreeViewItem;
procedure FormCreate(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure TreeViewItem2Click(Sender: TObject);
procedure TreeViewItem3Click(Sender: TObject);
procedure TreeViewItem4Click(Sender: TObject);
procedure TreeViewItem5Click(Sender: TObject);
procedure TreeViewItem6Click(Sender: TObject);
procedure TreeViewItem7Click(Sender: TObject);
procedure TreeViewItem8Click(Sender: TObject);
procedure TreeViewItem9Click(Sender: TObject);
procedure TreeViewItem10Click(Sender: TObject);
procedure TreeViewItem11Click(Sender: TObject);
procedure TreeViewItem12Click(Sender: TObject);
```

```
procedure TreeViewItem13Click(Sender: TObject);
procedure TreeViewItem14Click(Sender: TObject);
procedure TreeViewItem15Click(Sender: TObject);
procedure TreeViewItem16Click(Sender: TObject);
procedure TreeViewItem17Click(Sender: TObject);
procedure TreeViewItem18Click(Sender: TObject);
procedure TreeViewItem19Click(Sender: TObject);
procedure TreeViewItem20Click(Sender: TObject);
procedure TreeViewItem21Click(Sender: TObject);
procedure TreeViewItem22Click(Sender: TObject);
procedure TreeViewItem23Click(Sender: TObject);
procedure TreeViewItem24Click(Sender: TObject);
procedure TreeViewItem25Click(Sender: TObject);
procedure TreeViewItem27Click(Sender: TObject);
procedure TreeViewItem28Click(Sender: TObject);
procedure TreeViewItem29Click(Sender: TObject);
procedure TreeViewItem30Click(Sender: TObject);
procedure TreeViewItem31Click(Sender: TObject);
procedure TreeViewItem32Click(Sender: TObject);
procedure TreeViewItem33Click(Sender: TObject);
procedure TreeViewItem34Click(Sender: TObject);
procedure TreeViewItem35Click(Sender: TObject);
procedure TreeViewItem36Click(Sender: TObject);
procedure TreeViewItem37Click(Sender: TObject);
procedure TreeViewItem38Click(Sender: TObject);
procedure TreeViewItem39Click(Sender: TObject);
procedure TreeViewItem40Click(Sender: TObject);
procedure TreeViewItem41Click(Sender: TObject);
procedure TreeViewItem42Click(Sender: TObject);
procedure TreeViewItem43Click(Sender: TObject);
procedure TreeViewItem44Click(Sender: TObject);
procedure TreeViewItem45Click(Sender: TObject);
procedure TreeViewItem46Click(Sender: TObject);
procedure TreeViewItem47Click(Sender: TObject);
procedure TreeViewItem48Click(Sender: TObject);
procedure TreeViewItem49Click(Sender: TObject);
procedure TreeViewItem50Click(Sender: TObject);
procedure TreeViewItem52Click(Sender: TObject);
procedure TreeViewItem53Click(Sender: TObject);
procedure TreeViewItem54Click(Sender: TObject);
procedure TreeViewItem55Click(Sender: TObject);
procedure TreeViewItem56Click(Sender: TObject);
procedure TreeViewItem57Click(Sender: TObject);
procedure TreeViewItem58Click(Sender: TObject);
procedure TreeViewItem59Click(Sender: TObject);
procedure TreeViewItem60Click(Sender: TObject);
procedure TreeViewItem61Click(Sender: TObject);
procedure TreeViewItem62Click(Sender: TObject);
procedure TreeViewItem63Click(Sender: TObject);
procedure TreeViewItem64Click(Sender: TObject);
procedure TreeViewItem65Click(Sender: TObject);
procedure TreeViewItem66Click(Sender: TObject);
```

```
    procedure TreeViewItem68Click(Sender: TObject);
    procedure TreeViewItem70Click(Sender: TObject);
    procedure TreeViewItem71Click(Sender: TObject);
    procedure TreeViewItem73Click(Sender: TObject);
    procedure TreeViewItem74Click(Sender: TObject);
    procedure Pit_fClick(Sender: TObject);
    procedure TreeViewItem77Click(Sender: TObject);
    procedure TreeViewItem78Click(Sender: TObject);
    procedure TreeViewItem79Click(Sender: TObject);
    procedure TreeViewItem76Click(Sender: TObject);
    procedure TreeViewItem80Click(Sender: TObject);
    procedure TreeViewItem81Click(Sender: TObject);
    procedure TreeViewItem83Click(Sender: TObject);
    procedure TreeViewItem84Click(Sender: TObject);
    procedure TreeViewItem85Click(Sender: TObject);
    procedure TreeViewItem86Click(Sender: TObject);
    procedure TreeViewItem87Click(Sender: TObject);
    procedure TreeViewItem88Click(Sender: TObject);
    procedure TreeViewItem90Click(Sender: TObject);
    procedure TreeViewItem91Click(Sender: TObject);
    procedure TreeViewItem92Click(Sender: TObject);
    procedure TreeViewItem93Click(Sender: TObject);
    procedure TreeViewItem100Click(Sender: TObject);
    procedure TreeViewItem101Click(Sender: TObject);
    procedure TreeViewItem102Click(Sender: TObject);
    procedure TreeViewItem104Click(Sender: TObject);
    procedure TreeViewItem105Click(Sender: TObject);
    procedure TreeViewItem106Click(Sender: TObject);
    procedure TreeViewItem107Click(Sender: TObject);
    procedure TreeViewItem95Click(Sender: TObject);
    procedure TreeViewItem96Click(Sender: TObject);
    procedure TreeViewItem97Click(Sender: TObject);
    procedure TreeViewItem98Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  fmFileDataBrowser: TfmFileDataBrowser;

//=============================================================
implementation
//=============================================================

{$R *.fmx}

procedure TfmFileDataBrowser.Button1Click(Sender: TObject);
begin
  inherited;
Form1.Show;
```

```
end;

procedure TfmFileDataBrowser.FormCreate(Sender: TObject);
begin
  inherited;
  //
end;

procedure TfmFileDataBrowser.FormShow(Sender: TObject);
begin
  inherited;
  LoadLangFromStrings(Langs.LangStr[CurLang]);
  Caption := Translate(Caption);
end;


procedure OpenBase(database, table: string);
begin
 Form1.FDConnection1.Connected := false;
  Form1.FDTable1.Active := false;
  Form1.FDConnection1.Params.Database := database;
  Form1.FDTable1.TableName := table;
  Form1.FDConnection1.Connected := true;
  Form1.FDTable1.Active := true;
  Form1.Show;
end;

procedure TfmFileDataBrowser.TreeViewItem100Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Base\Modeling\Solids.sdb','orebody_c');
end;

procedure TfmFileDataBrowser.TreeViewItem101Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Base\Modeling\Solids.sdb','orebody_f');
end;

procedure TfmFileDataBrowser.TreeViewItem102Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Base\Modeling\Solids.sdb','orebody_v');
end;

procedure TfmFileDataBrowser.TreeViewItem104Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Modeling\Tins.sdb','pit_f');
end;

procedure TfmFileDataBrowser.TreeViewItem105Click(Sender: TObject);
```

```
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Modeling\Tins.sdb','pit_v');
end;

procedure TfmFileDataBrowser.TreeViewItem106Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Modeling\Tins.sdb','relief_f');
end;

procedure TfmFileDataBrowser.TreeViewItem107Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Modeling\Tins.sdb','relief_v');
end;

procedure TfmFileDataBrowser.TreeViewItem10Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','lines');
end;

procedure TfmFileDataBrowser.TreeViewItem11Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','mainminerals');
end;

procedure TfmFileDataBrowser.TreeViewItem12Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','minerals');
end;

procedure TfmFileDataBrowser.TreeViewItem13Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','moos');
end;

procedure TfmFileDataBrowser.TreeViewItem14Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','ores');
end;

procedure TfmFileDataBrowser.TreeViewItem15Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','oresorts');
end;
```

```
procedure TfmFileDataBrowser.TreeViewItem16Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','patterns');
end;

procedure TfmFileDataBrowser.TreeViewItem17Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','period');
end;

procedure TfmFileDataBrowser.TreeViewItem18Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','phystech');
end;

procedure TfmFileDataBrowser.TreeViewItem19Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','planets');
end;

procedure TfmFileDataBrowser.TreeViewItem20Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','pointtypes');
end;

procedure TfmFileDataBrowser.TreeViewItem21Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','polytypes');
end;

procedure TfmFileDataBrowser.TreeViewItem22Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','rocktypes');
end;

procedure TfmFileDataBrowser.TreeViewItem23Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','taxrates');
end;

procedure TfmFileDataBrowser.TreeViewItem24Click(Sender: TObject);
begin
  inherited;
```

```
  OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','textures');
end;

procedure TfmFileDataBrowser.TreeViewItem25Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','units');
end;

procedure TfmFileDataBrowser.TreeViewItem27Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','age');
end;

procedure TfmFileDataBrowser.TreeViewItem28Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','chemics');
end;

procedure TfmFileDataBrowser.TreeViewItem29Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','default');
end;

procedure TfmFileDataBrowser.TreeViewItem2Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','age');
end;

procedure TfmFileDataBrowser.TreeViewItem30Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','elementclass');
end;

procedure TfmFileDataBrowser.TreeViewItem31Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','elements');
end;

procedure TfmFileDataBrowser.TreeViewItem32Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','eon');
end;

procedure TfmFileDataBrowser.TreeViewItem33Click(Sender: TObject);
```

```
begin
  inherited;
    OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','epoch');
end;

procedure TfmFileDataBrowser.TreeViewItem34Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','era');
end;

procedure TfmFileDataBrowser.TreeViewItem35Click(Sender: TObject);
begin
  inherited;
   OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','lines');
end;

procedure TfmFileDataBrowser.TreeViewItem36Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','mainminerals');
end;

procedure TfmFileDataBrowser.TreeViewItem37Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','minerals');
end;

procedure TfmFileDataBrowser.TreeViewItem38Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','moos');
end;

procedure TfmFileDataBrowser.TreeViewItem39Click(Sender: TObject);
begin
  inherited;
   OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','ores');
end;

procedure TfmFileDataBrowser.TreeViewItem3Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','chemics');
end;

procedure TfmFileDataBrowser.TreeViewItem40Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','oresorts');
end;
```

```
procedure TfmFileDataBrowser.TreeViewItem41Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','patterns');
end;

procedure TfmFileDataBrowser.TreeViewItem42Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','period');
end;

procedure TfmFileDataBrowser.TreeViewItem43Click(Sender: TObject);
begin
  inherited;
   OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','phystech');
end;

procedure TfmFileDataBrowser.TreeViewItem44Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','planets');
end;

procedure TfmFileDataBrowser.TreeViewItem45Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','pointtypes');
end;

procedure TfmFileDataBrowser.TreeViewItem46Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','polytypes');
end;

procedure TfmFileDataBrowser.TreeViewItem47Click(Sender: TObject);
begin
  inherited;
    OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','rocktypes');
end;

procedure TfmFileDataBrowser.TreeViewItem48Click(Sender: TObject);
begin
  inherited;
    OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','taxrates');
end;

procedure TfmFileDataBrowser.TreeViewItem49Click(Sender: TObject);
begin
  inherited;
```

```
 OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','textures');
end;

procedure TfmFileDataBrowser.TreeViewItem4Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','default');
end;



procedure TfmFileDataBrowser.TreeViewItem50Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Reference\Reference_eng.sdb','units');
end;

procedure TfmFileDataBrowser.TreeViewItem52Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Exploration\Exploration.sdb','Assays');
end;

procedure TfmFileDataBrowser.TreeViewItem53Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Base\Exploration\Exploration.sdb','Collars');
end;

procedure TfmFileDataBrowser.TreeViewItem54Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Base\Exploration\Exploration.sdb','DirectResection');
end;

procedure TfmFileDataBrowser.TreeViewItem55Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Base\Exploration\Exploration.sdb','Framework');
end;

procedure TfmFileDataBrowser.TreeViewItem56Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Base\Exploration\Exploration.sdb','GeodeticStations');
end;

procedure TfmFileDataBrowser.TreeViewItem57Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Base\Exploration\Exploration.sdb','Horizons');
end;
```

```
procedure TfmFileDataBrowser.TreeViewItem58Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Base\Exploration\Exploration.sdb','Inclins');
end;

procedure TfmFileDataBrowser.TreeViewItem59Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Base\Exploration\Exploration.sdb','Instruments');
end;

procedure TfmFileDataBrowser.TreeViewItem5Click(Sender: TObject);
begin
  inherited;
    OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','elementclass');
end;

procedure TfmFileDataBrowser.TreeViewItem60Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Exploration\Exploration.sdb','Lithos');
end;

procedure TfmFileDataBrowser.TreeViewItem61Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Base\Exploration\Exploration.sdb','Organization');
end;

procedure TfmFileDataBrowser.TreeViewItem62Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Base\Exploration\Exploration.sdb','SurveyStations');
end;

procedure TfmFileDataBrowser.TreeViewItem63Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Base\Exploration\Exploration.sdb','Surveyors');
end;

procedure TfmFileDataBrowser.TreeViewItem64Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Base\Exploration\Exploration.sdb','TacheoMeasurements');
end;

procedure TfmFileDataBrowser.TreeViewItem65Click(Sender: TObject);
begin
  inherited;
```

```
 OpenBase('D:\Gexoblock\Data\Base\Exploration\Exploration.sdb','TacheoStations');
end;

procedure TfmFileDataBrowser.TreeViewItem66Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Base\Exploration\Exploration.sdb','TacheoTitle');
end;

procedure TfmFileDataBrowser.TreeViewItem68Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Modelling\Dholes.sdb','assays');
end;

procedure TfmFileDataBrowser.TreeViewItem6Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','elements');
end;

procedure TfmFileDataBrowser.TreeViewItem70Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Modelling\Grids2D.sdb','pit');
end;

procedure TfmFileDataBrowser.TreeViewItem71Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Base\Modelling\Grids2D.sdb','relief');
end;

procedure TfmFileDataBrowser.TreeViewItem73Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Base\Modelling\Grids3D.sdb','marvin');
end;

procedure TfmFileDataBrowser.TreeViewItem74Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Base\Modelling\Grids3D.sdb','points3d');
end;

procedure TfmFileDataBrowser.TreeViewItem76Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Modelling\Meshes3D.sdb','orebody_c');
end;

procedure TfmFileDataBrowser.TreeViewItem77Click(Sender: TObject);
```

```
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Modelling\Meshes2D.sdb','pit_v');
end;

procedure TfmFileDataBrowser.TreeViewItem78Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Modelling\Meshes2D.sdb','relief_f');
end;

procedure TfmFileDataBrowser.TreeViewItem79Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Base\Modelling\Meshes2D.sdb','relief_v');
end;

procedure TfmFileDataBrowser.TreeViewItem7Click(Sender: TObject);
begin
  inherited;
   OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','eon');
end;

procedure TfmFileDataBrowser.TreeViewItem80Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Modelling\Meshes3D.sdb','orebody_f');
end;

procedure TfmFileDataBrowser.TreeViewItem81Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Base\Modelling\Meshes3D.sdb','orebody_v');
end;

procedure TfmFileDataBrowser.TreeViewItem83Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Modelling\Points2D.sdb','blocks');
end;

procedure TfmFileDataBrowser.TreeViewItem84Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Modelling\Points2D.sdb','civil');
end;

procedure TfmFileDataBrowser.TreeViewItem85Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Base\Modelling\Points2D.sdb','hor10');
end;
```

```
procedure TfmFileDataBrowser.TreeViewItem86Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Base\Modelling\Points2D.sdb','pit');
end;

procedure TfmFileDataBrowser.TreeViewItem87Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Base\Modelling\Points2D.sdb','points2d');
end;

procedure TfmFileDataBrowser.TreeViewItem88Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Base\Modelling\Points2D.sdb','relief');
end;

procedure TfmFileDataBrowser.TreeViewItem8Click(Sender: TObject);
begin
  inherited;
  OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','epoch');
end;

procedure TfmFileDataBrowser.TreeViewItem90Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Modelling\Points3D.sdb','assays');
end;

procedure TfmFileDataBrowser.TreeViewItem91Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Modelling\Points3D.sdb','points3d');
end;

procedure TfmFileDataBrowser.TreeViewItem92Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Modelling\Points3D.sdb','points3dvor');
end;

procedure TfmFileDataBrowser.TreeViewItem93Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Modelling\Points3D.sdb','test');
end;

procedure TfmFileDataBrowser.TreeViewItem95Click(Sender: TObject);
begin
  inherited;
```

```
OpenBase('D:\Gexoblock\Data\Base\Modelling\Polygons.sdb','delfin_f');
end;

procedure TfmFileDataBrowser.TreeViewItem96Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Modelling\Polygons.sdb','delfin_v');
end;

procedure TfmFileDataBrowser.TreeViewItem97Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Modelling\Polygons.sdb','pit_f');
end;

procedure TfmFileDataBrowser.TreeViewItem98Click(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Modelling\Polygons.sdb','pit_v');
end;

procedure TfmFileDataBrowser.TreeViewItem9Click(Sender: TObject);
begin
  inherited;
 OpenBase('D:\Gexoblock\Data\Reference\Reference_rus.sdb','era');
end;

procedure TfmFileDataBrowser.Pit_fClick(Sender: TObject);
begin
  inherited;
OpenBase('D:\Gexoblock\Data\Base\Modelling\Meshes2D.sdb','pit_f');
end;

end.
```

Модуль dDialogs

```
unit dDialogs;
interface
uses
  System.SysUtils,
  System.Classes,
  FMX.Types,
  FMX.Dialogs,
  FMX.Printer;

type
  TdmDialogs = class(TDataModule)
    OpenDialog: TOpenDialog;
    SaveDialog: TSaveDialog;
    PrintDialog: TPrintDialog;
    PrinterSetupDialog: TPrinterSetupDialog;
```

```
    PageSetupDialog: TPageSetupDialog;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  dmDialogs: TdmDialogs;
implementation
{%CLASSGROUP 'FMX.Controls.TControl'}
{$R *.dfm}
end.
```

Модуль HelpAbout

```
unit fHelpAbout;
interface
uses
  System.SysUtils,
  System.Types,
  System.UITypes,
  System.Classes,
  System.Variants,
  System.Math.Vectors,
  System.ImageList,
  FMX.ImgList,
  FMX.Menus,
  FMX.Types,
  FMX.Controls,
  FMX.Forms,
  FMX.Dialogs,
  FMX.Objects,
  FMX.Types3D,
  FMX.Layers3D,
  FMX.Controls3D,
  FMX.StdCtrls,
  FMX.Layouts,
  FMX.ListBox,
  FMX.Controls.Presentation,

  fInitialDialog,
  uGlobals;

type
  TfmHelpAbout = class(TfmInitialDialog)
    TextTitle: TText;
    LabelSF: TLabel;
    Label4: TLabel;
    Image3D1: TImage3D;
    Image1: TImage;
    LabelBlockModelling: TLabel;
    LabelDataVisualization: TLabel;
```

```
    LabelReserveCalculation: TLabel;
    Langs: TLang;
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  fmHelpAbout: TfmHelpAbout;

//=====================================================================
implementation
//=====================================================================

{$R *.fmx}

procedure TfmHelpAbout.FormShow(Sender: TObject);
begin
  inherited;
  Langs.Lang := CurLang;
  Caption := TransManually(Caption, Langs);
end;

end.
```

Модуль InitialDialog

```
unit fInitialDialog;

interface

uses
  System.SysUtils,
  System.Types,
  System.UITypes,
  System.Classes,
  System.Variants,
  FMX.Types,
  FMX.Controls,
  FMX.Forms,
  FMX.Dialogs,
  FMX.StdCtrls,
  FMX.Controls.Presentation,

  fInitialForm, FMX.Menus;

type
  TfmInitialDialog = class(TfmInitialForm)
    PanelTop: TPanel;
    PanelMiddle: TPanel;
    PanelBottom: TPanel;
    ButtonOK: TButton;
```

```
    ButtonCancel: TButton;
    ButtonHelp: TButton;
    procedure ButtonHelpClick(Sender: TObject);
  private
    { Private declarations }
  public
    function Execute: boolean; virtual;
  end;

var
  fmInitialDialog: TfmInitialDialog;

//=====================================================================
==
implementation
//=====================================================================
==

{$R *.fmx}

procedure TfmInitialDialog.ButtonHelpClick(Sender: TObject);
begin
  inherited;
  //in VCL Application.HelpContext(HelpContext);
  ButtonHelp.HelpContext := 0; //??? HelpContext;
end;

function TfmInitialDialog.Execute: boolean;
begin
  Result := ShowModal = mrOk;
end;

end.
```

Модуль InitialForm

```
unit fInitialForm;

interface

uses
  System.SysUtils,
  System.Types,
  System.UITypes,
  System.Classes,
  System.Variants,
  System.ImageList,
  FMX.Types,
  FMX.Controls,
  FMX.Forms,
  FMX.Dialogs,
  FMX.ImgList,
  FMX.Menus;
```

```
type
  TfmInitialForm = class(TForm)
  private
    { Private declarations }
  public
    procedure ReadIniFile; virtual;
    procedure WriteIniFile; virtual;
    { translate a text,which is not in the autotranslation process of TLang}
    function TransManually(sOriginal: string; aLang: TLang): string; virtual;
    procedure LoadTranslationsFromFile(aLangCode: string; aFileName: string;
      aLang: TLang); virtual;
  end;

var
  fmInitialForm: TfmInitialForm;

// =====================================================================
implementation
// =====================================================================

{$R *.fmx}

function TfmInitialForm.TransManually(sOriginal: string; aLang: TLang): string;
begin
  // first check, if the selected language has a mapping
  if aLang.Resources.IndexOf(aLang.Lang) >= 0 then
  begin
    // get the resource stringlist of the current language and get the translation
    result := aLang.LangStr[aLang.Lang].Values[sOriginal];
    if result = '' then
    begin
      // text not found in mapping - keep original text
      result := sOriginal;
    end;
  end
  else
  begin // language not found: must be the default language
    result := sOriginal;
  end;
end;

procedure TfmInitialForm.LoadTranslationsFromFile(aLangCode: string;
  aFileName: string; aLang: TLang);
var
  TranStrings: TStrings;
begin
  if (aLangCode <> '') and (FileExists(aFileName)) and (Assigned(aLang))
  then
  begin
    TranStrings := TStringList.Create;
    TranStrings.LoadFromFile(aFileName);
    aLang.Resources.AddObject(aLangCode, TranStrings);
```

```
  end;
end;

{ TfmInitialForm }

procedure TfmInitialForm.ReadIniFile;
begin
  //
end;

procedure TfmInitialForm.WriteIniFile;
begin
  //
end;

initialization

end.
```

Модуль MethodDialog

```
unit fMethodDialog;

interface

uses
  System.SysUtils,
  System.Types,
  System.UITypes,
  System.Classes,
  System.Variants,
  System.ImageList,
  FMX.Types,
  FMX.Graphics,
  FMX.Controls,
  FMX.Forms,
  FMX.Dialogs,
  FMX.StdCtrls,
  FMX.Controls.Presentation,
  FMX.Layouts,
  FMX.ListBox,
  FMX.ImgList,
  FMX.Edit,

  fInitialDialog, FMX.Menus;

type
  TfmMethodDialog = class(TfmInitialDialog)
    ImageListInput: TImageList;
    ToolBar2: TToolBar;
    sbMap: TSpeedButton;
    ImageListOutput: TImageList;
    sbPlot: TSpeedButton;
    sbTable: TSpeedButton;
```

```
    LabelEmpty: TLabel;
    EditOutput: TEdit;
    LabelOutput: TLabel;
    ProgressBar: TProgressBar;
    PanelInputA: TPanel;
    LabelInputA: TLabel;
    ToolBarInputA: TToolBar;
    sbDholes: TSpeedButton;
    sbPoints2D: TSpeedButton;
    sbPoints3D: TSpeedButton;
    sbPolygons: TSpeedButton;
    sbTINs: TSpeedButton;
    sbSolids: TSpeedButton;
    sbGrids2D: TSpeedButton;
    sbGrids3D: TSpeedButton;
    sbMeshes2D: TSpeedButton;
    sbMeshes3D: TSpeedButton;
    PanelAttributes: TPanel;
    LabelAttributesA: TLabel;
    ListBoxAttributesA: TListBox;
    PanelModels: TPanel;
    LabelModelsA: TLabel;
    ListBoxInputA: TListBox;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  fmMethodDialog: TfmMethodDialog;

//=============================================================
implementation
//=============================================================

{$R *.fmx}

end.
```

Модуль DualDialog

```
unit fMethodDualDialog;

interface

uses
  System.SysUtils,
  System.Types,
  System.UITypes,
  System.Classes,
  System.Variants,
  System.ImageList,
  FMX.Types,
```

```
  FMX.Graphics,
  FMX.Controls,
  FMX.Forms,
  FMX.Dialogs,
  FMX.StdCtrls,
  FMX.ImgList,
  FMX.Edit,
  FMX.Layouts,
  FMX.ListBox,
  FMX.Controls.Presentation,

  fMethodDialog;

type
  TfmMethodDualDialog = class(TfmMethodDialog)
    Panel1: TPanel;
    LabelInputB: TLabel;
    ToolBar1: TToolBar;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    SpeedButton5: TSpeedButton;
    SpeedButton6: TSpeedButton;
    SpeedButton7: TSpeedButton;
    SpeedButton8: TSpeedButton;
    SpeedButton9: TSpeedButton;
    SpeedButton10: TSpeedButton;
    Panel2: TPanel;
    LabelAttributesB: TLabel;
    ListBoxAttributesB: TListBox;
    Panel3: TPanel;
    LabelModelsB: TLabel;
    ListBoxModelsB: TListBox;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  fmMethodDualDialog: TfmMethodDualDialog;

//=====================================================================
implementation
//=====================================================================

{$R *.fmx}

end.
```

Модуль MethodGridGeneration

```
unit fMethodGridGeneration;
```

```
interface

uses
  System.SysUtils,
  System.Types,
  System.UITypes,
  System.Classes,
  System.Variants,
  FMX.Types,
  FMX.Graphics,
  FMX.Controls,
  FMX.Forms,
  FMX.Dialogs,
  FMX.StdCtrls,
  fMethodDialog,
  System.ImageList,
  FMX.ImgList,
  FMX.Edit,
  FMX.Layouts,
  FMX.ListBox,
  FMX.Controls.Presentation,

  uGlobals, FMX.Objects;


type
  TfmMethodGridGeneration = class(TfmMethodDialog)
    Langs: TLang;
    gbModelSize: TGroupBox;
    Image1: TImage;
    Image2: TImage;
    Image3: TImage;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  fmMethodGridGeneration: TfmMethodGridGeneration;

//=====================================================
implementation
//=====================================================

{$R *.fmx}

procedure TfmMethodGridGeneration.FormCreate(Sender: TObject);
begin
  inherited;
```

```
  Langs.Lang := CurLang;
  Caption := TransManually(Caption, Langs);

end;

end.
```

Модуль MethodInterpolation

```
unit fMethodInterpolation;
interface
uses
  System.SysUtils,
  System.Types,
  System.UITypes,
  System.Classes,
  System.Variants,
  System.ImageList,
  FMX.Types,
  FMX.Graphics,
  FMX.Controls,
  FMX.Forms,
  FMX.Dialogs,
  FMX.StdCtrls,
  fMethodDualDialog,
  FMX.ImgList,
  FMX.Edit,
  FMX.Layouts,
  FMX.ListBox,
  FMX.Controls.Presentation,
  FMX.Menus,

  uGlobals,
  fInitialForm;

type
  TfmMethodInterpolation = class(TfmMethodDualDialog)
    Panel4: TPanel;
    LabelMethod: TLabel;
    RadioButtonIDW: TRadioButton;
    RadioButtonCPI: TRadioButton;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    RadioButton3: TRadioButton;
    RadioButtonLinearByTIN: TRadioButton;
    ButtonOptions: TButton;
    CheckBoxOpenCL: TCheckBox;
    CheckBoxVariance: TCheckBox;
    Langs: TLang;
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
```

```
  end;

var
  fmMethodInterpolation: TfmMethodInterpolation;

//=========================================================
implementation
//=========================================================

{$R *.fmx}

procedure TfmMethodInterpolation.FormShow(Sender: TObject);
begin
  inherited;
  Langs.Lang := CurLang;
  Caption := TransManually(Caption, Langs);
end;

end.
```

Модуль ToolsOptions

```
unit fToolsOptions;
interface
uses
  System.SysUtils,
  System.Types,
  System.UITypes,
  System.Rtti,
  System.Classes,
  System.Variants,
  FMX.Types,
  FMX.Controls,
  FMX.Forms,
  FMX.Dialogs,
  FMX.TreeView,
  FMX.Layouts,
  FMX.Grid,
  FMX.ListBox,
  FMX.StdCtrls,
  FMX.Controls.Presentation,
  FMX.Menus,

  fInitialDialog,
  uGlobals;

type
  TfmToolsOptions = class(TfmInitialDialog)
    TreeView: TTreeView;
    TreeViewItem1: TTreeViewItem;
    TreeViewItem2: TTreeViewItem;
    TreeViewItem3: TTreeViewItem;
    ListBox: TListBox;
```

```
    ButtonEn: TButton;
    ButtonRu: TButton;
    procedure FormCreate(Sender: TObject);
    procedure ListBoxChange(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  fmToolsOptions: TfmToolsOptions;

//=============================================================================
========
implementation
//=============================================================================
========

{$R *.fmx}

procedure TfmToolsOptions.FormCreate(Sender: TObject);
var
  i: integer;
  Item: TListBoxItem;
begin
  inherited;
  Item := TListBoxItem.Create(Self);
  Item.Parent := ListBox;
  Item.Text := 'en';
  Item.AutoTranslate := True;

{
  Langs.Resources.LoadFromFile('..\Gexoblock.lng');
  for i := 0 to Langs.Resources.Count - 1 do
  begin
    Item := TListBoxItem.Create(Self);
    Item.AutoTranslate := True;
    Item.Parent := ListBox;
    Item.Text := Langs.Resources[i];
    if Langs.Lang = Item.Text then
      ListBox.ItemIndex := ListBox.Count - 1;
  end;
}
end;

procedure TfmToolsOptions.ListBoxChange(Sender: TObject);
begin
{
  if ListBox.Selected <> nil then
    Langs.Lang := ListBox.Selected.Text;
}
```

```
end;
end.
```

Выпускная квалификационная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

«___» _____ _____ г.

_____ _____

_

(подпись) *(Ф.И.О.)*