

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

УФИМСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ИНФОРМАТИКИ И РОБОТОТЕХНИКИ

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

Направление 09.03.04 – Программная инженерия

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Тема: Программа для предупреждения пересечения стволов
скважин

	ФИО	Подпись	Дата
Студент	Синявский Г. Н.		
Руководитель работы	Еникеева К. Р.		
Консультант	Еникеева К. Р.		
Контроль программного продукта			
Председатель комиссии по предзащите			
Рецензент			

Допущен к защите
Зав. кафедрой ВМК, д.т.н., проф.
_____ Н.И. Юсупова
“ _____ ” _____ 2016 г.

УФА — 2016г.

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

УФИМСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ИНФОРМАТИКИ И РОБОТОТЕХНИКИ

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

”УТВЕРЖДАЮ”

Зав. кафедрой ВМК, д.т.н., проф.

Н.И. Юсупова

«08» октября 2015 г.

ЗАДАНИЕ

на подготовку выпускной квалификационной работы

студента Синявского Глеба Николаевича

1. Тема работы - Программа для предупреждения пересечения стволов скважин
(утверждена распоряжением по факультету № 72/1 от “ 15 октября 2015 г)
2. Срок представления работы “20” января 2016 г.
3. Описание задачи
Необходимо разработать программный продукт, позволяющий усреднять и визуализировать замеры стволов скважин, а так же позволять оценивать расстояния между ними.
4. Математическая часть
В качестве математических моделей, принятых для реализации в рамках программного продукта, необходимо использовать уравнения для расчета координат ствола методом среднего угла.
5. Спецификация входных и выходных данных
Входные данные - csv-файлы, содержащие результаты замера ствола скважины. Выходные - визуализация скважины в пространстве, визуализации оценки расстояний между стволами.
6. Применяемые инструментальные средства
Библиотека построение графического интерфейса - Qt. СУБД - SQLite. Библиотека визуализации - MathGL.

7. Особые условия эксплуатации программного продукта

Основная ОС для запуска программного продукта - Windows 7 и старше, но продукт должен разрабатываться как кроссплатформенный и иметь возможность запуска под управлением ОС Linux.

8. Дополнительные условия

Продукт должен иметь возможность импортировать csv произвольного формата, для этого должен быть разработан мастер импорта, позволяющий выбирать диапазон ячеек таблицы и указывать их тип.

Руководитель работы _____

Консультант _____

Дата выдачи «08» октября 2015 г.

Оглавление

Аннотация	5
Введение	6
Описание предметной области	6
Мотивация, актуальность проблемы	6
Цели, задачи ВКР	7
Содержание работы по главам	8
1. Анализ проблемы и постановка задачи	9
1.1 Анализ предметной области	9
1.2 Содержательная постановка проблемы	11
1.3 Формальная постановка задачи	11
1.4 Структура решения задачи, декомпозиция задачи на подзадачи	12
2. Математическое и информационное обеспечение	13
2.1 Классификация подзадач (отнесение подзадач к классу задач)	13
2.2 Математические модели подзадач (где применимо)	13
2.3 Методы решения подзадач (где применимо)	13
2.4 Информационные модели для подзадач (где применимо)	13
2.5 Алгоритмы и структуры данных для подзадач	13
3. Программное обеспечение	14
3.1 Аналитический обзор существующих программных технологий, применимых при решении поставленных задач	14
3.2 Архитектура разрабатываемого программного продукта	15
3.3 Язык программирования и инструментальные средства разработки	15
3.3.1 Язык C++	15
3.3.2 SQLite	15
3.3.3 Qt	15
3.3.4 MathGL	15
3.3.5 Обоснованность выбора технологий	15
3.4 Технологии разработки ПО (моделирование разработки ПО, управление разработкой ПО, конфигурирование ПО, технологии тестирования ПО)	16
3.5 Описание структуры программного продукта	16
3.6 Описание интерфейса пользователя	16
4. Оценка качества решения	17
4.1 Тестирование ПО	17
4.2 Оценка качества программного продукта	19
4.3 Вычислительный эксперимент и анализ результатов	30
ЗАКЛЮЧЕНИЕ	31
Результаты работы	31
Выводы	31
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	32

ПРИЛОЖЕНИЯ	33
Программная документация	33
Техническое задание	33
Руководство программиста	35
Руководство пользователя	36

Аннотация

Представленная дипломная работа содержит описание программного решения задачи предупреждения пересечения стволов скважин.

В работе было проведено исследование данной задачи, ее анализ и декомпозиция на подзадачи. В соответствии с полученной структурой была разработана информационная и функциональные модели решения задачи.

В рамках дипломной работы было разработано и внедрено в эксплуатацию программное обеспечение, реализующее поставленную задачу.

В соответствующих разделах работы приводится описание технологической базы созданного продукта, анализируются имеющиеся аналоги, описывается структура и процесс функционирования созданного программного обеспечения.

Введение

При бурении близкорасположенных скважин очень остро стоит проблема предотвращения пересечения стволов скважин. Связано это, в первую очередь с финансовыми и временными расходами на ликвидацию последствий такой аварии: герметизация пересекаемой скважины, ликвидация участка пересекающей и восстановление траектории пересекаемой скважин. При бурении с кустовых площадок риск пересечения стволов минимизируется бурением скважин, отдаляющихся друг от друга. Такой способ предотвращения пересечения стволов не подходит при бурении замораживающих скважин, которые должны располагаться на строго определенном расстоянии друг от друга.

Описание предметной области

Замораживание грунтов — искусственное охлаждение грунтов в естественном залегании до отрицательных температур с целью их упрочения и достижения необходимой степени водонепроницаемости.

Замораживание грунтов возможно при различных глубинах, сочетаниях грунтов, скоростях движения грунтовых вод и степени их минерализации. Замораживание грунтов — основной способ при работе в сложных гидрогеологических условиях как при замораживании водоносных рыхлых, так и водоносных трещиноватых пород.

Замораживание грунтов применяют при возведении фундаментов зданий и сооружений, строительстве шахт, тоннелей, метрополитенов, противofильтрационных завес, мостов, плотин, подземных хранилищ и др.

Для охлаждения грунта используют холодильные установки с системой погружаемых в грунт труб (замораживающих колонок), по которым циркулирует холодоноситель, охлажденный до -20 -40°C (рассольный способ замораживания), или хладагент, который непосредственно испаряется в замораживающей колонке при температуре от -35 до -196°C (безрассольный способ замораживания). В качестве холодоносителя применяют водные растворы солей (например, хлориды кальция, натрия, лития) или специальные жидкости, которые замерзают при низких температурах, а в качестве хладагента — аммиак, углекислоту, фреон и др. В процессе непрерывного теплообмена холодоносителя (хладагента) с грунтом вокруг каждой трубы образуются ледопородные цилиндры, которые в дальнейшем смыкаются, образуя замкнутое ледопородное ограждение по контуру подземного сооружения или массив замороженного грунта.

Для предупреждения протекания ледопородного ограждения необходимо строго контролировать расстояния между замораживающими скважинами не допускать их пересечения.

Мотивация, актуальность проблемы

Решения о написании собственного продукта связано со сложностью и дороговизной существующих решений

Таблица 1: Существующие решения

Название продукта	Недостатки	Стоимость
-------------------	------------	-----------

Таблица 1: Существующие решения

Название продукта	Недостатки	Стоимость
Landmark Compass	<ul style="list-style-type: none"> • Стоимость • Необходимость обучения сотрудников • Сложность внедрения и поддержки • Избыточный функционал 	Порядка \$170000
Paradigm Sysdrill	<ul style="list-style-type: none"> • Стоимость • Необходимость обучения сотрудников • Избыточный функционал 	Порядка \$150000

Цели, задачи ВКР

Целью дипломной работы является разработка программного обеспечения, позволяющего визуализировать, усреднять и производить анализ замеров стволов скважин, на основании данных, полученных с измерительного оборудования. Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ существующих программных продуктов;
- разработка функциональной и информационной моделей, программного обеспечения;
- разработка модуля импорта данных
- разработка системы управления содержимым БД и усреднением замеров
- разработка модуля визуализации замеров
- разработка модуля расчетов расстояний между стволами
- разработка модуля визуализации расстояний между стволами

Содержание работы по главам

- **Глава 1** - анализ предметной области и постановка задачи;
- **Глава 2** - описание математического и информационного обеспечения, методы решения;
- **Глава 3** - описание архитектуры и используемых технологий;
- **Глава 4** - оценка качества решения;

1. Анализ проблемы и постановка задачи

1.1 Анализ предметной области

Бурение скважин — это процесс сооружения направленной цилиндрической горной выработки в земле, диаметр "D" которой мал по сравнению с её длиной по стволу "H", без доступа человека на забой. Начало скважины на поверхности земли называют устьем, дно — забоем, а стенки скважины образуют ее ствол.

Кустовое бурение — сооружение скважин (в основном наклонно направленных), устья которых группируются на близком расстоянии друг от друга на общей ограниченной площадке (основании). Применяется при разработке месторождений под застроенными участками, при разработке нефтяных и газовых месторождений в определённых климатических условиях (например, в зимний период, когда наблюдается большой снеговой покров, или весной во время распутицы и значительных паводков), на территории с сильно пересечённым рельефом местности или в пределах акваторий.

В настоящее время практически все эксплуатационные скважины бурятся кустовым методом, когда устья нескольких скважин в кусте расположены близко друг к другу (4–5 м) на одной технологической площадке, а забои находятся в узлах сетки разработки. Число скважин в кусте колеблется от 2 до нескольких десятков.

Искусственное искривление скважин применяется с целью:

- добычи нефти и газа из труднодоступных участков, занятых на поверхности промышленными и жилыми объектами, оврагами, горами, реками, озерами, болотами, лесами, морями;
- экономии отводимых под строительство скважин плодородных земельных участков, лесов и др.;
- экономии затрат на строительство оснований, подъездных путей, линий электропередач, связи, трубопроводов;
- сокращения средств и времени на строительно-монтажные работы и обслуживание при эксплуатации скважин с близко расположенными устьями;
- обхода зон катастрофических поглощений, обвалов и аварий в стволе скважины;
- вскрытия продуктивных пластов, залегающих под пологим сбросом или между двумя параллельными сбросами;
- проходки стволов на нефтяные пласты, залегающие под соляными куполами, в связи с трудностью бурения через них (соль «плышет», срезает бурильные и обсадные колонны);
- бурения стволов для глушения открытых фонтанов и тушения пожаров;
- перебуривания части ствола скважины;
- вскрытия продуктивного пласта под определенным углом для увеличения поверхности дренажа и дебита скважины;
- многозабойного вскрытия продуктивного пласта.

Помимо искусственного искривления скважин так же имеет место быть самопроизвольное искривление, связанное с геологическими, техническими и технологическими факторами.

Геологические факторы:

- перемежаемость по твердости - чередование мягких и твердых горных пород;
- жеоды - инородные тела в составе горной породы, отличающиеся по твердости;

Технические факторы:

- несоосность вышки относительно осей стола ротора и шахтового направления;
- негоризонтальность стола ротора;
- использование искривленных труб (ведущих и бурильных) и труб, у которых резьбы нарезаны под углом;
- эксцентричное забуривание нижележащего участка скважины.

Технологические факторы:

- влияние осевой нагрузки;
- влияние частоты вращения бурильной колонны.

Методы предупреждения самопроизвольного искривления скважин

- для предупреждения естественного искривления скважин необходимо исключить или уменьшить действие управляемых технических факторов и нейтрализовать действие неуправляемых геологических условий;
- технические причины искривлений должны быть устранены до начала бурения скважины;
- действие технологических причин искривления могут быть сведены к минимуму центрированием низа бурильной колонны, увеличением его жесткости, регулированием осевой нагрузки;
- цель центрирования нижней части бурильной колонны - препятствовать отклонению оси долота от оси скважины;
- увеличение жесткости и массы нижней части бурильной колонны повышает устойчивость к изгибу, уменьшает длину сжатой части, позволяет использовать повышенные нагрузки на долото;
- для компенсации геологических причин искривления (наклонно-залегающие анизотропные породы) можно использовать методы наклонно направленного разбуривания ствола в направлении, противоположном естественному искривлению;
- метод буровых трасс – перенос устья скважины по азимуту и величине смещения при самопроизвольном искривлении скважин.

В связи с неуправляемым действием геологических условий, при бурении следует контролировать данные инклинометрии и вовремя принимать решения об искусственном искривлении ствола для предотвращения пересечения скважин и минимизации отхода траектории от проектной.

Поскольку глубина промерзания грунта вокруг ствола замораживающей скважины равна примерно 70см, то, с учетом допуска и погрешности, такие скважины должны буриться на расстоянии примерно 119см друг от друга по всей длине ствола, что требует дополнительного контроля за данным инклинометрии и увеличивает вероятность их пересечения.

В настоящей работе описан программный продукт, позволяющий анализировать данные инклинометрии и оценивать расстояния между стволами по всей их длине, тем самым упрощая принятие решения об искусственном искривлении стволов и предупреждении их пересечения.

1.2 Содержательная постановка проблемы

Целью выпускной квалификационной работы является разработка программного обеспечения для предупреждения пересечения стволов скважин. Входной информацией для программы являются данные измерительного оборудования, выходной - графическое представление стволов и графическое представление расстояний между ними.

1.3 Формальная постановка задачи

Формальной постановке задачи соответствует контекстная диаграмма методологии IDEF0, описывающая входные и выходные данные, управляющие воздействия и механизмы, влияющие на систему в целом, приведённая на Рис.1

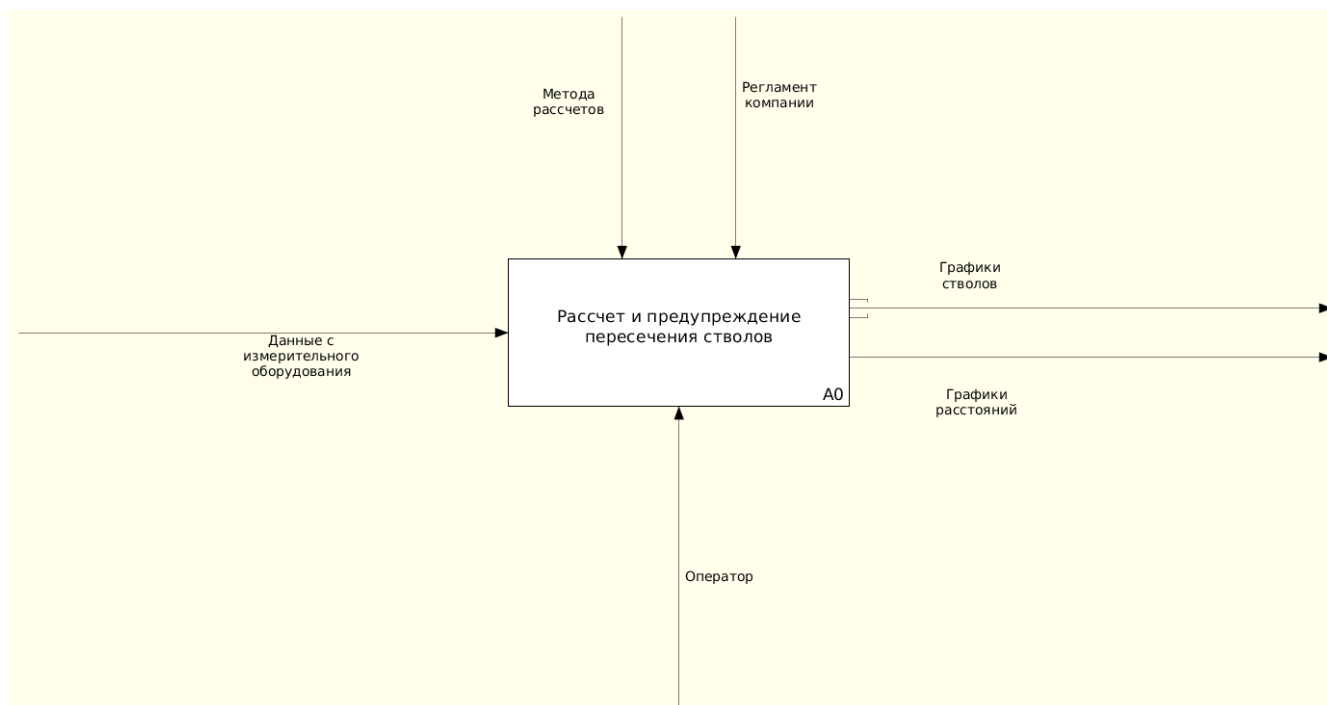


Рис. 1: Общая контекстная диаграмма

1.4 Структура решения задачи, декомпозиция задачи на подзадачи

Общую задачу можно декомпозировать и выделить следующие подзадачи:

- импорт данных измерительного оборудования;
- усреднение траекторий;
- визуализация стволов;
- визуализация расстояний.

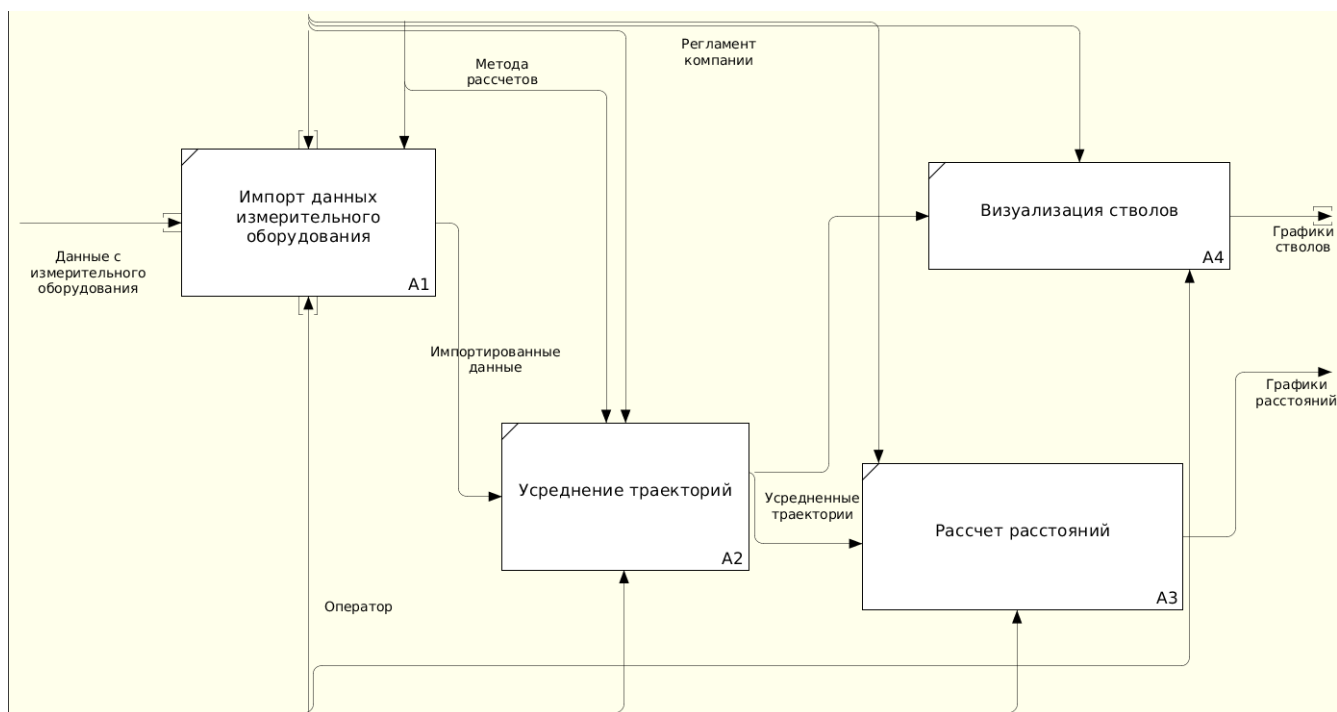


Рис. 2: Структура решения общей задачи

2. Математическое и информационное обеспечение

??? - тут как-то нужно затолкать алгоритм расчета траектории, то есть перевод из данных инклинометрии в декартову систему координат, тот самый метод среднего угла. Еще можно описать алгоритм усреднения замеров и расчета расстояний между стволами. Алгоритмы я могу описать, но как и куда их тут затолкать не представляю

2.1 Классификация подзадач (отнесение подзадач к классу задач)

???

2.2 Математические модели подзадач (где применимо)

???

2.3 Методы решения подзадач (где применимо)

???

2.4 Информационные модели для подзадач (где применимо)

???

2.5 Алгоритмы и структуры данных для подзадач

3. Программное обеспечение

3.1 Аналитический обзор существующих программных технологий, применимых при решении поставленных задач

Существует два основных подхода к разработке программного обеспечения: процедурный и объектно-ориентированный. Каждый из них имеет свои преимущества и недостатки.

При использовании процедурного подхода, обычно появляется огромное количество глобальных переменных, и структурирование происходит на уровне модулей (файлов).

Объектно-ориентированное или объектное программирование — парадигма программирования, в которой основными концепциями являются понятия объектов и классов.

Класс — это тип, описывающий устройство объектов. Понятие «класс» подразумевает некоторое поведение и способ представления. Понятие «объект» подразумевает нечто, что обладает определённым поведением и способом представления. Говорят, что объект — это экземпляр класса. Класс можно сравнить с чертежом, согласно которому создаются объекты. Обычно классы разрабатывают таким образом, чтобы их объекты соответствовали объектам предметной области.

Класс является описываемой на языке терминологии (пространства имён) исходного кода моделью ещё не существующей сущности, т. е. объекта.

Объект — сущность в адресном пространстве вычислительной системы, появляющаяся при создании экземпляра класса (например, после запуска результатов компиляции (и связывания) исходного кода на выполнение).

Перечислим некоторые достоинства ООП:

- классы позволяют проводить конструирование из полезных компонент, обладающих простыми инструментами, что дает возможность абстрагироваться от деталей реализации.
- данные и операции вместе образуют определенную сущность и они не «размазываются» по всей программе, как это нередко бывает в случае процедурного программирования.
- локализация кода и данных улучшает наглядность и удобство сопровождения программного обеспечения.
- инкапсуляция информации защищает наиболее критичные данные от несанкционированного доступа.

Model-view-controller (MVC, «модель-представление-контроллер», «модель-вид-контроллер») — схема использования нескольких шаблонов проектирования, с помощью которых модель приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные. Данная схема проектирования часто используется для построения архитектурного каркаса, когда переходят от теории к реализации в конкретной предметной области.

Предметно-ориентированное проектирование (реже проблемно-ориентированное, англ. Domain-driven design, DDD) — это набор принципов и схем, помогающих разработчикам создавать изящные системы объектов. При правильном применении оно приводит к созданию программных абстракций, которые называются моделями предметных областей. В эти модели входит сложная бизнес-логика, устраняющая промежуток между реальными условиями области применения продукта и кодом.

3.2 Архитектура разрабатываемого программного продукта

Для создания программного продукта был применен объектно-ориентированный подход к программированию, т.к. решение данной задачи требует жесткого контроля над пользователем при работе с системы. Объектно-ориентированный подход применялся там где, происходит сбор информации т.к. необходимо четко определить формат данных.

Qt использует концепцию MVC для отделения данных(Model) от их представления(View) в таких компонентах, как QTreeView, QListView и QTableView.

3.3 Язык программирования и инструментальные средства разработки

3.3.1 Язык C++

На данный момент, C++ остаётся одним из самых популярных и производительных языков программирования и применяется практически во всех прикладных областях программирования, от низкоуровневого программирования для микроконтроллеров, до высокопроизводительных серверных приложений и компьютерных игр.

3.3.2 SQLite

SQLite — это встраиваемая кроссплатформенная СУБД, которая поддерживает достаточно полный набор команд SQL и доступна в исходных кодах (на языке C). На данный момент является самой популярной встраиваемой СУБД. Применяется как на персональный компьютерах, так и в мобильных ОС и ”умных” телевизорах.

3.3.3 Qt

Qt — кроссплатформенный инструментарий разработки ПО на языке программирования C++, доступен в исходных текстах. Позволяет создавать кроссплатформенные приложения с богатыми возможностями графического интерфейса, работой с сетью, мультимедиа, БД и 3D-графикой. В окружении каждой поддерживаемой ОС будет выглядеть максимально похоже на ”родные” приложения системы.

3.3.4 MathGL

MathGL — кроссплатформенная библиотека для визуализации данных. Имеет интеграцию с Qt.

3.3.5 Обоснованность выбора технологий

На данный момент указанные технологии являются единственным способом, как выполнить требования о кроссплатформенности, так и получить лёгкий в поддержке продукт, базирующийся на надёжных и поддерживаемых библиотеках.

3.4 Технологии разработки ПО (моделирование разработки ПО, управление разработкой ПО, конфигурирование ПО, технологии тестирования ПО)

??? на ум приходит только система контроля версий git, больше ничего из вспомогательного софта я не использовал

3.5 Описание структуры программного продукта

??? - в методичке требуется записать сюда диаграмму классов, но сгенерить ее автоматом как-то не получится, а рисовать лень, может воды какой налить?

3.6 Описание интерфейса пользователя

??? - тут было бы здорово сослаться на приложение "руководство пользователя" ибо там весь gui описан

4. Оценка качества решения

4.1 Тестирование ПО

Тестирование является важной и обязательной частью процесса разработки. Процесс тестирования можно разделить на 3 этапа:

- проверка в нормальных условиях;
- проверка в экстремальных условиях;
- проверка в исключительных ситуациях.

Тестирование в нормальных условиях

При проверке в нормальных условиях программа функционировала соответствующим образом: введённые данные были без потерь сохранены в базе данных в нужном формате и в результате запросов были выданы верные сведения.

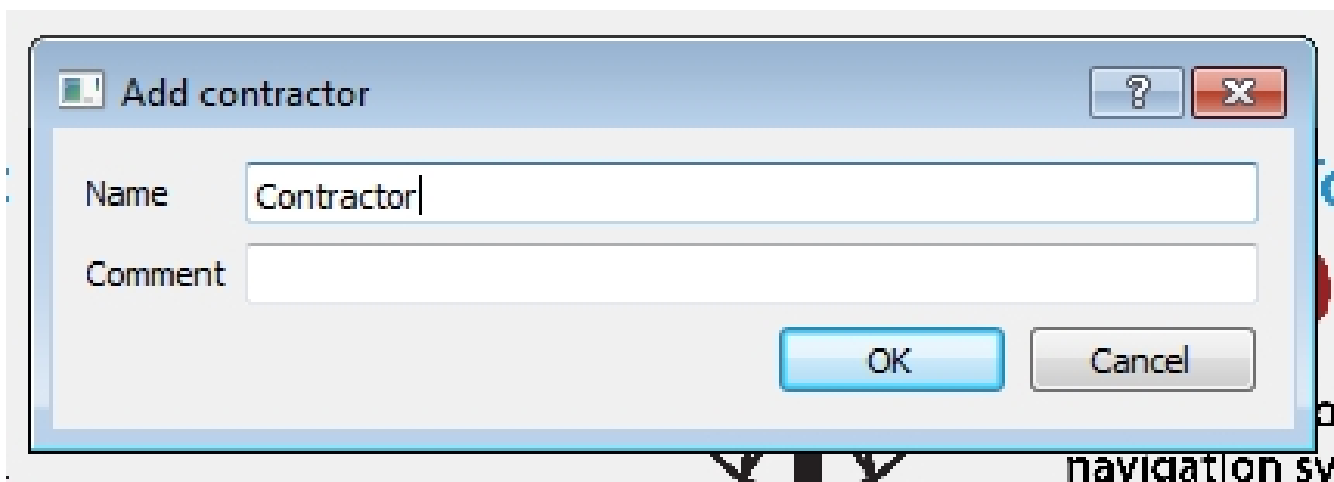


Рис. 3: Ввод корректных параметров подрядчика

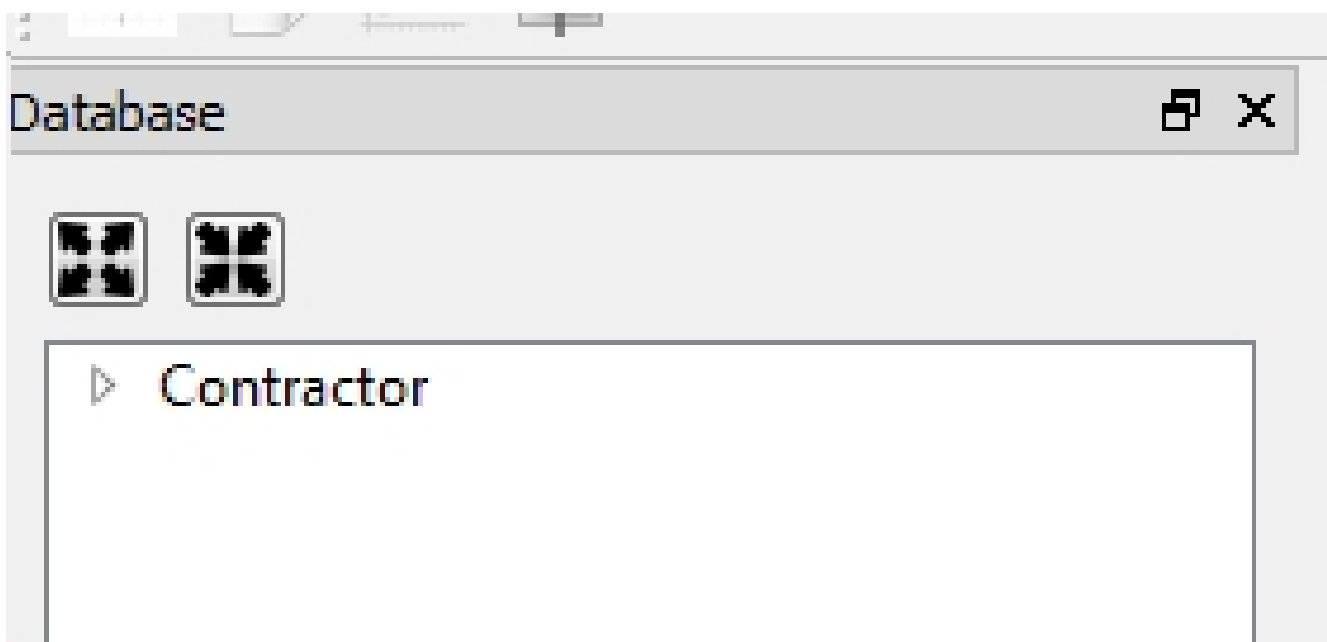


Рис. 4: Созданный подрядчик

Тестирование в экстремальных условиях

Проводилась проверка на ввод нулевых и отсутствующих параметров. Программа не позволяет ввести неверные значения, т.н. "защита от дурака" (Рис. 26).

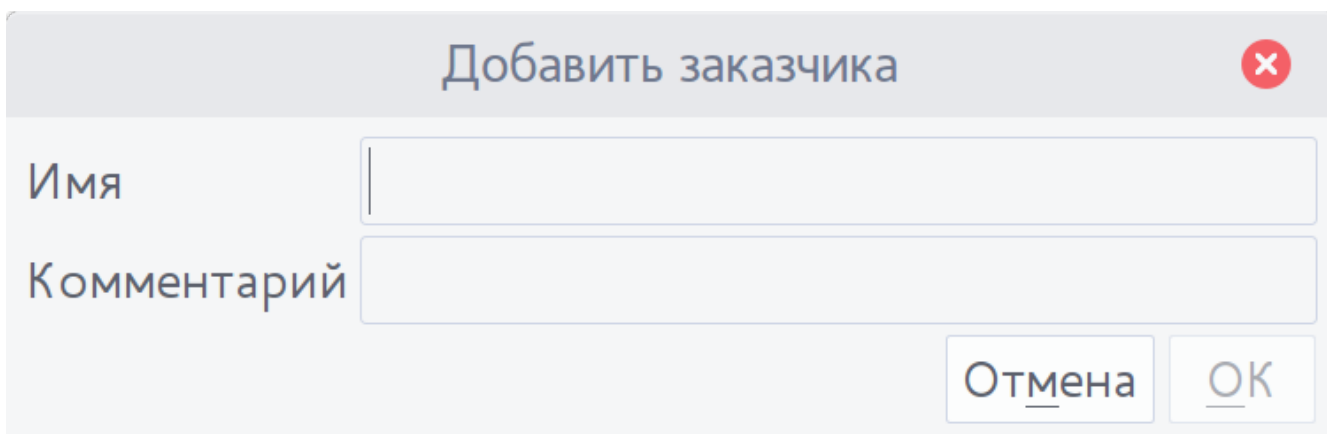


Рис. 5: Недоступная кнопка ОК при попытка создать заказчика без названия

Тестирование в исключительных ситуациях

Тестирование устойчивости программы при вводе неверных данных проводилось с самого начала разработки. Построение интерфейса программы предусматривает предотвращение возможности совершения пользователем действий, приводящих к исключительным ситуациям.

Практически невозможна ситуация, когда в результате сбоя разработанное ПО выйдет из-под контроля и нарушит целостность исходных данных, системы или других прикладных программ.

Анализ тестирования

Тестирование, проведённое в различных условиях, подтверждает работоспособность программы. Возможно, в процессе эксплуатации программы потребуются некоторые её доработки.

4.2 Оценка качества программного продукта

Метрическая оценка качества программного продукта.

В данной части дипломной работы проводится оценка качества программного продукта согласно ГОСТ 28195-89.

Определение подкласса программных средств

Данное программное средство относится к подклассу 509 – Прочие ПС.

Показатели надёжности программного средства

Таблица 2: Оценочные элементы фактора “Надёжность ПС”

Код элемента	Наименование	Метод оценки	Оценка
Средства восстановления при ошибках на входе			
H0101	Наличие требований к программе по устойчивости функционирования при наличии ошибок во входных данных	Экспертный	1
H0102	Возможность обработки ошибочных ситуаций	То же	1
H0103	Полнота обработки ошибочных ситуаций	»	1
H0104	Наличие тестов для проверки допустимых значений входных данных	» с	0
H0105	Наличие системы контроля полноты входных данных	»	0
H0106	Наличие средств контроля корректности входных данных	»	1
H0107	Наличие средств контроля непротиворечивости входных данных	»	0
H0108	Наличие проверки параметров и адресов по диапазону их значений	»	1
H0109	Наличие обработки граничных результатов	»	1
H0110	Наличие обработки неопределенностей	»	0,6
			0,8
Средства восстановления при сбоях оборудования			
H0201	Наличие требований к программе по восстановлению процесса выполнения в случае сбоя операционной системы, процессора, внешних устройств	»	0
H0202	Наличие требований к программе по восстановлению результатов при отказах процессора, ОС	»	1
H0203	Наличие средств восстановления процесса в случае сбоев оборудования	»	0
H0204	Наличие возможности разделения по времени выполнения отдельных функций программ	»	1
H0205	Наличие возможности повторного старта с точки останова	»	1
			0,6
Реализация управления средствами восстановления			
H0301	Наличие централизованного управления процессами, конкурирующими из-за ресурсов	»	1

Таблица 2: Оценочные элементы фактора “Надёжность ПС”

Код элемента	Наименование	Метод оценки	Оценка
H0302	Наличие возможности автоматически обходить ошибочные ситуации в процессе вычисления	»	0
H0303	Наличие средств, обеспечивающих завершение процесса решения в случае помех	»	1
H0304	Наличие средств, обеспечивающих выполнение программы в сокращенном объеме в случае ошибок или помех	»	0
H0305	Показатель устойчивости к искажаемым воздействиям	Расчетный	0
			0,4
Функционирование в заданных режимах			
H0401	Вероятность безотказной работы	То же	1
Обеспечение обработки заданного объема информации			
H0501	Оценка по среднему времени восстановления	»	1
H0502	Оценка по продолжительности преобразования входного набора данных в выходной	»	1
			1

Показатели сопровождения

Таблица 3: Оценочные элементы фактора “Сопровождаемость ПС”

Код элемента	Наименование	Метод оценки	Оценка
Простота архитектуры проекта			
C0101	Наличие модульной схемы программы	Экспертный	1
C0102	Оценка программы по числу уникальных модулей	»	1 1
Сложность архитектуры проекта			
C0201	Наличие ограничений на размеры модуля	»	0
Межмодульные связи			
C030	Наличие требований к независимости модулей программы от типов и форматов выходных данных	»	0
C0301	Наличие проверки корректности передаваемых данных	»	1
C0302	Оценка простоты программы по числу точек входа и выхода	Расчетный	0,01
C0303	Осуществляется ли передача результатов работы модуля через вызывающий его модуль	Экспертный	1
C0304	Осуществляется ли контроль за правильностью данных, поступающих в вызывающий модуль от вызываемого	»	1
			0,6
Экспертиза принятой системы идентификации			
C0601	Использование при построении программ метода структурного программирования	»	1

Таблица 3: Оценочные элементы фактора “Сопровождаемость ПС”

Код элемента	Наименование	Метод оценки	Оценка
C0602	Соблюдение принципа разработки программы сверху вниз	»	1
C0603	Оценка программы по числу циклов с одним входом и одним выходом	»	1
C0604	Оценка программы по числу циклов	»	1
			1
Комментарии логики программ проекта			
C0801	Наличие комментариев ко всем машинозависимым частям программы	»	0
C0802	Наличие комментариев к машинозависимым операторам программы	»	0
C0803	Наличие комментариев в точках входа и выхода программы	»	1
			0,3
Оформление текста программ			
C0901	Соответствие комментариев принятым соглашениям	»	0
C0902	Наличие комментариев-заголовков программы с указанием её структурных и функциональных характеристик	»	0
C0903	Оценка ясности и точности описания последовательности функционирования всех элементов программы	»	0
			0
Простота кодирования			
C1001	Используется ли язык высокого уровня	»	1
C1002	Оценка простоты программы по числу переходов по условию	Расчетный	0,3
			0,6

Показатели удобства применения

Таблица 4: Оценочные элементы фактора “Удобство применения ПС”

Код элемента	Наименование	Метод оценки	Оценка
Освоение работы ПС			
Y0101	Возможность освоения программных средств по документации	Экспертный	1
Y0102	Возможность освоения ПС на контрольном примере при помощи ЭВМ	То же	1
Y0103	Возможность поэтапного освоения ПС	»	1
			1
Документация для освоения			

Таблица 4: Оценочные элементы фактора “Удобство применения ПС”

Код элемента	Наименование	Метод оценки	Оценка
Y0201	Полнота и понятность документации для освоения	»	1
Y0202	Точность документации для освоения	»	1
Y0203	Техническое исполнение документации	»	0,4
			0,8
Полнота пользовательской документации			
Y0301	Наличие краткой аннотации	»	1
Y0302	Наличие описания решаемых задач	»	1
Y0303	Наличие описания структуры функции ПС	»	1
Y0304	Наличие описания основных функций ПС	»	1
Y0306	Наличие описания частных функций	»	1
Y0307	Наличие описания алгоритмов	»	0
Y0308	Наличие описания межмодульных интерфейсов	»	0
Y0309	Наличие описания пользовательских интерфейсов	»	1
Y0310	Наличие описания входных и выходных данных	»	1
Y0311	Наличие описания диагностических сообщений	»	0
Y0312	Наличие описания основных характеристик ПС	»	1
Y0314	Наличие описания программной среды функционирования ПС	»	1
Y0315	Достаточность документации для ввода ПС в эксплуатацию	»	1
Y0316	Наличие информации технологии переноса для мобильных программ	»	0
			0,7
Точность пользовательской документации			
Y0401	Соответствие оглавления содержанию документации	»	1
Y0402	Оценка оформления документации	»	1
Y0403	Грамматическая правильность изложения документации	»	1
Y0404	Отсутствие противоречий	»	1
Y0405	Отсутствие неправильных ссылок	»	1
Y0406	Ясность формулировок и описаний	»	1
Y0407	Отсутствие неоднозначных формулировок и описаний	»	1
Y0408	Правильность использования терминов	»	1
Y0409	Краткость, отсутствие лишней детализации	»	1
Y0410	Единство формулировок	»	1
Y0411	Единство обозначений	»	1
Y0412	Отсутствие ненужных повторений	»	1
Y0413	Наличие нужных объяснений	»	1
			1
Понятность пользовательской документации			
Y0501	Оценка стиля изложения	»	1
Y0502	Дидактическая разделенность	»	1

Таблица 4: Оценочные элементы фактора “Удобство применения ПС”

Код элемента	Наименование	Метод оценки	Оценка
Y0503	Формальная разделенность	»	1
Y0504	Ясность логической структуры	»	1
Y0505	Соблюдение стандартов и правил изложения в документации	»	1
Y0506	Оценка по числу ссылок вперед в тексте документов	»	0
			0,8
Техническое исполнение пользовательской документации			
Y0601	Наличие оглавления	»	1
Y0602	Наличие предметного указателя	»	0
Y0603	Наличие перекрестных ссылок	»	0
Y0604	Наличие всех требуемых разделов	»	1
Y0605	Соблюдение непрерывности нумерации страниц документов	»	1
Y0606	Отсутствие незаконченных разделов абзацев, предложений	»	1
Y0607	Наличие всех рисунков, чертежей, формул, таблиц	»	1
Y0608	Наличие всех строк и примечаний	»	1
Y0609	Логический порядок частей внутри главы	»	1
			0,8
Прослеживание вариантов пользовательской документации			
Y0701	Наличие полного перечня документации	»	1
Эксплуатация			
Y0801	Уровень языка общения пользователя с программой	»	1
Y0802	Легкость и быстрота загрузки и запуска программы	»	1
Y0803	Легкость и быстрота завершения работы программы	»	1
Y0804	Возможность распечатки содержимого программы	»	0,7
Y0805	Возможность приостановки и повторного запуска работы без потерь информации	»	1
			0,9
Управление меню			
Y0901	Соответствие меню требованиям пользователя	»	1
Y0902	Возможность прямого перехода вверх и вниз по многоуровневому меню (пропуск уровней)	»	1
			1
Функция Help			
Y1001	Возможность управления подробностью получаемых выходных данных	»	1
Y1002	Достаточность полученной информации для продолжения работы	»	1
			1
Управление данными			
Y1101	Обеспечение удобства ввода данных	»	1

Таблица 4: Оценочные элементы фактора “Удобство применения ПС”

Код элемента	Наименование	Метод оценки	Оценка
У1102	Легкость восприятия	»	1
			1
Рабочие процедуры			
У1201	Обеспечение программой выполнения предусмотренных рабочих процедур	»	1
У1202	Достаточность информации, выдаваемой программой для составления дополнительных процедур	»	1
			1

Показатели эффективности

Таблица 5: Оценочные элементы фактора “Эффективность ПС”

Код элемента	Наименование	Метод оценки	Оценка
Уровень автоматизации			
Э0101	Проблемно-ориентированные функции	Экспертный	1
Э0102	Машинно-ориентированные функции	То же	1
Э0103	Функции ведения и управления	»	1
Э0104	Функции ввода/вывода	»	1
Э0105	Функции защиты и проверки данных	»	0
Э0106	Функции защиты от несанкционированного доступа	»	1
Э0107	Функции контроля доступа	»	1
Э0108	Функции защиты от внесения изменений	»	1
Э0109	Наличие соответствующих границ функциональных областей	»	1
Э0110	Число знаков после запятой в результатах вычислений	»	1
			0,9
Временная эффективность			
Э0201	Время выполнения программ	»	1
Э0202	Время реакции и ответов	»	1
Э0203	Время подготовки	»	1
Э0205	Затраты времени на защиту данных	»	0
Э0206	Время компиляции	»	1
			0,8
Ресурсоемкость			
Э0301	Требуемый объем внутренней памяти	»	1
Э0302	Требуемый объем внешней памяти	»	1
Э0303	Требуемые периферийные устройства	»	1
Э0304	Требуемое базовое программное обеспечение	»	1
			1

Показатели универсальности

Таблица 6: Оценочные элементы фактора “Универсальность ПС”

Код элемента	Наименование	Метод оценки	Оценка
Зависимость от используемого комплекса технических средств			
Г0701	Оценка зависимости программ от ёмкости оперативной памяти ЭВМ	»	1
Г0702	Оценка зависимости временных характеристик программы от скорости вычислений ЭВМ	»	1
Г0703	Оценка зависимости функционирования программы от числа внешних запоминающих устройств и их общей емкости	»	0
Г0704	Оценка зависимости функционирования программы от специальных устройств ввода-вывода	»	1
			0,7
Зависимость от базового программного обеспечения			
Г0801	Применение специальных языков программирования	»	1
Г0802	Оценка зависимости программы от программ операционной системы	»	1
Г0803	Зависимость от других программных средств	»	1
			1
Изоляция немобильности			
Г0901	Оценка локализации переносимой части программы	»	1
			1
Простота кодирования			
Г1001	Оценка использования отрицательных или булевых выражений	»	1
Г1002	Оценка программы по использованию условных переходов	»	1
Г1003	Оценка программы по использованию безусловных переходов	»	0
Г1004	Оформление процедур входа и выхода из циклов	»	1
Г1005	Ограничения на модификацию переменной индексации в цикле	»	1
Г1007	Оценка программы по использованию локальных переменных	»	1
Г1006	Оценка модулей по направлению потока управления	»	0
			0,7
Число комментариев			
Г1101	Оценка программы по числу комментариев	»	1
			1
Качество комментариев			
Г1201	Наличие заголовка в программе	»	1
Г1202	Комментарии к точкам ветвлений	»	1
Г1203	Комментарии к машинозависимым частям программы	»	1

Таблица 6: Оценочные элементы фактора “Универсальность ПС”

Код элемента	Наименование	Метод оценки	Оценка
Г1204	Комментарии к машинозависимым операторам программы	»	0
Г1205	Комментарии к операторам объявления переменных	»	1
Г1206	Оценка семантики операторов	»	1
Г1207	Наличие соглашений по форме представления комментариев	»	0
Г1208	Наличие общих комментариев к программам	»	1
			0,7
Использование описательных средств языка			
Г1301	Использование языков высокого уровня	»	1
Г1302	Семантика имен используемых переменных	»	1
Г1303	Использование отступов, сдвигов и пропусков при формировании текста	»	1
Г1304	Размещение операторов по строкам	»	1
			1
Независимость модулей			
Г1401	Передача информации для управления по параметрам	»	1
Г1402	Наличие передачи результатов работы между модулями	»	1
Г1403	Наличие проверки правильности данных, получаемых модулями от вызываемого модуля	»	1
Г1404	Использование общих областей памяти	»	1
Г1405	Параметрическая передача входных данных	»	1
			1

Показатели корректности

Таблица 7: Оценочные элементы фактора “Корректность ПС”

Код элемента	Наименование	Метод оценки	Оценка
Требования, предъявляемые к полноте документации разработчика			
K0101	Наличие всех необходимых документов для понимания и использования ПС	Экспертный	1
K0102	Наличие описания и схемы иерархии модулей программы	»	1
K0103	Наличие описания основных функций	»	1
K0104	Наличие описания частных функций	»	1
K0105	Наличие описания данных	»	1
K0106	Наличие описания алгоритмов	»	1
K0107	Наличие описания интерфейсов между модулями	»	1
K0108	Наличие описания интерфейсов с пользователями	»	1
K0109	Наличие описания используемых числовых методов	»	0
K0110	Указаны ли все численные методы	»	0

Таблица 7: Оценочные элементы фактора “Корректность ПС”

Код элемента	Наименование	Метод оценки	Оценка
K0111	Наличие описания всех параметров	»	0
K0112	Наличие описания методов настройки системы	»	1
K0113	Наличие описания всех диагностических сообщений	»	1
K0114	Наличие описания способов проверки работоспособности программы	»	1
			0,8
Полнота программной документации			
K0201	Реализация всех исходных модулей	»	1
K0202	Реализация всех основных функций	»	1
K0203	Реализация всех частных функций	»	1
K0204	Реализация всех алгоритмов	»	1
K0205	Реализация всех взаимосвязей в системе	»	1
K0206	Реализация всех интерфейсов между модулями	»	1
K0207	Реализация возможности настройки системы	»	1
K0208	Реализация диагностики всех граничных и аварийных ситуаций	»	1
K0209	Наличие определения всех данных (переменные, индексы, массивы и проч.)	»	1
K0210	Наличие интерфейсов с пользователем	»	1
			1
Непротиворечивость документации разработчика			
K0301	Отсутствие противоречий в описании частных функций	»	1
K0302	Отсутствие противоречий в описании основных функций в разных документах	»	1
K0303	Отсутствие противоречий в описании алгоритмов	»	1
K0304	Отсутствие противоречий в описании взаимосвязей в системе	»	1
K0305	Отсутствие противоречий в описании интерфейсов между модулями	»	1
K0306	Отсутствие противоречий в описании интерфейсов с пользователем	»	1
K0307	Отсутствие противоречий в описании настройки системы	»	1
K0309	Отсутствие противоречий в описании иерархической структуры сообщений	»	1
K0310	Отсутствие противоречий в описании диагностических сообщений	»	1
K0311	Отсутствие противоречий в описании данных	»	1
			1
Непротиворечивость программы			
K0401	Отсутствие противоречий в выполнении основных функций	»	1

Таблица 7: Оценочные элементы фактора “Корректность ПС”

Код элемента	Наименование	Метод оценки	Оценка
K0402	Отсутствие противоречий в выполнении частных функций	»	1
K0403	Отсутствие противоречий в выполнении алгоритмов	»	1
K0404	Правильность взаимосвязей	»	1
K0405	Правильность реализации интерфейса между модулями	»	1
K0406	Правильность реализации интерфейса с пользователем	»	1
K0407	Отсутствие противоречий в настройке системы	»	1
K0408	Отсутствие противоречий в диагностике системы	»	1
K0409	Отсутствие противоречий в общих переменных	»	1
			1
Единообразие интерфейсов между модулями и пользователями			
K0501	Единообразие способов вызова модулей	»	1
K0502	Единообразие процедур возврата управления из модулей	»	1
K0503	Единообразие способов сохранения информации для возврата	»	0
K0504	Единообразие способов восстановления информации для возврата	»	0
K0505	Единообразие организации списков передаваемых параметров	»	0
			0,4
Единообразие кодирования и определения переменных			
K0601	Единообразие наименования каждой переменной и константы	»	1
K0602	Все ли одинаковые константы встречаются во всех программах под одинаковыми именами	»	0
K0603	Единообразие определения внешних данных во всех программах	»	1
K0604	Используются ли разные идентификаторы для разных переменных	»	1
K0605	Все ли общие переменные объявлены как общие переменные	»	1
K0606	Наличие определений одинаковых атрибутов	»	1
			0,8
Соответствие документации стандартам			
K0701	Комплектность документации в соответствии со стандартами	»	1
K0702	Правильное оформление частей документов	»	1
K0703	Правильное оформление титульных и заглавных листов документов	»	1
K0704	Наличие в документах всех разделов в соответствии со стандартами	»	1

Таблица 7: Оценочные элементы фактора “Корректность ПС”

Код элемента	Наименование	Метод оценки	Оценка
K0705	Полнота содержания разделов в соответствии со стандартами	»	0
K0706	Деление документов на структурные элементы: разделы, подразделы, пункты, подпункты	»	1
			0,8
Соответствие ПС стандартам программирования			
K0801	Соответствие организации и вычислительного процесса эксплуатационной документации	»	1
K0802	Правильность заданий на выполнение программы, правильность написания управляющих и операторов (отсутствие ошибок)	»	1
K0803	Отсутствие ошибок в описании действий пользователя	»	1
K0804	Отсутствие ошибок в описании запуска	»	1
K0805	Отсутствие ошибок в описании генерации	»	1
K0806	Отсутствие ошибок в описании настройки	»	1
			1
Полнота тестирования проекта			
K1001	Наличие требований к тестированию программ	»	0
K1002	Достаточность требований к тестированию программ	»	0
K1003	Отношение числа модулей, отработавших в процессе тестирования и отладки (Q _{тм}) к общему числу модулей (Q _{тм})	Расчетный	1
K1004	Отношение числа логических блоков, отработавших в процессе тестирования и отладки (Q _{тб}), к общему числу логических блоков в программе (Q _{тб})	То же	1
			0,7

Абсолютные показатели критериев *i*-ого фактора качества определяется по формуле:

$$P_i = \sum_{k=0}^n (P_{jk}^M * V_{jk}^M)$$

, где

P_{jk}^M - итоговая оценка *k*-той метрики *j*-того критерия;

V_{jk}^M - весовой коэффициент *j*-того показателя;

n - число метрик, относящихся к *j*-тому критерию.

Таким образом, абсолютные показатели составляют:

Таблица 8: Результаты оценки качества программного продукта

Фактор качества	Оценка
Надёжность	0,7
Сопровождаемость	0,5
Удобство применения	0,9

Таблица 8: Результаты оценки качества программного продукта

Фактор качества	Оценка
Эффективность	0,9
Универсальность	0,7
Корректность	0,7

Все показатели принимают значения в пределах требуемой нормы.

Выводы

В результате проделанной работы была произведена оценка качества программного продукта “Программа для предупреждения пересечения стволов скважин”.

Показатель оценки надёжности равен 0,7. Эта величина показывает, что программа оснащена определенными базовыми методами защиты от сбоев и злоумышленников.

Высокий показатель универсальности равен 0,7 говорит о том, что данный модуль может быть перенесен в другие приложения.

Значение показателя сопровождаемости равное 0,5 говорит о необходимости дальнейшей работы по улучшению наглядности и устойчивости функционирования.

Полученные оценки 0,9 означают, что программа достаточно эффективна и удобна в применении.

4.3 Вычислительный эксперимент и анализ результатов

??? - тут вот тоже нифига не понятно. Может убрать главу совсем. Никто не заметит=)

ЗАКЛЮЧЕНИЕ

Результаты работы

В представленной работе описан программный продукт для предотвращения пересечения стволов скважин. Он позволяет осуществлять следующие действия: импортировать данные измерительных устройств, визуализировать траектории стволов и оценивать расстояния между ними.

Такой программный продукт может найти применение в строительстве с использованием технологии искусственного замораживания грунтов, а так же в бурении нефтяных и газовых скважин.

Для работы с данной программой требуется знания в области нефтяной промышленности, а также понимание принципов инклинометрии. Требования к наличию знаний компьютерных навыков минимальна.

Выводы

??? - в голове что-то типа ”в процессе выполнения ВКР я познал дзен и понял всю тщетность бытия”

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

ПРИЛОЖЕНИЯ

Программная документация

Техническое задание

Наименование программы

”Приложение для предупреждения пересечения стволов скважин ”Collisions”

Характеристика области применения программы

Программа (“Collisions”) разрабатывается в рамках выпускной квалификационной работы. Программа находится в стадии внедрения.

Основания для разработки

Разработка программного обеспечения ведется в соответствии с заданием на дипломное проектирование, составленным совместно с руководителем дипломной работы и утвержденным кафедрой ВМиК.

Назначение разработки

Программа предназначена для визуализации, усреднения замеров стволов скважин, а так же для предупреждения их пересечения.

Требования к функциональным характеристикам

Данный программный комплекс должен обладать следующими функциями:

- внесение, редактирование и удаление данных о подрядчиках;
- внесение, редактирование и удаление данных о клиентах;
- внесение, редактирование и удаление данных о месторождениях;
- внесение, редактирование и удаление данных о кустах;
- внесение, редактирование и удаление данных о скважинах;
- внесение, редактирование и удаление данных о стволах;
- внесение, редактирование и удаление данных о замерах;
- импорт данных замера из csv файлов, полученных от измерительного оборудования;
- визуализация кустов, скважин, стволов и замеров;
- усреднение замеров ствола;
- расчет и визуализация расстояний между стволами скважин в кусте;

Требования к надёжности

Программный продукт (ПП) должен обеспечивать: устойчивую и корректную работу с базой данных, сохранность информации в случаях возникновения сбоев.

Требования к обеспечению надёжного функционирования программы

Надёжное (устойчивое) функционирование программы должно быть обеспечено выполнением Заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

- а) организацией бесперебойного питания технических средств;

б) регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;

в) регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов

Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 30-ти минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Отказы из-за некорректных действий пользователей системы

Отказы программы вследствие некорректных действий пользователя при взаимодействии с программой недопустимы.

Требования к квалификации и численности персонала

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 2 штатных единиц — системный администратор и конечный пользователь программы — оператор.

В перечень задач, выполняемых системным администратором, должны входить:

- а) задача поддержания работоспособности технических средств;
- б) задачи установки (инсталляции) и поддержания работоспособности системных программных средств — операционной системы;
- в) задача установки (инсталляции) программы.

Требования к составу и параметрам технических средств

Для выполнения программы желательна следующая аппаратная конфигурация:

- ПК с x86 - совместимым процессором 1ГГц и выше;
- оперативная память - не менее 512Мб;
- минимум 200Мб свободного пространство на диске;
- OS Windows 7 или старше или ОС на базе ядра Linux

Требования к организации входных данных

Входными данными являются:

- данные замера ствола, полученные от измерительного оборудования;
- данные должны быть в формате CSV;
- точная структура CSV файла не оговаривается, но он обязан содержать данные о последовательных замерах глубины по стволу, зенитного угла и азимута;
- данные о подрядчиках, клиентах, месторождениях, кустах, скважинах, стволах и замерах;

Требования к формированию выходных данных

Выходными данными являются графическое представление замеров стволов, а так же графическое представление расстояний между стволами.

Требования к реализуемым методам решения

Методы решения, используемые в работе программы, должны быть эффективными и высокопроизводительными, позволять получать верный результат за приемлемое время, а также контролировать случаи возникновения некорректной работы.

Требования к исходным кодам и языкам программирования

Система должна быть написана на языке C++ и иметь удобный графический интерфейс.

Состав и требования к программной документации

В состав программной документации должны входить:

- техническое задание;
- руководство программиста;
- руководство пользователя.

Руководство программиста

Назначение и условия применения программы

Приложение для предупреждения пересечения стволов скважин. Программный продукт должен работать на любых ПК с x86-совместимым процессором с частотой 1ГГц и выше, оперативной памятью не менее 512мб и доступным дисковым пространством минимум 200Мб, работающий под управлением ОС Windows 7 или ОС на базе ядра Linux.

Структура программы

Программа написана с использованием архитектурного подхода MVC и состоит из набора классов. Один сpp файл содержит только один класс, каждый сpp файл имеет соответствующий одноименный h файл. Файлы исходных кодов сгруппированы в следующие поддиректории:

- корень проекта - содержит классы основных окон, виджеты для визуализации данных и некоторые вспомогательные классы;
- delegates - т.н. делегаты, классы, отвечающие за отображение данных в ячейках таблиц и списков;
- dialogs - классы, отвечающие за логику работы диалоговых окон;
- entities - классы, описывающие базовые структуры данных, вроде Подрядчика или Месторождения;
- import_wizard - классы, отвечающие за логику мастера импорта данных;
- log - классы, отвечающие за логику журналов;
- menus - классы, отвечающие за различные контекстные меню;
- mixins - вспомогательные классы, от которых наследуются некоторые классы приложения;
- models - модели данных, большая часть из них описывает таблицы БД;
- views - классы, отвечающие за отображение моделей;

Программа хранит свои настройки с использованием абстракции над стандартной системой хранения настроек для текущей платформы: для Windows это реестр, для Linux-систем - это текстовый файл `./config/SPT/Collisions.conf`.

БД продукта представляет собой файл `db.sqlite`, он может быть прочитан и отредактирован любой, поддерживающей формат `sqlite`, утилитой.

Сообщения программисту

В программе не предусмотрен вывод сообщений специально для программиста, однако в ходе работы программы могут появиться общие сообщения программы.

Руководство пользователя

Назначение и условия применения программы

Приложение для предупреждения пересечения стволов скважин. Пользование программой не требует специальной квалифицированной подготовки.

Условия применения программы

Программный продукт должен работать на любых ПК с x86-совместимым процессором с частотой 1ГГц и выше, оперативной памятью не менее 512мб и доступным дисковым пространством минимум 200Мб, работающий под управлением ОС Windows 7 или ОС на базе ядра Linux.

Требования к квалификации пользователя программы

- знакомство с любой из поддерживаемых ОС
- знакомство с руководством пользователя
- знакомство с руководством пользователя

Установка программы

Копировать директорию с ПП на компьютер, при необходимости создать на рабочем столе(зависит от ОС)

Запуск программы

Для запуска программы необходимо исполнить бинарный файл `Collisions`(или `Collisions.exe` для ОС Windows)

Интерфейс программы

После запуска программы открывается главное окно программы(Рис.6)

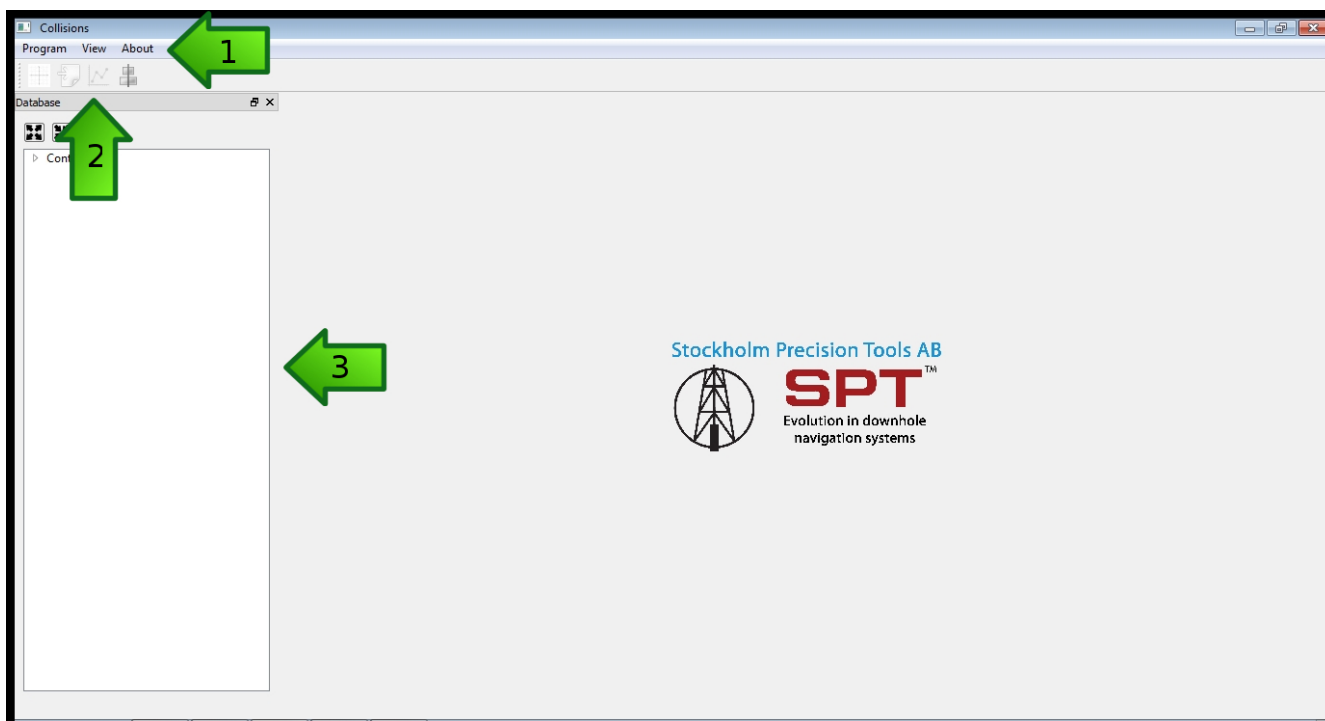


Рис. 6: Главное окно программы: (1) главное меню программы (2) панель инструментов (3) панель дерева базы данных

Для добавления подрядчика в базу необходимо воспользоваться пунктом меню Program->Add contractor (Программа->Добавить подрядчика)(Рис.7)

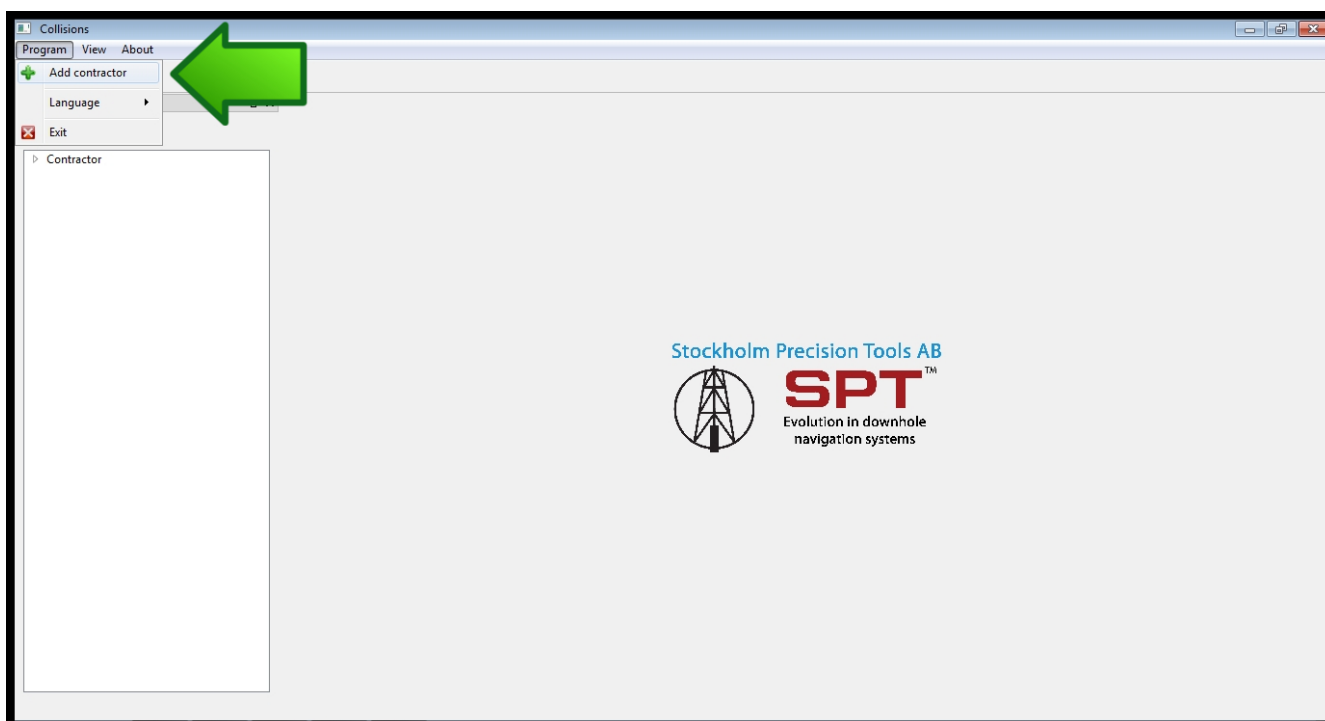


Рис. 7: Меню добавления нового подрядчика

В появившемся диалоге ввести название/имя подрядчика, и, при необходимости, комментарий(

Рис.8)

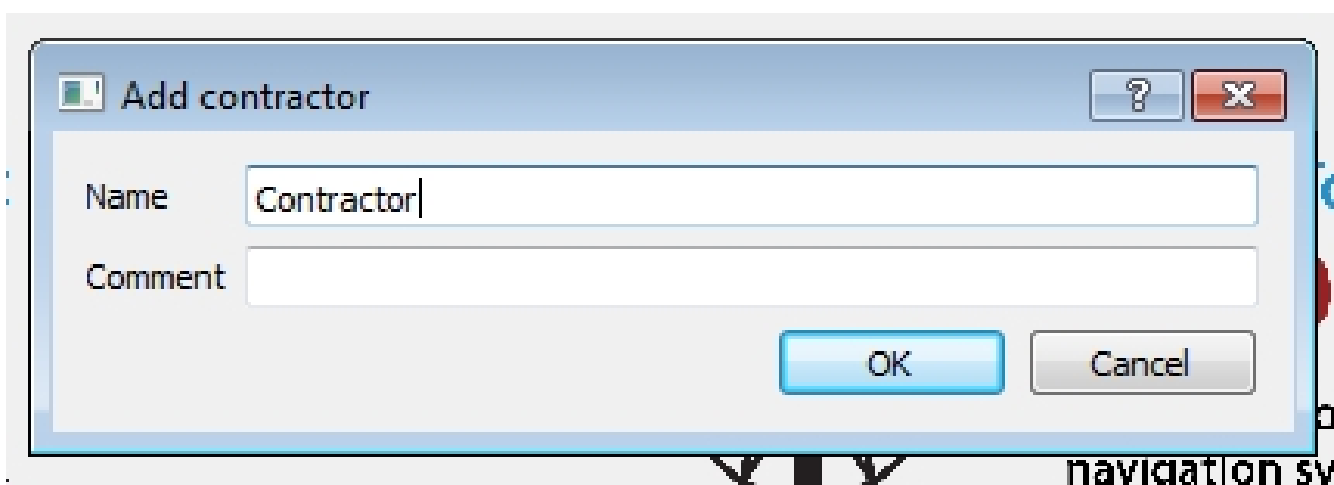


Рис. 8: Диалог добавления нового подрядчика

После нажатия Ok Подрядчик будет добавлен в базу(Рис.9)

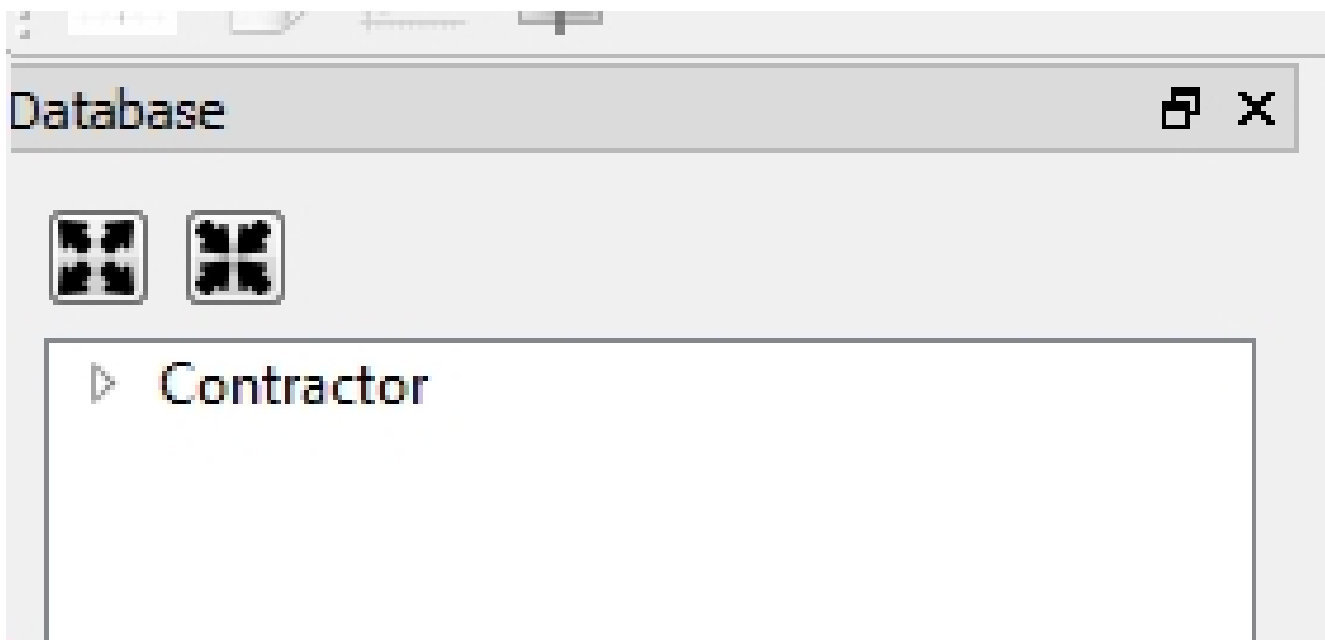


Рис. 9: Созданный подрядчик

Другие элементы добавляются в базу отличным от подрядчика методом. Для добавления элемента необходимо кликнуть по его родителю (для заказчика это подрядчик, для месторождения - заказчик и т.д) в дереве базы и в открывшейся справа таблице нажать Insert. В таблице появится пустая строка для добавления нового элемента, первое поле строки будет активно для редактирования (Рис.10 и Рис.11)

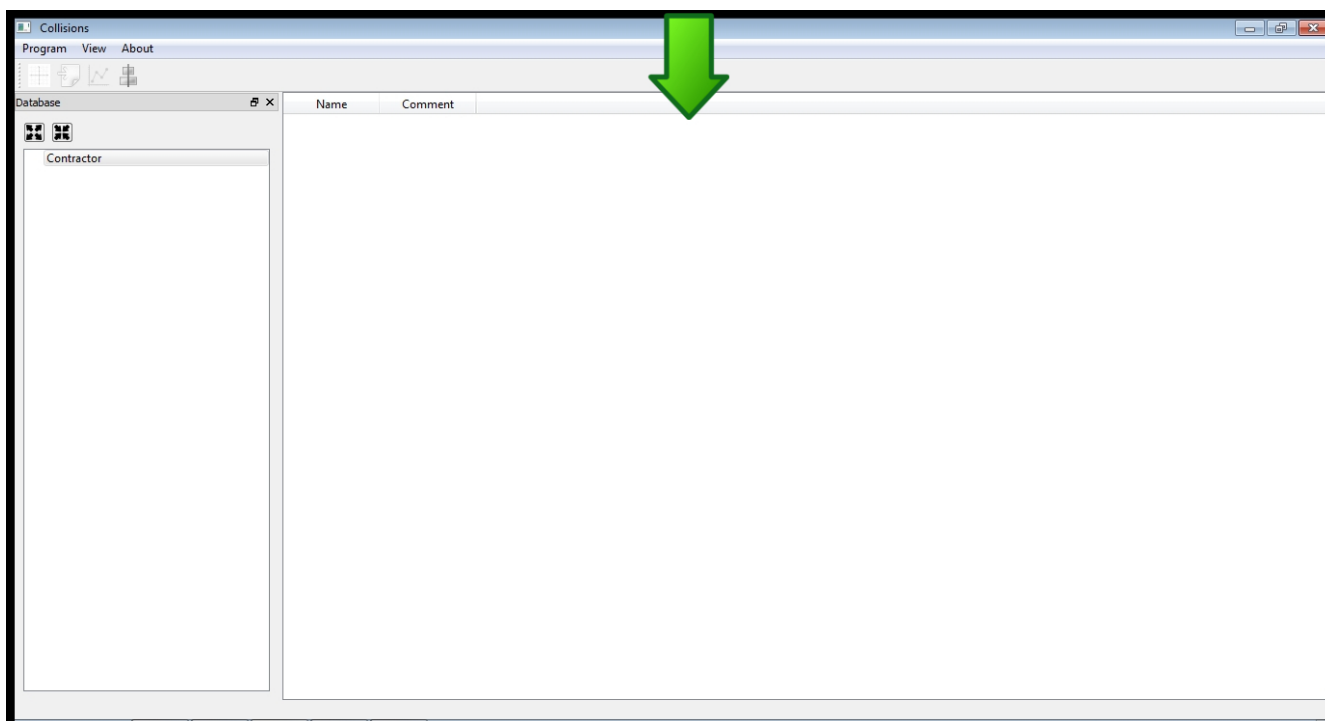


Рис. 10: Таблица для редактирования и просмотра содержимого элемента базы

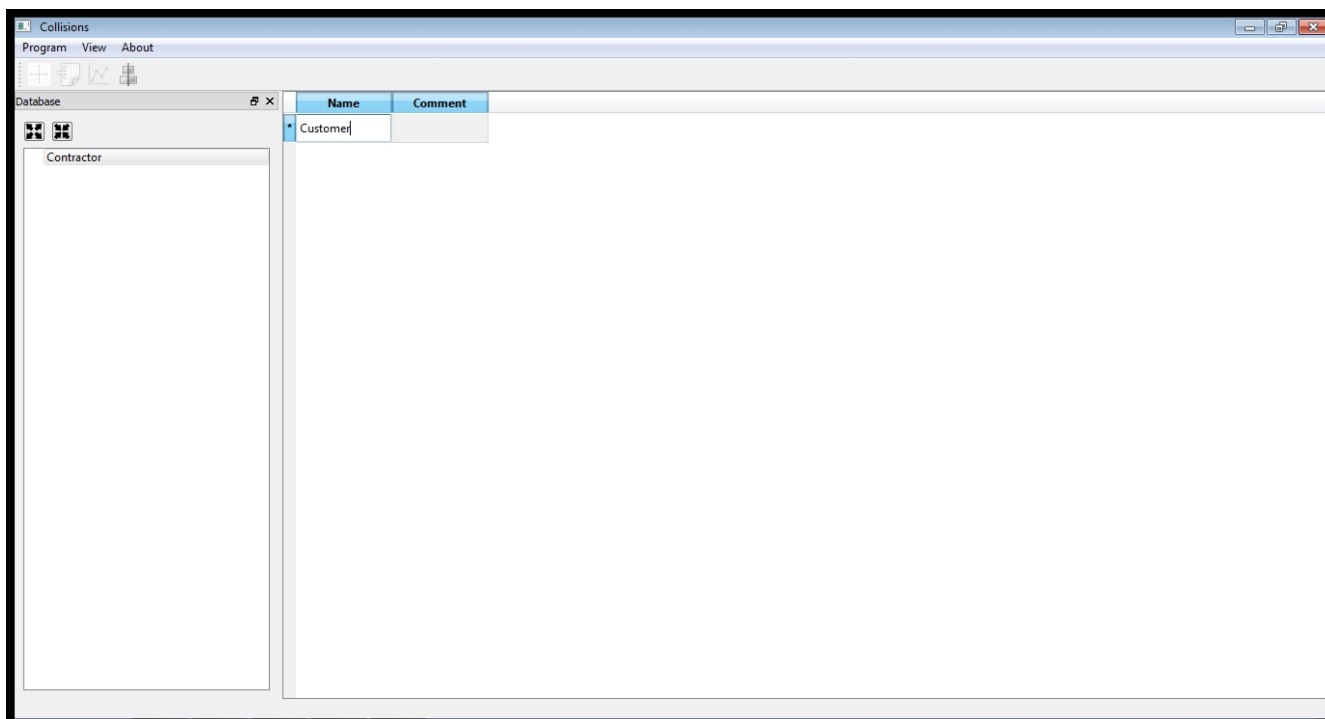


Рис. 11: Добавление нового элемента в таблицу

Для перемещения между полями новой строки необходимо использовать Tab, для сохранения элемента в базе - Enter. Все остальные элементы (Месторождения, кусты, скважины, стволы, замеры и точки замера) добавляются аналогично. См. изображения (Рис.12, Рис.13 и Рис.11)

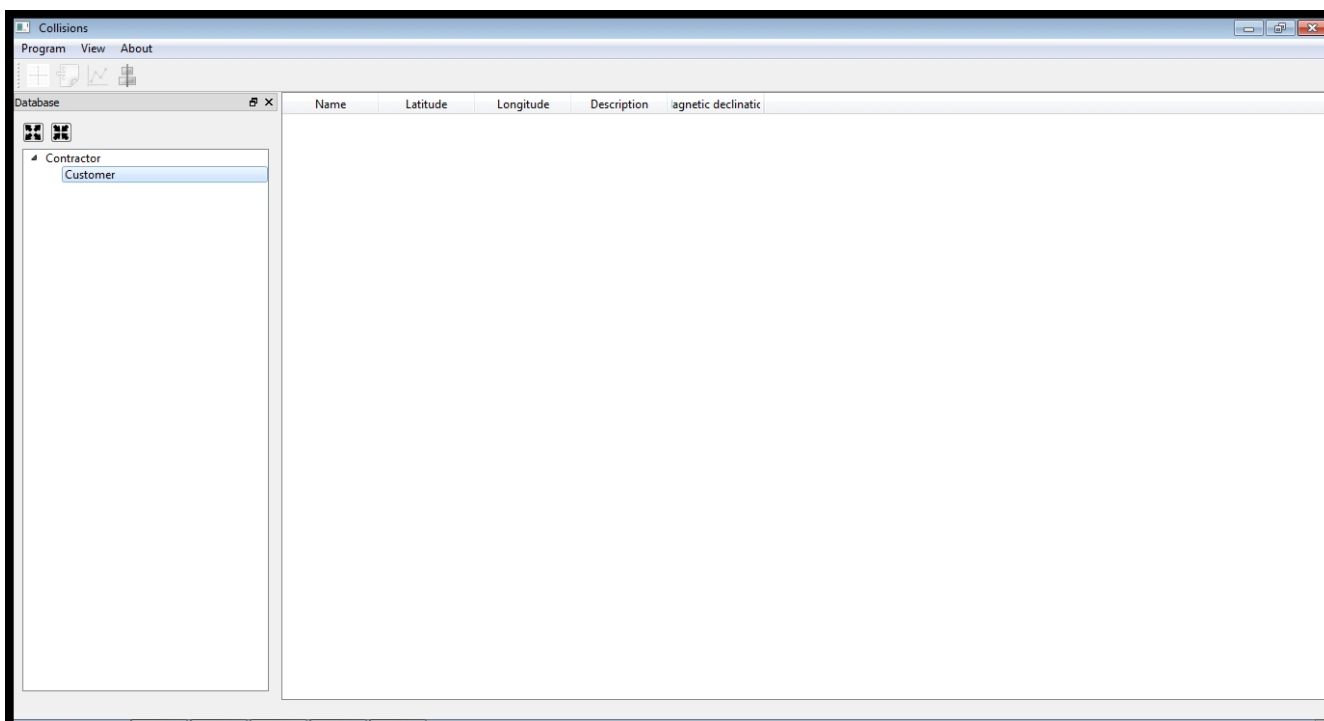


Рис. 12: Таблица месторождений

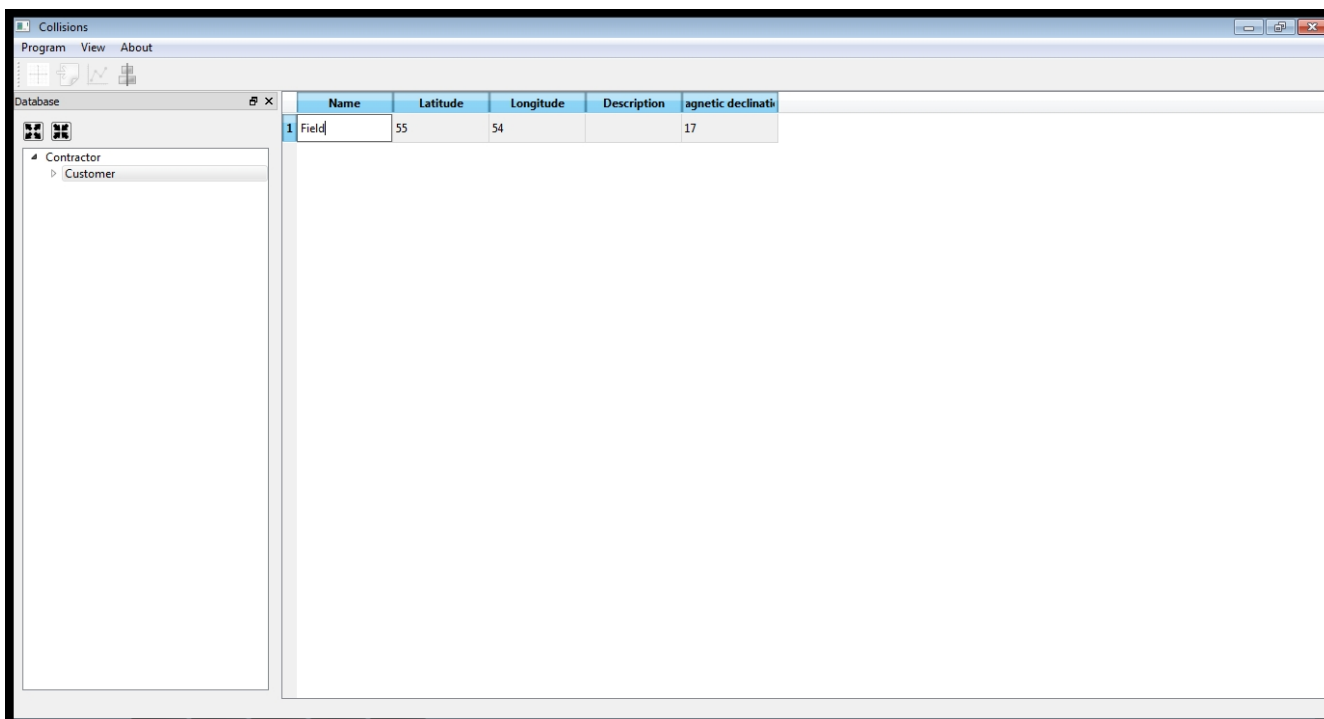


Рис. 13: Добавление месторождения

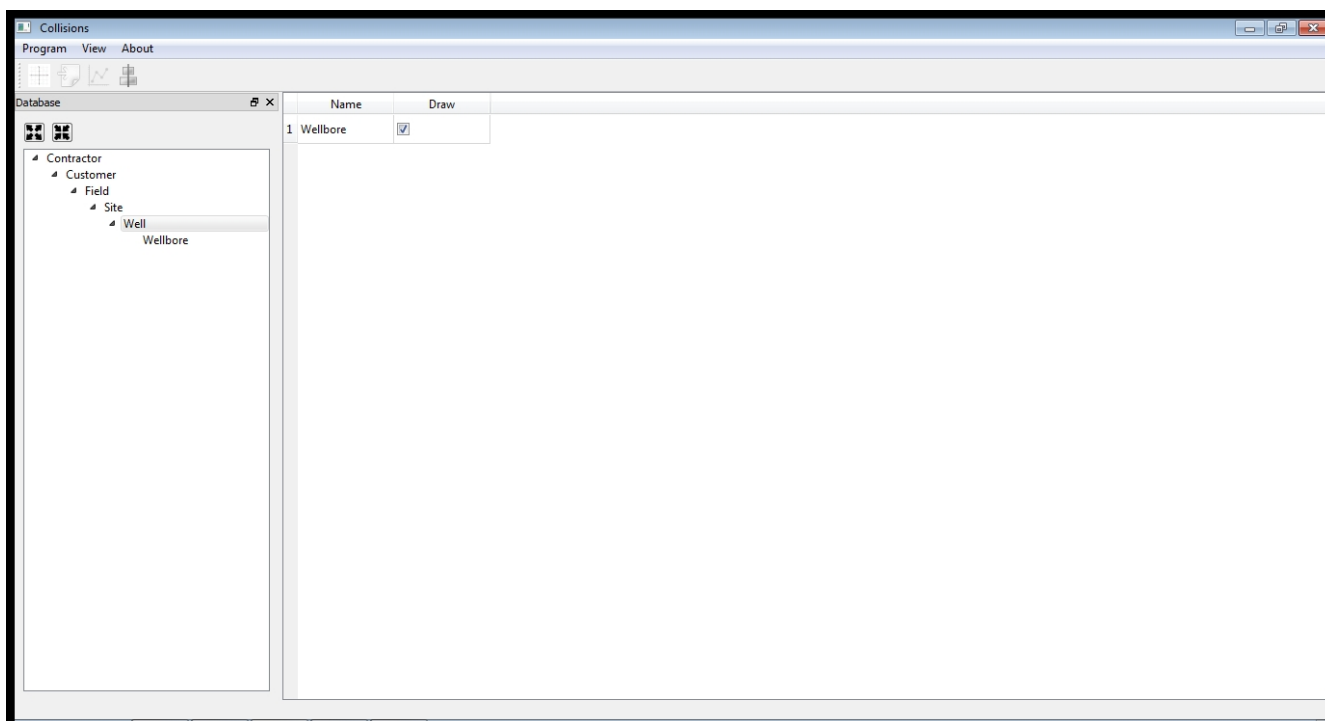


Рис. 14: Таблица стволов

Программа поддерживает импорт замеров из буфера обмена и некоторых форматов текстовых файлов(например, csv). Для импортирования замера необходимо выбрать ствол в дереве базы и нажать на кнопку Import(Импорт) на панели инструментов. (Рис.15)

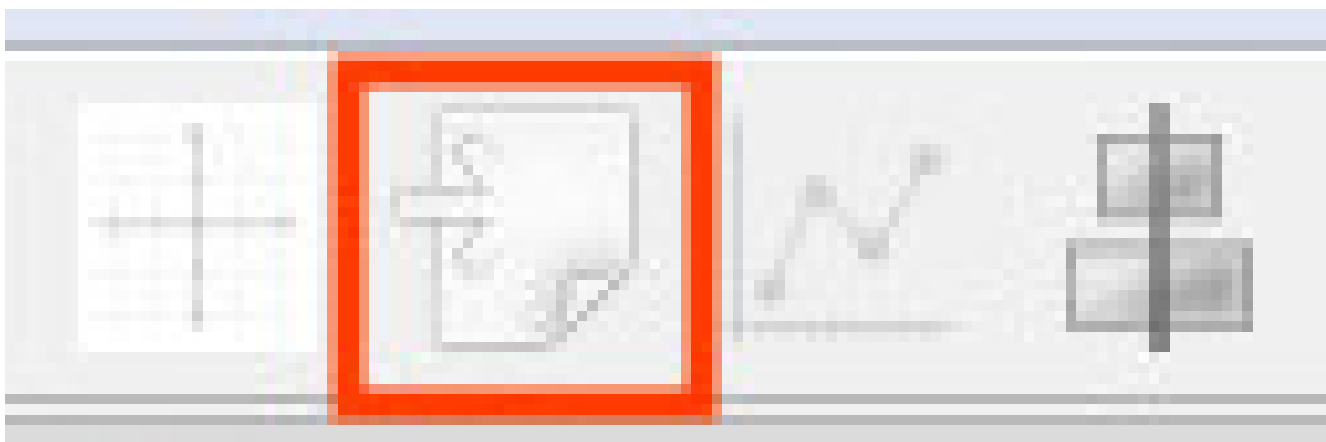


Рис. 15: Кнопка импорта

После нажатия появится мастер импорта замера(Рис.16)

Рис. 16: Мастер импорта замера: (1,2) источник импорта, файл или буфер обмена (3) название замера (4) дата создания замера (4) символ, используемых для разделения столбцов(для CSV это обычно запятая или точка с запятой, для буфера обмена - символ табуляции)

При импорте из файла необходимо ввести путь к файлу в соответствующее поле или нажать на кнопку "...". При нажатии на нее откроется стандартный диалог выбора файлов. После выбора файла(если импорт происходит не из буфера обмена) и заполнения остальных полей формы можно перейти на следующую страницу мастера, это делается кнопкой Next(Далее). На следующей странице необходимо выбрать те столбцы и строки, которых содержат необходимы данные. Для импорта замеров необходимы: измеренная глубина, зенит и азимут. Выделить данные в таблице можно зажав правую кнопку мыши и потянув курсор в нужную сторону. (Рис.17 и Рис.18)

Import wizard

Data

Select the items with data

	1	2	3	4	5	6	7
25	18.08.2012 13:5...	30,0	0,21	199,48	112,96	97,40	97,40
26	18.08.2012 13:5...	40,0	0,19	145,04	142,59	127,03	127,03
27	18.08.2012 13:4...	50,0	0,30	110,47	169,68	154,12	154,12
28	18.08.2012 13:4...	60,0	0,17	0,45	254,74	239,18	239,18
29	18.08.2012 13:3...	70,0	0,39	321,84	263,49	247,93	247,93
30	18.08.2012 13:3...	80,0	0,59	313,88	256,75	241,19	241,19
31	18.08.2012 13:2...	90,0	0,68	296,41	258,80	243,24	243,24
32	18.08.2012 13:2...	100,0	0,46	274,49	262,69	247,13	247,13
33	18.08.2012 13:1...	110,0	0,62	260,57	255,83	240,27	240,27
34	18.08.2012 13:1...	120,0	0,70	235,18	253,33	237,77	237,77
35	18.08.2012 13:1...	130,0	0,46	203,64	259,35	243,79	243,79
36	18.08.2012 13:0...	140,0	0,35	175,59	269,65	254,09	254,09
37	18.08.2012 13:0...	150,0	0,27	47,69	5,90	350,34	350,34

Next Cancel

Рис. 17: Страница с данными

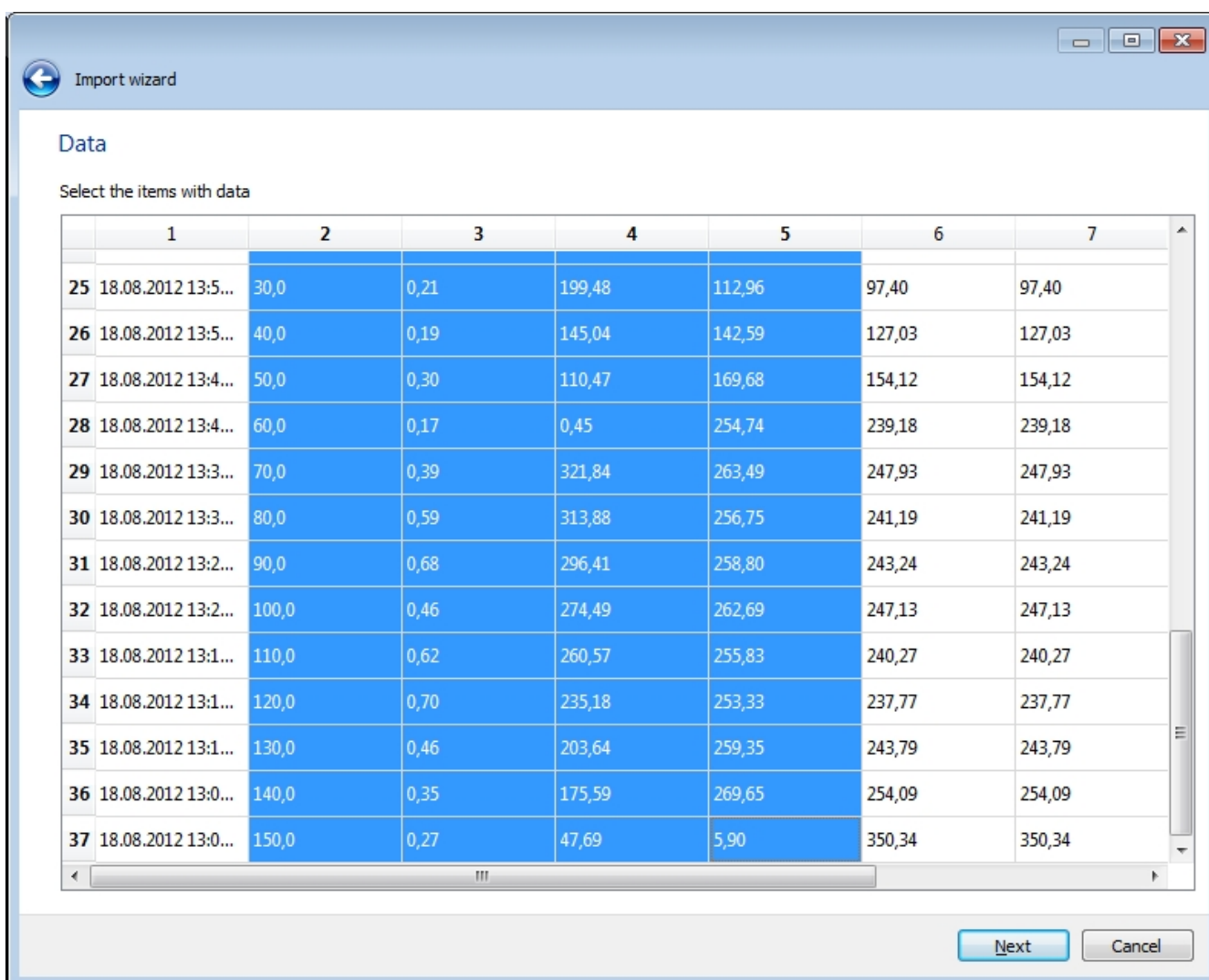


Рис. 18: Выбор элементов с данными

После выделения элементов с данными можно переходить на следующую страницу мастера.(Рис.19)

Import wizard

Measured depth Column 1 Incination Column 2 Azimuth Column 4

	1	2	3	4
1	10	0.23	233.74	109.77
2	20	0.02	247.04	81.94
3	30	0.21	199.48	112.96
4	40	0.19	145.04	142.59
5	50	0.3	110.47	169.68
6	60	0.17	0.45	254.74
7	70	0.39	321.84	263.49
8	80	0.59	313.88	256.75
9	90	0.68	296.41	258.8
10	100	0.46	274.49	262.69
11	110	0.62	260.57	255.83
12	120	0.7	235.18	253.33
13	130	0.46	203.64	259.35
14	140	0.35	175.59	269.65
15	150	0.27	47.69	5.9

Finish Cancel

Рис. 19: Страница выбора столбцов с данными

На этой странице необходимо выбрать какие столбцы содержат нужные данные(глубину, зенит и азимут). В выпадающих списках сверху диалога нужно выбрать соответствующие номера столбцов. После нажатия кнопки Finish(Завершить) в базу будут добавлен замер и соответствующие точки (Рис.20)

	Measured depth	Inclination	Azimuth	Toolface	Device temperature	Easting	Northing	TVD	Subsea TVD	DLS	Draw
1	10	0,23	109,77	0	0	0,03283650237	0,02309072556	9,999919429	9,99992	0,318499519	<input checked="" type="checkbox"/>
2	20	0,02	81,94	0	0	0,05627550865	0,02068714393	19,99989167	19,9999	0,2186009681	<input checked="" type="checkbox"/>
3	30	0,21	112,96	0	0	0,09434852407	0,01570853332	29,99981795	29,9998	0,1999410992	<input checked="" type="checkbox"/>
4	40	0,19	142,59	0	0	0,135044989	-0,01583044905	39,99968541	39,9997	0,1053439675	<input checked="" type="checkbox"/>
5	50	0,3	169,68	0	0	0,1629369264	-0,07887612954	49,99944777	49,9994	0,1597449447	<input checked="" type="checkbox"/>
6	60	0,17	254,74	0	0	0,1331674074	-0,1261310307	59,9992918	59,9993	0,3723083512	<input checked="" type="checkbox"/>
7	70	0,39	263,49	0	0	0,05175677853	-0,1417861443	69,99894816	69,9989	0,2241170487	<input checked="" type="checkbox"/>
8	80	0,59	256,75	0	0	-0,08321541267	-0,1652940164	79,9980961	79,998	0,2081404141	<input checked="" type="checkbox"/>
9	90	0,68	258,8	0	0	-0,2495181531	-0,201325904	89,99656176	89,9966	0,09282332276	<input checked="" type="checkbox"/>
10	100	0,46	262,69	0	0	-0,3873224067	-0,2237810967	99,995587	99,9956	0,2233776568	<input checked="" type="checkbox"/>
11	110	0,62	255,83	0	0	-0,5330713382	-0,2514260419	109,9944866	109,994	0,1725688644	<input checked="" type="checkbox"/>
12	120	0,7	253,33	0	0	-0,7029952418	-0,2982947039	119,9929329	119,993	0,08502519771	<input checked="" type="checkbox"/>
13	130	0,46	259,35	0	0	-0,8403635285	-0,3316798997	129,9919336	129,992	0,2476157235	<input checked="" type="checkbox"/>
14	140	0,35	269,65	0	0	-0,9411248655	-0,3413821129	139,9914213	139,991	0,1319116561	<input checked="" type="checkbox"/>
15	150	0,27	5,9	0	0	-0,8889296654	-0,3988949485	149,9911197	149,991	1,429258547	<input checked="" type="checkbox"/>

Рис. 20: Импортированный замер

Для создания усредненного замера ствола нужно воспользоваться кнопкой Create average(Создать усредненный) на панели инструментов



Рис. 21: Кнопка добавления усредненного замера

После нажатия в базу будет добавлен усредненный замер с именем <Имя ствола>-average. Если замер у ствола был один, то он продублируется.

Проектные замеры необходимы, чтобы для каждого ствола можно было индивидуально задать минимально и максимально допустимые расстояния. В случае, если проектный замер для ствола задан, будут использоваться его настройки, иначе - указанные в настройках.

Приложение Collisions позволяет визуализировать отдельные замеры, скважины и кусты, для отображение графика элемента нужно использовать кнопку Plot(График) на панели инструментов (Рис.22)



Рис. 22: Кнопка просмотра графика

После нажатия кнопки появится диалог просмотра (Рис.23)

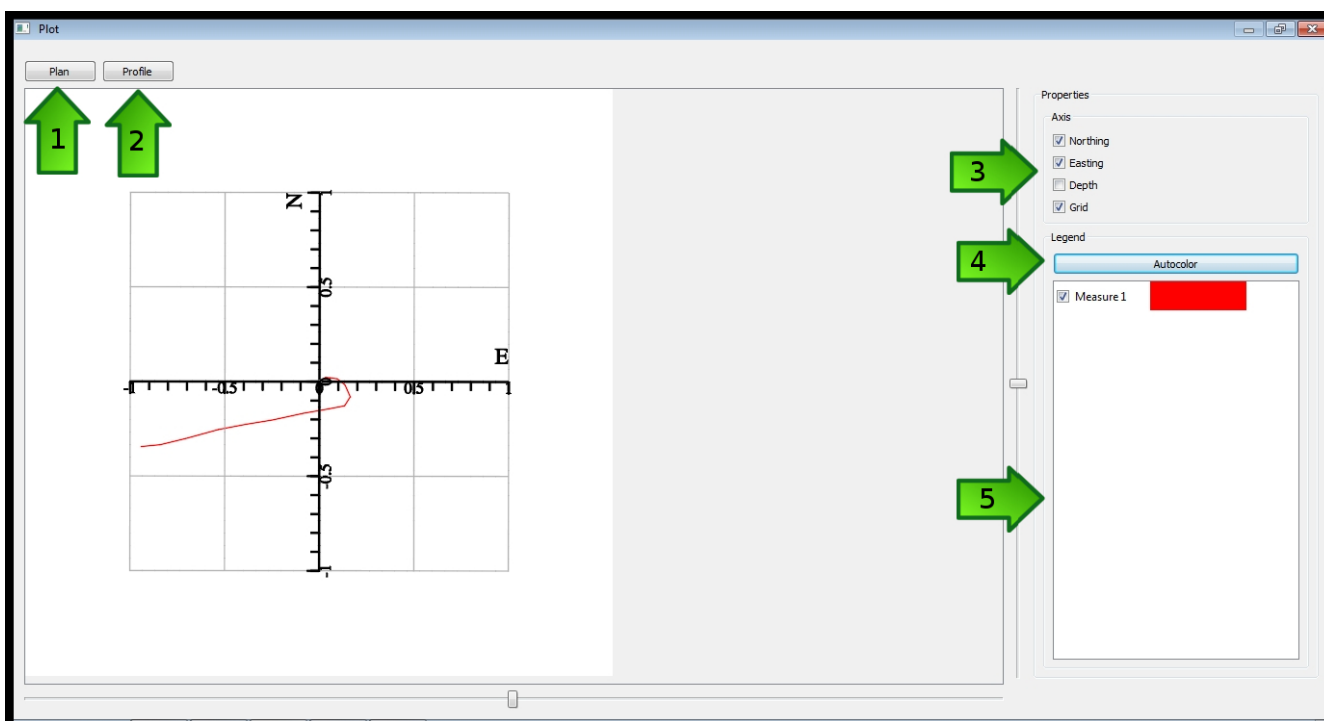


Рис. 23: Диалог просмотра графиков: (1) просмотр плана (2) просмотр профиля (3) управление отображением графика(показ/скрытие осей и сетки) (4) автоматическая раскраска графиков (5) - легенда, настройки цвета и отображение графиков.

В открывшемся диалоге будут изображены:

- Для замера - его графики
- Для ствола - все его замеры
- Для скважины - усредненные замеры всех её стволов
- Для куста - все его скважины

Диалог настроек позволяет настроить значения по-умолчанию для минимально и максимально допустимого расстояния между стволами. (Рис.24)

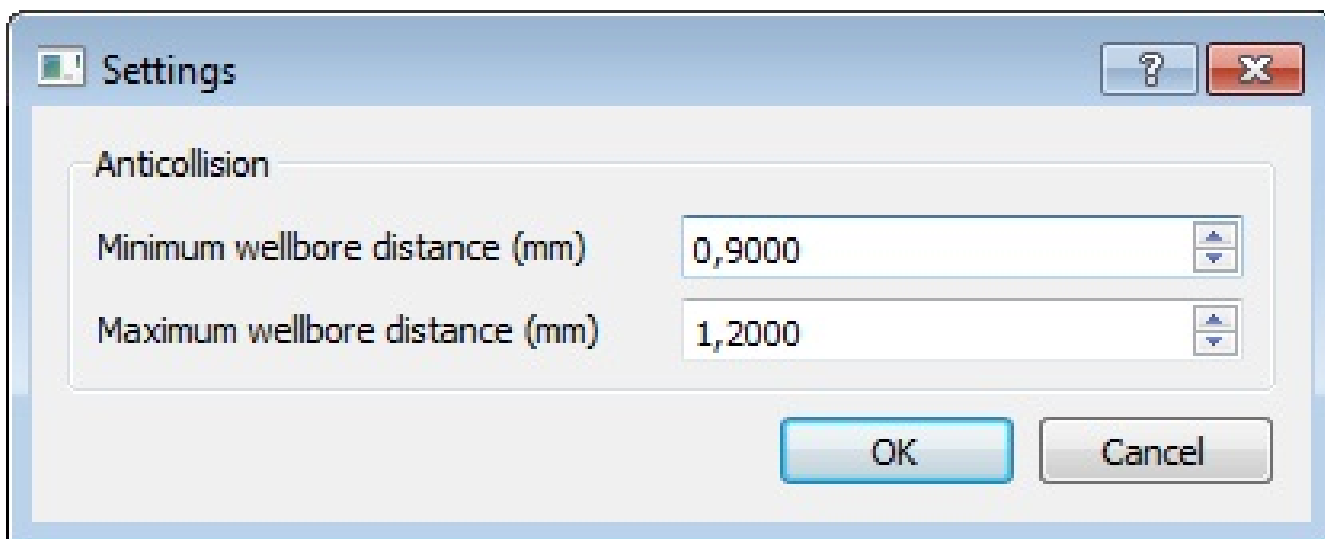


Рис. 24: Диалог настроек

Для просмотра графика расстояний между стволами нужно выбрать куст в дереве базы и воспользоваться кнопкой Anticollision(Предупреждение пересечения стволов) на панели инструментов. В открывшемся диалоге необходимо выбрать ствол, который будет принят за основной, а так же указать с какими стволами будут рассчитываться расстояния

Для просмотра графика расстояний между стволами нужно выбрать куст в дереве базы и воспользоваться кнопкой Anticollision(Предупреждение пересечения стволов) на панели инструментов. В открывшемся диалоге необходимо выбрать ствол, который будет принят за основной, а так же указать с какими стволами будут рассчитываться расстояния (Рис.25)

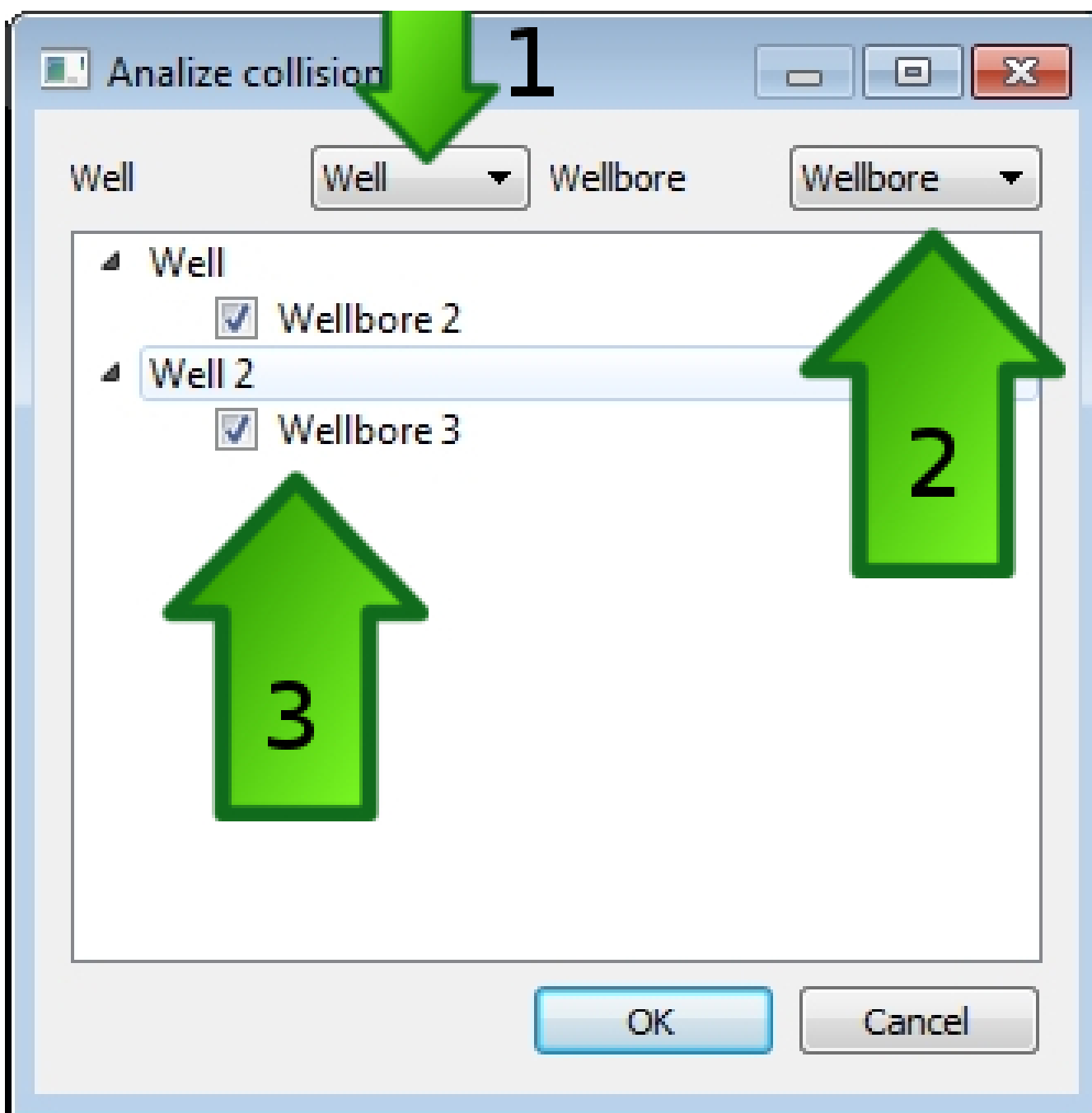


Рис. 25: Диалог настроек процесса предупреждение пересечения стволов: (1) выбор базовой скважины (2) выбор базового ствола (3) дерево стволов для отображения

После нажатия Ok откроется диалог с графиком расстояний. Зеленой зоной выделено расстояние между минимально и максимально допустимыми расстояниями. (Рис.26)

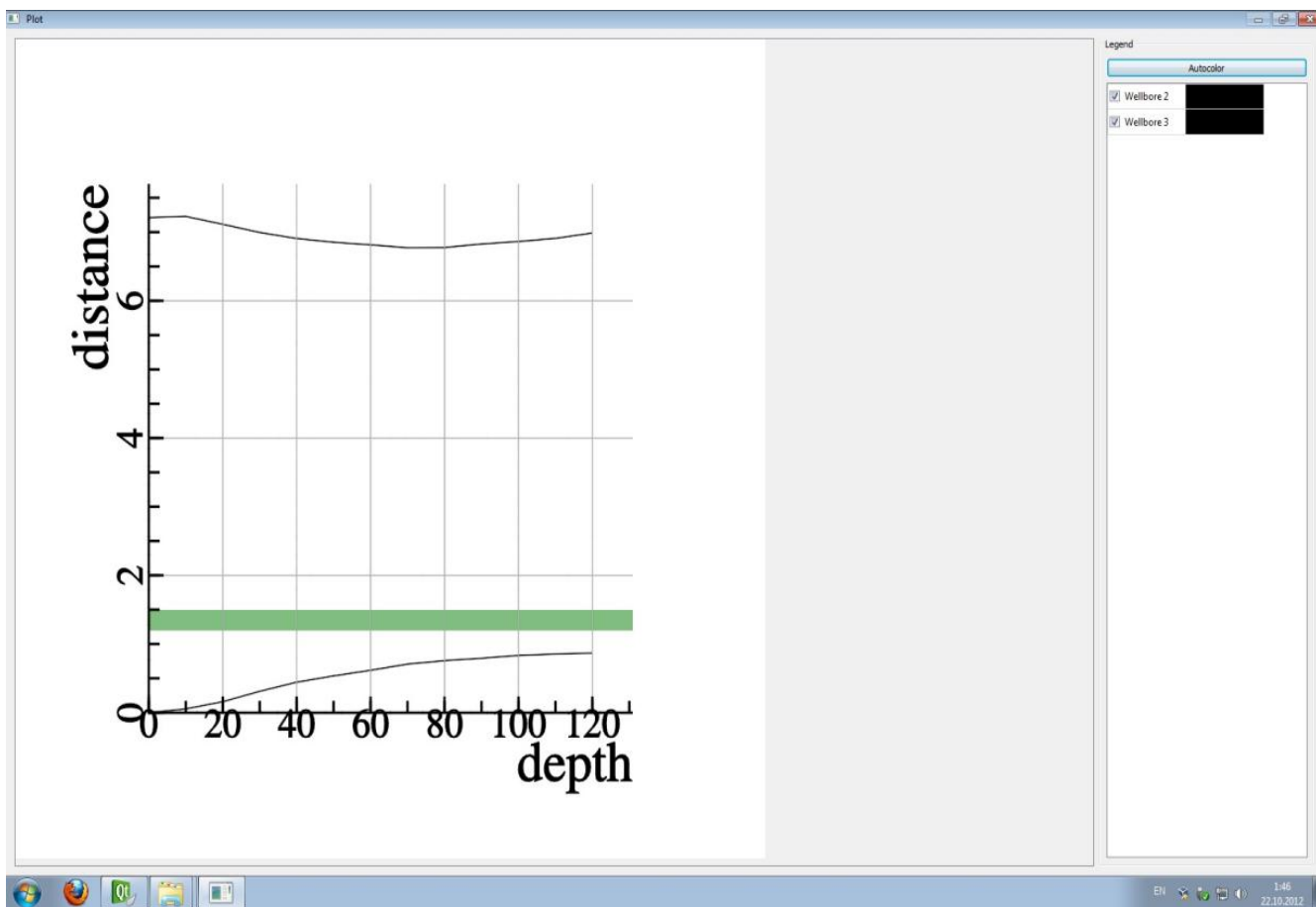


Рис. 26: Диалог графика предупреждения пересечения стволов

Для каждого ствола скважины можно задать координаты устья либо выбрать точку из соседних стволов этой же скважины. Окно свойств вызывается из контекстного меню ствола в дереве базы данных (Рис.27 , Рис.28 и Рис.29)

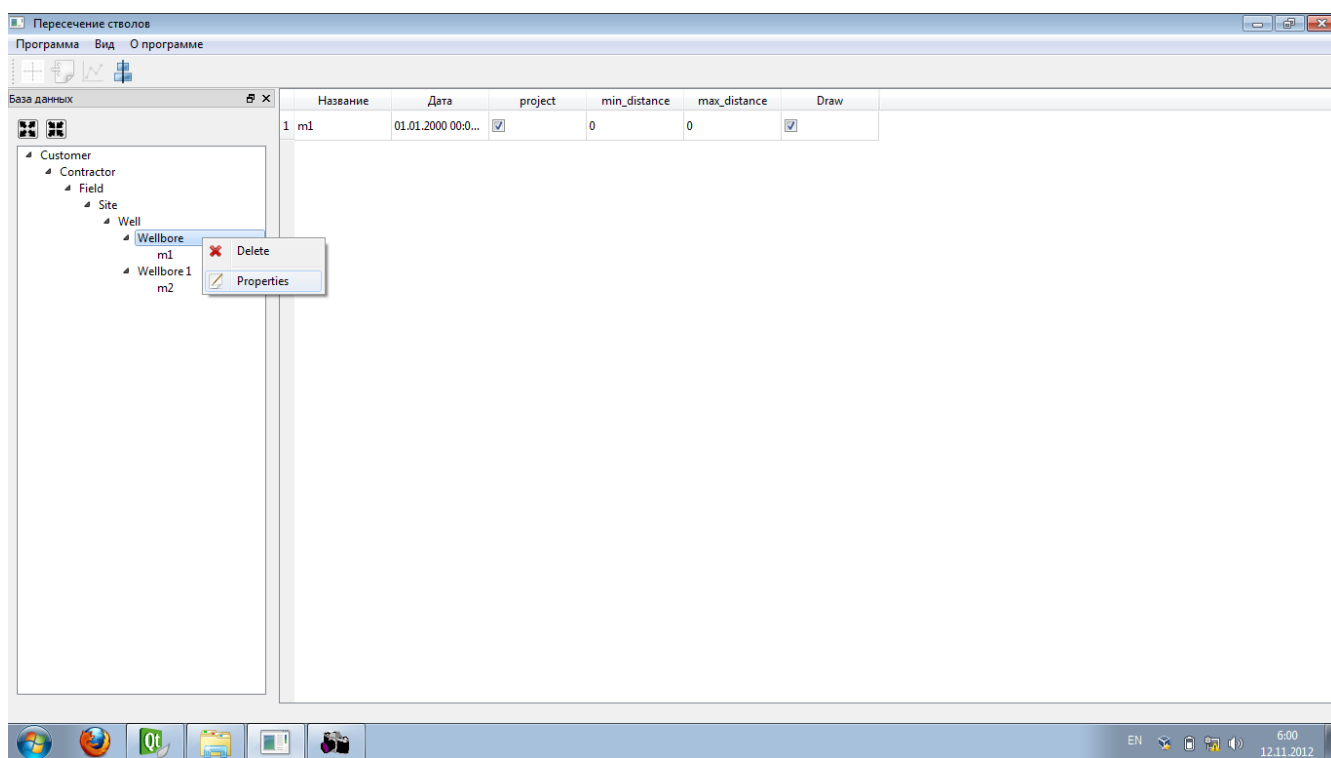


Рис. 27: Меню свойств ствола

The image shows a software window titled "Wellbore properties". It contains a "Head" section with two radio buttons: "Manual" (selected) and "Point of other wellbore". Below the "Manual" option are six input fields with spinners, each with a numerical value: MD (40,0000), Inclination (0,0000), Azimuth (0,0000), Easting (0,0000), Northing (0,0000), and TVD (40,0000). At the bottom of the "Head" section is an empty dropdown menu. At the bottom of the window are two buttons: "OK" and "Отмена".

Parameter	Value
MD	40,0000
Inclination	0,0000
Azimuth	0,0000
Easting	0,0000
Northing	0,0000
TVD	40,0000

Рис. 28: Установка устья ствола вручную

Wellbore properties

Head

☐ Manual

MD: 40,0000

Inclination: 0,0000

Azimuth: 0,0000

Easting: 0,0000

Northing: 0,0000

TVD: 40,0000

☒ Point of other wellbore

m2: 40
m2: 10
m2: 20
m2: 30
m2: 40

OK Отмена

Рис. 29: Установка устья ствола из соседнего ствола той же скважины