**"Разработка автоматизированной системы учета договоров для отеля "**

**Содержание**

Введение

# 1. Описание процесса учета договоров на предприятии

2. Постановка задачи

2.1 Назначение комплекса задач

2.2 Выходные данные

2.3 Входные данные

## 3. Проектирование информационного обеспечения

3.1 Описание задачи

3.2 Проектирование Базы данных методом **"**Сущность - связь**"**

3.2.2 Построение диаграммы ER типа

3.2.3 Генерация набора предварительных отношений

3.2.4 Проверка отношений на НФБК

3.2.5 Исследование окончательного набора отношений на избыточность

## 3.3 Проектирование структуры БД при помощи CASE-средства Erwin

3.3.1 Проектирование логической и физической схемы БД

3.3.2 Исследование информационной модели

4. Программное обеспечение

4.1 Определение функций выполняемых приложением

4.2 Разработка эскиза меню

4.3 Описание компонентов

4.4 Назначение модулей программы

5. Руководство пользователя

5.1 Введение

5.2 Назначение и условие применения

5.2.1 Функции автоматизируемые программно

5.2.2 Технические и программные требования

5.3 Подготовка к работе

5.4 Описание операций

5.4.1 Работа c основным приложением "Таблица основных договоров"

5.4.2 Работа c основным приложением "Таблица дополнительных договоров"

5.5 Рекомендации по освоению

Заключение

Список использованных источников

Приложение A.Описание модулей программы

Приложение В.Отчеты

**Введение**

В нашей стране с каждым днем все больше и больше предприятий. Каждое из них занимается различной деятельностью. Трудно представить работу любой фирмы без заключения каких-либо договоров. Примеров можно привести бесчисленное множество-ведь перед фактическим проведением любого вида работ нужно заключить договор(или множество договоров). В наше время договор имеет огромную ценность так , как если его не заключить-то велика вероятность быть обманутым(заплатят половину суммы, или вообще ничего не заплатят).

В недалеком прошлом на некоторых предприятиях не было даже компьютеров для обработки информации о договорах, приходилось записывать все данные в ведомости и трудно было определить, когда у отедльно взятого договора заканчивалось время его действия(т.е. он являлся просроченным), количество дополнительных договоров и др. информацию относящуюся к договорам. С появлением компьютеров на предприятиях ситуация немного улучшилась, однако программы с которыми работали сотрудники предприятия при учете договоров оставляли желать лучшего.

# 1. Описание процесса учета договоров

В соответствии с пунктом 2 статьи ГК-граждане и юридические лица, осуществляют свои осуществляют свои гражданские права, т.е. заключают договоры в соответствии с нормами, установленными законом (статья 432 ГК). Договор считается заключенным, если между сторонами в требуемой в подлежащих случаях форме , достигнуто соглашение по всем существующим условиям договора. Договор содержит следующие разделы:

1) Предмет договора.

2) Стоимость работ(сумма договора);

3) Порядок и условия расчетов;

4) Срок действия договора(дата договора, срок окончания);

5) Права и обязанности сторон;

6) Порядок сдачи и приемки работ;

7) Приложения.

Договоры бывают:

1)Исходящие от предприятия (оказание работ, услуг; поставки товаров, гражданско-правового характера).

2)Входящие от других предприятий(оказание работ, услуг; поставки товаров и т.д.).

Такие договора заключаются с физическими и юридическими лицами.

Типы договоров:

1) C поставщиками и подрядчиками;

2)За оказанные работы и услуги;

3)Поставка товара.

Для того, что бы иметь полную информацию обо всех заключенных с предприятиями или физическими лицами договорах необходимо вести учет договоров, где отражается информация:

1. В основных договорах:
   * Регистрационный номер.
   * Юридический номер;
   * Дата регистрации договора;
   * Контрагент;
   * Предмет договора;
   * Сумма договора.
2. В дополнительных-то же самое, что и в основных.

Если контрагент заключает не один а несколько договоров, то остальные договоры (все, кроме одного первого ) заносятся в базу дополнительных договоров. Если контрагент заключает новый договор и при этом не вышел срок действия старого, то он заносится в базу дополнительных договоров, иначе старый договор удаляется и новый добавляется в базу основных договоров. Один контрагент может заключать множество договоров, но не наоборот. Контрагентом может являться физическое лицо(один человек), либо юридическое лицо(предприятие). Исполнителем так же может быть физ. лицо, либо юрид. лицо. Контрагент не может заключить дополнительный договор, если он не заключил основного. Информация об исполнителе не может существовать без основного договора. Информация о контрагенте не может существовать без основного договора.

# 2. Постановка задачи

* 1. **Назначение комплекса задач**

В помощь сотрудникам предприятия, занимающимся учетом договоров, разрабатывается автоматизированная система, которая будет выполнять следующие задачи:

1)Регистрация/изменение/удаление основного договора;

2)Регистрация/изменение/удаление дополнительного договора;

3)Добавление/изменение/удаление информации о физическом лице (контрагенте);

4)Добавление/изменение/удаление информации о юридическом лице (контрагенте);

5)Добавление/изменение/удаление планируемой даты;

6)Вывод просроченных договоров;

7)Сортировки основных договоров;

8)Поиска договора;

9)Фильтрации договоров;

10)Вывода диаграмм;

11)Формирования отчетности(по основным, дополнительным договорам);

12)Подсчет количества сделок(договоров) при формировании основных и дополнительных договоров;

13)Экспорт из БД в MsWord.

* 1. **Выходные данные**

Выходной информацией будут отчеты, формулируемые сотрудниками предприятия. Выходная информация для разрабатываемого ПО представлена в таблице 2.1.

Таблица 2.1 - Выходная информация для разрабатываемого ПО

|  |  |
| --- | --- |
| Идентификатор | отчет об основных договорах. |
| Форма представления данных | Приложение Б |
| Периодичность выдачи | По требованию |
| Получатель | Юрист |
| Список реквизитов | Рег.номер,юрид.номер,контрагент, дата договора, срок окончания, предмет договора, сумма договора, исполнитель |
| Идентификатор | отчет о дополнительных договорах |
| Форма представления данных | Приложение Б |
| Периодичность выдачи | По требованию |
| Получатель | Юрист |
| Список реквизитов | Рег.номер,юрид.номер,контрагент, дата договора, предмет договора, сумма договора, дата регистрации |
| Идентификатор | отчет об основных и дополнительных договорах |
| Форма представления данных | Приложение Б |
| Периодичность выдачи | По требованию |
| Получатель | Начальник |
| Список реквизитов | Дата регистрации, дата договора, контрагент, предмет договора, сумма договора. |
| Идентификатор | просроченные договора |
| Форма представоения данных | Экранная форма |
| Периодичность выдачи | По требованию |
| Получатель | Пользователь |
| Список реквизитов | Регистрационный номер, юридический номер, дата регистрации, дата договора, контрагент, предмет договора, сумма договора |

1. **Входные данные**

Входными данными будут документы, приходящие с внешних структур. Входная информация для разрабатываемого ПО представлена в таблице 2.2

Таблица 2.2 – Входная информация для разрабатываемого ПО

|  |  |
| --- | --- |
| Идентификатор | таблица основных договоров |
| Форма представления данных | Экранная форма |
| Периодичность поступления | По заключению договора |
| Источник | Предприятие |
| Список реквизитов | Рег.номер,юрид.номер,контрагент, дата договора, срок окончания, предмет договора, сумма договора, исполнитель |
| Идентификатор | таблица дополнительных договоров |
| Форма представления данных | Экранная форма |
| Периодичность поступления | По заключению основного договора |
| Источник | Предприятие |
| Список реквизитов | Рег.номер,юрид.номер,контрагент, дата договора, предмет договора, сумма договора, дата регистрации |
| Идентификатор | данные о физическом лице |
| Форма представления данных | Экранная форма |
| Периодичность поступления | По заключению основного договора |
| Источник | Предприятие |
| Список реквизитов | ФИО, адрес, домашний телефон, e-mail |
| Идентификатор | данные о юридическом лице |
| Форма представления данных | Экранная форма |
| Периодичность поступления | По заключению основного договора |
| Источник | Предприятие |
| Список реквизитов | Адрес фирмы, контактный телефон, е-mail, количество сделок. |
| Идентификатор | Данные о планируемой дате |
| Форма представления данных | Справочник |
| Периодичность поступления | По требованию начальников |
| Источник | Предприятие |
| Список реквизитов | дата |

**3. Проектирование информационного обеспечения**

**3.1 Описание задачи**

Необходимо разработать автоматизированную систему для учета договоров. Нужно хранить информацию: Об основных договорах (регистрационный номер, юридический номер, контрагент, дата договора, дата договора, срок окончания, исполнитель, предмет договора, сумма договора);

О дополнительных договорах(регистрационный номер, юридический номер, контрагент, дата договора, дата регистрации, срок окончания, предмет договора, сумма договора);

О контрагенте (физическое лицо)(фио, адрес, домашний телефон, e-mail);

О контрагенте (юридическое лицо)(адрес фирмы, контактный телефон, e-mail).

Дополнительного договора нет без основного . Информация о контрагенте не может существовать без основного договора.

Вывести отчеты об:основных договоров, дополнительных договоров, просроченных договоров, основных и дополнительных договоров.

## **Проектирование структуры базы данных методом** **"Сущность - связь"**

### 

### Построение диаграммы ER-типа

1) Определение типов сущностей

Стержневые сущности: Основные договоры, дополнительные договоры.

Характеристические сущности: Юридическое лицо, физическое лицо,информация о исполнителе.

2) Определение типов и характеристик связей

Сущности "Основные договоры" и "Дополнительные договоры" имеют степень отношения 1:n, класс принадлежности необязательный и обязательный соответственно. Следовательно, генерируем 2 отношения по одному на сущность.

Сущности "Планируемая дата" и "Основные договоры" имеют степень отношения 1:n, класс принадлежности не обязательный и обязательный соответственно. Следовательно, генерируем 2 отношения по одному на сущность.

Сущности "Основные договоры" и "Юридическое лицо" имеют степень отношения 1:n, класс принадлежности обязательный и обязательный соответственно. Следовательно, генерируем 2 отношения по одному на сущность. Сущности "Физическое лицо" и" имеют степень отношения 1:1, класс принадлежности обязательный и обязательный соответственно. Следовательно, генерируем 2 отношения, по одному на сущность.

Сущности "Основные" и "Информация о исполнителе" имеют степень отношения 1:n, класс принадлежности обязательный и обязательный соответственно. Следовательно генерируем 2 отношения, по одному отношению на сущность.

По выделенным отношениям построим диаграмму ER-типа:

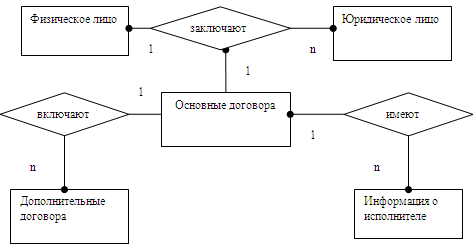


Рис 3.1 - Диаграмма ER-типа

### Генерация набора предварительных отношений

Построим предварительный набор отношений, и определим их атрибуты:

Основные договоры (N\_Agr#, Регистрационный\_номер, Юридический\_номер, Дата\_договора, Дата\_регистрации,Срок\_окончания , Контрагент, Объект\_договора, Сумма\_договора, Планируемая\_дата# );

Дополнительные договоры (N\_Sup\_Agr#, Регистрационный\_номер, Юридический\_номер, Дата\_регистрации, Дата\_договора,Предмет\_договора,Сумма\_договора, Срок\_окончания, Контрагент#);

Юридическое лицо (N\_Contr1#, Контактный\_телефон, Количество\_сделок, e-mail, Контрагент#);

Информация о исполнителе (N\_P#, Адрес,Сотовый\_телефон, Количество\_выполненых\_сделок, Заработная\_плата);

Физическое лицо(N\_Contr2#, Адрес, Телефон, Контрагент#, e-mail).

**3.2.3 Проверка отношений на НФБК**

Отношение Основные договора:

Список функциональных зависимостей:

N\_Agr#🡪Регистрационный номер

N\_Agr#🡪Юридический номер

N\_Agr#🡪Дата договора

N\_Agr#🡪Дата регистрации

N\_Agr#🡪Срок окончания

N\_Agr#🡪Контрагент

N\_Agr#🡪Сумма договора

N\_Agr#🡪Предмет договора

N\_Agr#🡪дата#

Детерминанты: N\_Agr#

Возможные ключи: N\_Agr#

Отношение Основные договор находится в НФБК

Отношение Дополнительные договора:

Список функциональных зависимостей:

N\_Sup\_Agr#🡪Доп\_Регистрационный номер

N\_Sup\_Agr#🡪Доп\_Юридический номер

N\_Sup\_Agr#🡪Доп\_Дата регистрации

N\_Sup\_Agr#🡪Доп\_Дата договора

N\_Sup\_Agr#🡪Доп\_Предмет договора

N\_Sup\_Agr#🡪Доп\_Сумма договора

N\_Sup\_Agr#🡪Доп\_Срок окончания

N\_Sup\_Agr#🡪Доп\_Контрагент#

Детерминанты: N\_Sup\_Agr#

Возможные ключи: N\_Sup\_Agr#

Отношение Дополнительные договора находится в НФБК

Отношение Юридическое лицо:

Список функциональных зависимостей:

N\_Contr1#🡪 Контактный телефон

N\_Contr1#🡪 Количество сделок

N\_Contr1#🡪 e-mail

N\_Contr1#🡪 Контрагент#

Детерминанты: N\_Contr1#

Возможные ключи: N\_Contr1#

Отношение Юридическое лицо находится в НФБК

Отношение Физическое лицо:

Список функциональных зависимостей:

N\_Contr2#🡪 Адрес

N\_Contr2#🡪 Телефон

N\_Contr2#🡪 e-mail

N\_Contr2#🡪Контрагент#

Детерминанты: N\_Contr2#

Возможные ключи: N\_Contr2#

Отношение Физическое лицо находится в НФБК

Отношение Планируемая дата:

Список функциональных зависимостей:

N\_D#🡪Дата

Детерминанты: N\_D#

Возможные ключи: N\_D#

Отношение Спец.Одежда находится в НФБК

3.2.4 Исследование окончательного набора отношений на избыточность

Исследовав построенный предварительный набор отношений на избыточность, и проверив его на НФБК, получим следующие отношения:

Основные договоры (N\_Agr#, Регистрационный\_номер, Юридический\_номер, Дата\_договора, Дата\_регистрации,Срок\_окончания , Контрагент, Объект\_договора, Сумма\_договора, Планируемая\_дата# );

Дополнительные договоры (N\_Sup\_Agr#, Регистрационный\_номер, Юридический\_номер, Дата\_регистрации, Дата\_договора,Предмет\_договора,Сумма\_договора, Срок\_окончания, Контрагент#);

Юридическое лицо (N\_Contr1#, Контактный\_телефон, Количество\_сделок, e-mail, Контрагент#);

Физическое лицо (N\_Contr2#, Адрес, Телефон, Контрагент#, e-mail);

Просроченные договоры (N\_FD#,Рег\_номер,Юрид\_номер,Дата\_регистрации,Дата\_договора,Исполнитель,Сумма\_договора, Объект\_договора, Контрагент#);

В полученном наборе отношений нет ни одного, атрибуты которого можно было бы найти в другом отношении или отношении, полученном из отношений набора серией JOIN операций.

## **Проектирование структуры БД при помощи CASE-средства Erwin**

3.3.1 Проектирование логической и физической схемы БД

Используя построенную диаграмму ER-типа представленную выше, спроектируем базу данных при помощи CASE-средства Erwin. Все связи из диаграммы ER-типа при переносе в нотацию IDEF1X CASE-средства Erwin имеют характеристики:

Таблица 1 – Характеристики связей

|  |  |  |
| --- | --- | --- |
| Связь | Тип связи | Количество элементов |
| Основной договор-дополнительный договор | Идентифицирующая | 1 или много |
| Основной договор-Юридическое лицо | Идентифицирующая | 1 или много |
| Основной договор-Физическое лицо | Идентифицирующая | 1 |
| Дополнительный договор-информация о исполнителе | Не Идентифицирующая | Много ко многим |

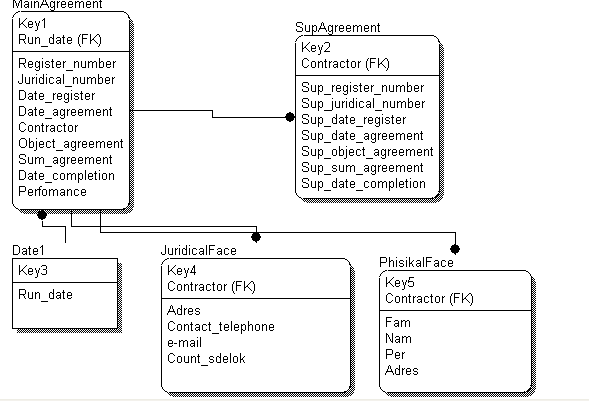


Рис 3.2 - "Логическая модель. Нотация IDEF0"

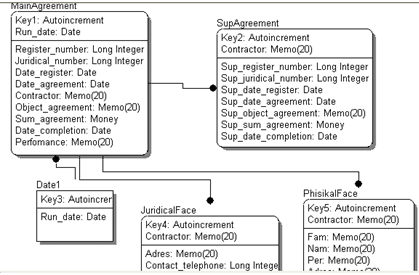


Рис 3.3 – Физическая модель данных

* + 1. Исследование информационной модели

На основе физической модели ERwin был сгенерирован SQL – скрипт (Приложение A), в котором представлены:

* 5 таблиц;
* 10 триггеров;
* 4 генераторов суррогатных ключей;

В результате проверки SQL-скрипта в CASE-средстве Erwin Examiner 4.0, получен отчет Рисунок 3.4.

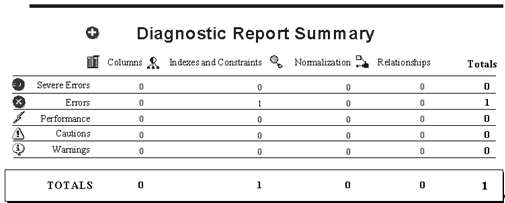


Рисунок 3.4 – Результат проверки SQL-скрипта

**Errors:** в качестве ошибок, в отчете были названы сущности без альтернативных ключей. Для исправления ошибок были добавлены соответствующие альтернативные ключи.

# 4. Программное обеспечение

# Описание функций, выполняемых приложением

Функции, выполняемые программой:

1)Добавление, удаление и редактирование данных (основных договоров,

дополнительных, словаря дат, а так же информации о контрагентах );

2)Просмотр просроченных договоров;

3)Сортировка (упорядочение) записей (строк) по возрастанию;

4)Поиска конкретной записи в БД;

5)Фильтрации данных (отбора записей);

6)Выводит графики;

7)Выполняет экспорт из БД в MsWord;

8)Формирование отчетности для основных и дополнительных договоров.

**4.2 Проектирование ПО с помощью CASE - пакета "Enterprise Architect 4.0"**

**4.2.1 Диаграмма вариантов использования**

Представление вариантов использования включает в себя всех действующих лиц, все прецеденты и их диаграммы для автоматизируемой системы.

Диаграмма вариантов использования разрабатываемой системы представлена на рисунке 4.1.



По каждому варианту использования могут выполняться следующие сценарии:

* Учет основных договоров:

1) добавить основной договор;

2) удалить основной договор;

3) изменить основной договор;

4) сортировка осн. Договоров;

5) поиск осн. Договоров;

6) фильтрация осн. Договоров;

7) вывод диаграмм(информации) об осн. Договоров;

8) экспорт информации об осн. Договоров;

9) вывод отчета (тип1);

10) просмотр просроченных договоров.

* Учет дополнительных договоров
  1. добавить дополнительный договор;
  2. удалить дополнительный договор;
  3. изменить доп. Договор;
  4. вывести отчет.
* Учет информации о контрагентах
  1. Добавить информацию о контрагенте;
  2. Удалить информацию о контрагенте;
  3. Изменить инф. о контрагенте;
  4. Вывести отчет.

**4.5.3 Диаграммы последовательности и классов**

Данная модель описывает взаимодействие между объектами системы и содержит диаграммы классов и диаграммы последовательности.

Диаграммы последовательности разрабатываемой системы представлены и диаграмма классов представлены в приложении Б.

**4.2.4 Диаграмма компонентов**

Представление компонентов содержит информацию о библиотеках кода, исполняемых файлов, динамических библиотеках и других компонентах модели.

Диаграмма компонентов разрабатываемой системы представлена на рисунке 4.5

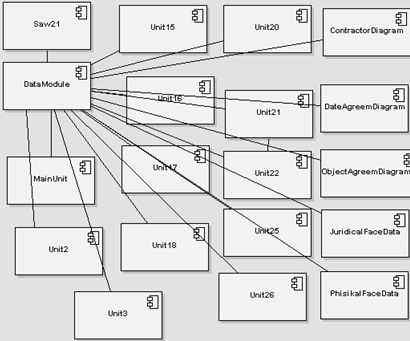


Рисунок 4.5 – диаграмма компонентов

4.3 Описание компонентов

Таблица 4.1 - Описание компонентов

|  |  |
| --- | --- |
| DataSourse | Предназначен для контроля над результатами запроса |
| Query | Предназначен для выполнения SQL запроса к БД |
| DBGrids | Предназначен для вывода результата запроса или целой таблицы БД на экран(работает через TDataSource) |
| TTable | Предназначен для установки связи с БД |
| DecisionCube | Реализует многомерный куб данных |
| DecisionGraph | Отображает графики, соответствующие выбору, сделанному пользователем в многомерном кубе |
| DecisionQuery | Отображает набор данных, используемый для построения куба. |
| DecisionSource | Источник данных, аналогичный DataSource, но приспособленный для задач DecisionCube |
| DecisionGrid | Отображает в табличном виде данные, соответствующие выбору, сделанному пользователем в многомерном кубе |
| WordApplication | Осуществляет соединение с сервером COM |
| WordDocument | Управляет работой сервера COM |
| WordFont | Управляет шрифтом сервера COM |
| WordParagraphFormat | Устанавливает формат абзаца сервера COM |

**4.4 Назначение модулей программы**

Таблица 4.2 - Назначение модулей программы

|  |  |  |
| --- | --- | --- |
| Название модуля | Назначение | Модули, связанные с данным модулем |
| DataModule | Модуль отвечает за связывание программы и базы данных, а так же за корректность выполнения запросов к БД | ALL MODULES |
| Saw21 | Модуль программы, отвечающий за сортировку базы данных. | DataModule |
| MainUnit | Главный модуль, отвечающий за все операции над БД .  Можно добавлять,удалять,редактировать,сортировать, искать, распечатывать,фильтровать основные договоры, а так же  добавлять, удалять,изменять ,печатать дополнительные договоры. Все операции кроме удаления происходят с помощью подключения других модулей. Удаление основных и дополнительных договоров происходит из данной формы. | DataModule,Saw21,Unit3,Unit15,Unit16,Unit17,Unit18,Unit20,Unit21,Unit22,Unit25,Unit26,Unit2, ContractorDiagram |
| Unit3 | Редактировать запись-этот модуль отвечает за редактирование основных договоров . | DataModule |
| Unit2 | Модуль программы, отвечающий за добавление основных договоров. | DataModule |
| Unit15 | Модуль программы, отвечающий за добавление дополнительных договоров. | DataModule |
| Unit16 | Модуль программы, отвечающий за добавление и редактирование дополнительных договоров. | DataModule |
| Unit17 | Модуль программы, отвечающий за вывод информации (зависимость контрагента от суммы) в виде диаграммы. Относится к основным договорам. Вызывается из контекстного меню главного модуля . | DataModule |
| Unit18 | Модуль программы, отвечающий за вывод информации (зависимость контрагента от суммы) в виде диаграммы. Относится к дополнительным договорам.  Вызывается из контекстного меню главного модуля . | DataModule |
| Unit20 | Модуль программы, отвечающий за экспорт данных из БД в MsWord.  Вызывается из контекстного меню главного модуля . | DataModule |
| Unit21 | Модуль программы, отвечающий за отображение просроченных договоров. | DataModule |
| Unit22 | Словарь дат- модуль программы , отвечающий за учет планируемых дат (добавление, сохранение , удаление и просмотр). | DataModule, Unit21 |
| Unit25 | Поиск- модуль программы, отвечающий за поиск в БД. | DataModule |
| Unit26 | Модуль программы, отвечающий за фильтрацию данных в БД. | DataModule |
| ContractorDiagram | Модуль программы, отвечающий за вывод информации(зависимость суммы договоров от даты) в виде диаграммы. Относится к основным договорам. Вызывается из контекстного меню главного модуля. | DataModule |
| DateAgreemDiagram | Модуль программы, отвечающий за вывод информации(зависимость количества контрагентов от даты договоров) в виде диаграммы.Относится к дополнительным договорам. | DataModule |
| ObjectAgreemDiagramm | Модуль программы, отвечающий за вывод информации(зависимость предмета договора от суммы договора) в виде диаграммы.  Относится к основным договорам. | DataModule |
| JuridicalFaceData | Модуль программы, отвечающий за добавление, редактирование и удаление информации о юридическом лице(контрагенте). | DataModule |
| PhisikalFace  Data | Модуль программы отвечающий за добалвение радактирование и удаление информации о физическом лице(контрагенте). | DataModule |

Общая конфигурация системы и маршруты передачи информации между аппаратными устройствами, задействованными в реализации системы изображена на рисунке 4.15

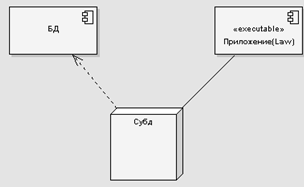


Рисунок 4.15 - диаграмма размещения

**4.5 Потоки данных**

Для формального представления автоматизируемого процесса была использована методология DFD, и позволяющий с ней работать программный продукт **BPWin** компаний **Computer Associates International** и **Wintertree Software**. Методология DFD при помощи CASE-технологии позволяет рассмотреть автоматизируемый процесс и провести ее анализ, выделив функциональные компоненты (процессы) (Смотри Приложение В Рисунки В.1 и В.2).

# 5. Руководство пользователя

## 

## **5.1 Введение**

Приложение "Law" предназначено для небольших предприятий, занимающихся ведением отчетности (учетом договоров). Приложение предназначено для работы юристов, или каких-либо других работников данной сферы. Для крупных предприятий данное приложение не подходит, так как не удовлетворяет всем требованиям.

## 

## **5.2 Назначение и условие применения**

### 

### 5.2.1 Функции, автоматизируемые програмно

Функции, выполняемые программой:

1) Добавление, удаление и редактирование данных (основных договоров,

дополнительных, словаря дат, а так же дополнительной информации);

2) Просмотр просроченных договоров;

3) Сортировка (упорядочение) записей (строк) по возрастанию;

3) Поиска конкретной записи в БД;

4) Фильтрации данных (отбора записей);

5) Выводит графики;

6) Выполняет экспорт в MsWord l;

7) Формирование отчетности для основных и дополнительных договоров.

### 5.2.2 Технические и программные требования

Требования к аппаратному обеспечению:

* x86 – процессор, производительностью не менее Pentium II 500MГц;
* ОЗУ – не менее 32 Mb;
* Манипулятор типа "мышь" или любое другое устройство, совместимое с ним;
* Разрешение экрана – не менее 640 на 480, количество цветов – не менее 16;
* Дисковое пространство ~ 40Mb (вместе с БД).

Требования к програмному обеспечению:

* ОС – Windows 98/ME/2000/XP/2003 Server;
* Установленная программа BDE Administrator, нужная для связи СУБД с БД при помощи псевдонима (в данном случае псевдоним-BCDEMOS).

## **5.3 Подготовка к работе**

Дистрибутив программы содержит следующие файлы:

Таблица 1 – Файлы базы данных

|  |  |
| --- | --- |
| Имя файла | Описание |
| Evgen.DB | Файл Paradox (Основные договора). Может находиться в любой директории. |
| Supagreement.DB | Файл Paradox7(Дополнительные договора).Находится в той же дериктории, что и основные и дополнительные договора. |
| Law.exe | Исполняемый модуль программы. |
| DBFirmData.DB | Файл Paradox (Данные о юридическом лице).Находится в той же директории, что и основные и дополнительные договора. |
| DBPhisikalFace.DB | Файл Paradox (Данные о физическом лице).Находится в той же директории, что и основные и дополнительные договора. |
| Perfomance | Файл Paradox( Данные о исполнителе-физическом лице) |
| Perfomance2 | Файл Paradox(Данные о исполнителе-юридическом лице) |

Общий объем описанных файлов (без файла БД) ~ 40Mb.

Запуск программы осуществляется запуском исполняемого файла(в том случае,если на компьютере установлена программа BDE Administrator).

Для инсталляции BDE следует:

1)Запустить командную строку ОС Windows(Пуск->Выполнить->cmd)

2)Обработать файл bdeinst.dll(находящийся в одной папке вместе с программой) утилитой regsvr32.exe, следующим образом:

regsvr32.exe <Полный путь к файлу>\bdeinst.dll

Например:

regsvr32.exe C:\EvgenLaw3\bdeinst.dll

3)Далее следует запустить BDE и открыть конфигурационный файл под названием IDAPI32.CFG, находящийся в одной директории вместе с программой.

Установка завершена.

Далее следует запустить исполняемый файл Law.exe любыми методами, допускаемыми Windows.

Проверка работоспособности осуществляется простым запуском программы, в случае возникновения ошибки (отсутствие необходимых библиотек, невозможность установить соединение с базой данных и т.д.) программа выдаст соответствующее сообщение об ошибке. Если программа не выдала ошибки, и подсоединение к базе прошло успешно, то программа работоспособна. При возникновении ошибки, а также для добавления пользователя следует обратиться к системному администратору.

## **5.4****Описание операций**

**5.4.1 Работа с основным приложением(таблица основных договоров).**

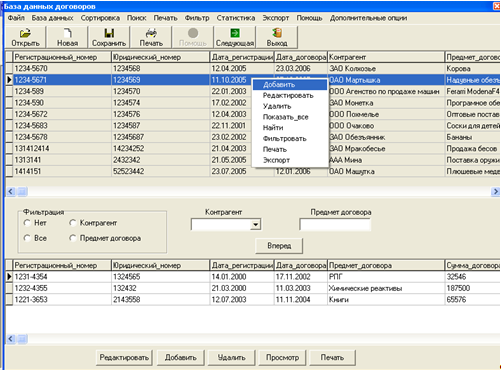


Рисунок 5.1 – Таблица основных и дополнительных договоров

Редактирование таблицы основных договоров осуществляется через диалоговые окна, возникающие при нажатии на соответствующий пункт меню на панели инструментов или при нажатии правой кнопки манипулятора мыши .

Для этой таблицы характерен набор пунктов меню:

* "Добавить" для добавления записи;
* "Удалить" для удаления записи;
* "Редактировать" позволяет изменить данные о договоре;
* "Показать все" возвращает базу в исходное состояние-если с ней были произведены какие-либо изменения(сортировка,поиск,фильтрация и т. д.);
* "Сортировка" производит упорядочение данных ;
* "Поиск" позволяет найти нужную запись;
* "Печать" выводит данные из БД на печать;
* "Фильтр" позволяет просмотреть данные из БД в заданном диапазоне;
* "Статистика" включает:
* Формирование диаграмм на основе данных из БД;
* Просмотр просроченных договоров;
* "Экспорт" включает:
* Перенос данных из БД в Ms Word;
* Перенос данных из БД в Ms Excel.

При изменении основного договора открывается новое окно, где пользователь изменяет данные (Регистрационный номер,юридический номер, дату регистрации, дату договора, срок окончания, контрагента, предмета договора, сумму договора и исполнителя) о договоре.

При добавлении основного договора открывается новое окно, где пользователь вносит данные (Регистрационный номер, юридический номер, дату регистрации, дату договора, срок окончания, контрагента, предмета договора, сумму договора и исполнителя) о договоре, а так же выбрав опцию

"Дополнительные сведения" в зависимости от того, кем является контрагент(т.e. фирма, либо физическое лицо)-появляется таблица, в которую пользователь может внести дополнительные сведения. Если контрагентом является физическое лицо, то пользователь может внести следующие данные(Фамилия, имя, отчество, адрес, телефон и e-mail(если таковой есть)), или внести информацию о контрагенте как о юридическом лице(фирме)(Адрес, контактный телефон, e-mail,количество сделок).

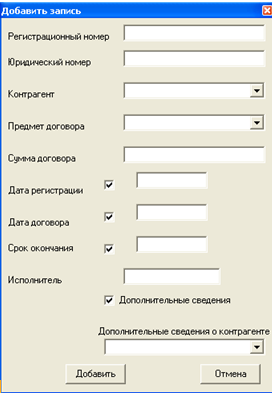


Рисунок 5.2 Добавление основного договора

После того как все данные заполнены, нажимаем кнопку "добавить", и запись об основном договоре появляется в главной таблице. Если же пользователь передумал заносить новый договор в таблицу, то следует нажать кнопку "отмена" после чего он вернется к главной таблице со старыми данными. При изменении данных о об основном договоре выдается то же окно, что и при добавлении осн. договора. При удалении выдается подтверждение на удаление и в зависимости от нажатой кнопки происходит либо удаление, либо отмена.

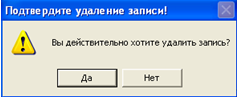


Рисунок 5.3 Запрос на подтверждение удаления основного договора

При нажатии правой кнопки графического манипулятора мыши и удалении записи-договор удаляется без запроса на подтверждение.

"Сортировка" производится по следующим полям:регистрационный номер,юридический номер,контрагент.

Отдельно имеется возможность произвести сортировку по всем полям-кроме исполнителя(т к оно не имеет индеска)

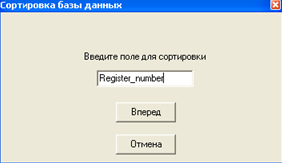


Рисунок 5.4 Сортировка БД основных договоров

"Поиск" осуществляется по :регистрационному номеру,юридическому номеру, дате регистрации, дате договора, предмету договора, контрагенту.

Для поиска нужно выбрать поле ,ввести данные в окно редактироваия и нажатать кнопку "Найти",или отменить нажатием кнопки "Отмена"

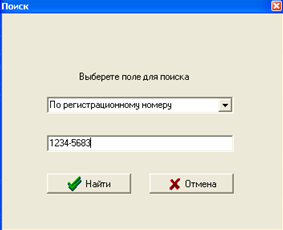
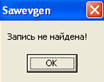


Рисунок 5.5 Поиск в БД основных договоров

Если запись не найдена, то выдается сообщение вида:



"Печать" осуществляется 2 способами:

1)Способом быстрой печати(т е нажатием на соответствующей кнопке меню);

2)Способом обычной печати(т.е. печать производится после предварительного просмотра сформированного документа).

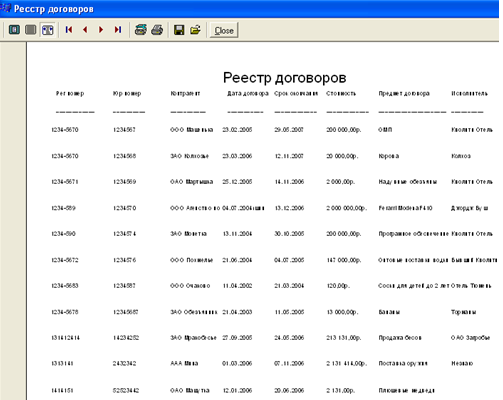


Рисунок 5.6 Выходной документ основных договоров

Для того, что бы произвести фильтрацию необходимо выбрать поле,по которому будет производиться фильтрация(дата регистрации, дата договора, срок окончания, сумма договора) и затем нажать кнопку "Фильтровать", или отменить нажатием кнопки "Отмена". При этом нужно ввести диапазон допустимых значений, которые принимает фильтруемое поле.

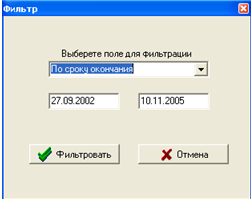


Рисунок 5.7 Фильтрация БД осн. Договоров

"Статистика" включает:

1)Две диаграммы (одна для основных, а другая для дополнительных договоров), отображающие зависимость контрагента (юридическое лицо???) от количества заключенных договоров. По оси X откладывается контрагент; по Y количество заключенных им договоров.

Из приведенной выше диаграммы видно, что:

-Максимальное количество договоров (3 договора) заключило два контрагента (выделены зеленым и синим цветами);

-Минимальное количество договоров (один договор) заключил один контрагент (выделен красным цветом).

2)Просмотр просроченных договоров, т.е. договоров, у которых срок окончания меньше планируемой даты.

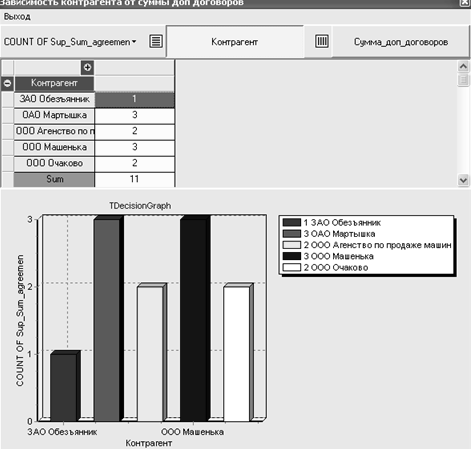


Рисунок 5.8 Диаграмма ависимости контрагента от суммы доп. Договора

Договоры, у которых срок окончания равен планируемой дате, считаются действительными (сравниваются сроки с точностью до дня, а часы и минуты не берутся в счет).

Для просмотра просроченных договоров необходимо:

-В контекстном меню нажав на кнопку "Просроченные договора" загрузить словарь дат

"Экспорт" включает:

1) Перенос данных из БД в MsWord;

При работе с "Экспортом данных из БД в Word" нажатием на соответствующей кнопке меню можно осуществлять следующие действия:

-"Соединиться" - проверить на наличие открытый документ;

-"Открыть нов" - открыть новый документ;

-"Добавить" - перенести в документ одну запись;

-"Сохранить" - сохранение изменений в документе;

-"Печать" - печать документа;

-Дополнительные опции;

При выборе дополнительных опций (если до запуска программы не был открыт документ) открываются следующие возможности:

-Кроме основных полей (регистрационный номер, юридический номер, дата регистрации, дата договора, контрагент, предмет договора,сумма договора) добавляются еще срок окончания и исполнитель при нажатии на кнопках "Срок окончания" и "Исполнитель".

-Добавляется возможность переноса всей таблици в MsWord.

Кнопка "Соединиться" осуществляет открытие и соединение с word , если пользователь случайно закрыл его.

Открыть новый документ можно нажатием на кнопке "Открыть нов".

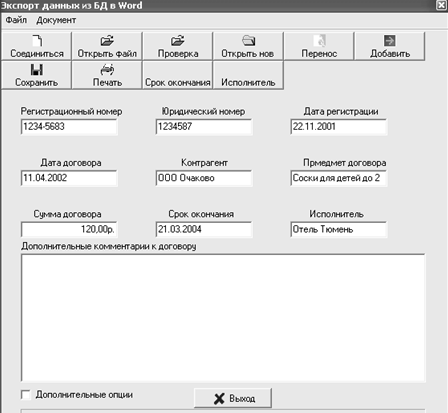


Рисунок 5.11 - Экспорт данных из БД в MsWord

**5.4.2 Работа с основным приложением (таблица дополнительных договоров)**

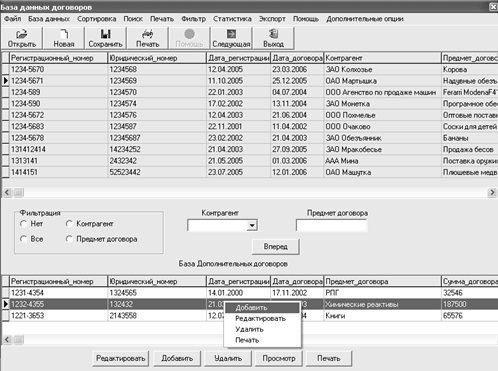


Рисунок 5.12 – Таблица основных и дополнительных договоров

Редактирование таблицы дополнительных договоров осуществляется через диалоговые окна, возникающие при нажатии на соответствующую кнопку или при нажатии правой кнопки манипулятора мыши.

Для этой таблици характерен набор пунктов меню:

* "Добавить" для добавления записи.
* "Удалить" для удаления записи.
* "Редактировать" позволяет изменить данные о договоре.
* "Просмотр" позволяет просмотреть данные перед печатью и потом распечатать.
* "Печать" выводит данные таблици дополнительных договоров на печать.

Все остальные операции (сортировка,поиск,фильтрация и т.д.) бессмысленны, т.к. таблица является детализируемой (а не основной), что ведет за собой нарушение ссылочной целостности данных.

При изменении дополнительного договора открывается новое окно, где пользователь изменяет данные (Регистрационный номер, юридический номер, дату регистрации, дату договора, срок окончанияпредмета договора, сумму договора и исполнителя) о договоре.

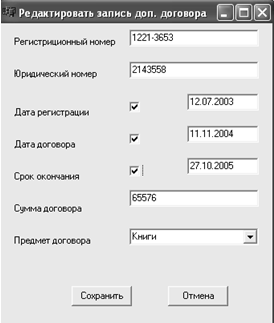


Рисунок 5.13 – Редактирование доп. Договоров

При добавлении дополнительного договора открывается новое окно, где пользователь вносит данные (Регистрационный номер, юридический номер, дату регистрации, дату договора, срок окончания, предмета договора, сумму договора и исполнителя) о договоре. Контрагента нельзя вносить в таблицу дополнительных договоров, т.к. она связана с таблицей основных договоров по этому полю. При внесении новой записи в таблицу дополнительных договоров данные о контрагенте автоматически добавляются из таблици основных договоров .Окно "Добавить запись доп. договора" выглядит так же как и редактирование.

При нажатии на кнопки "Просмотр" и "Печать" программа выполняет те же действия что и в таблице основных договоров. См. стр. 7

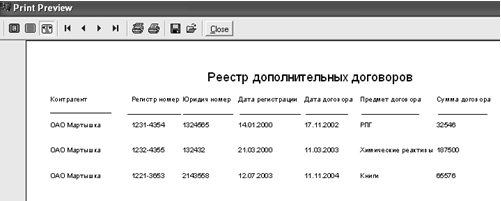


Рисунок 5.14 -Выходной документ дополнительных договоров

## 

## **5.5 Рекомендации по освоению**

Для освоения описываемого приложения пользователю достаточно опыта работы в самых обычных Windows – приложениях, поскольку интерфейс программы прост, и все его элементы снабжены всплывающими подсказками. Для закрепления навыков пользователю предоставляется возможность произвести базовые операции работы с б.д.. А также администратор может предоставить пользователю фиктивную БД, с которой можно отработать основные операции, реализуемые программой, после чего приступать к работе с реальной БД.

## **Заключение**

Приложение "Law" достаточно эффективно решает поставленную задачу, простой и понятный интерфейс программы доступен любому пользователю, имеется весь необходимый набор операций для выполнения поставленных задач.

К достоинствам программы следует причислить отсутствие требования знаний СУБД к пользователю, возможность экспорта данных в средства Ms Office, а так же различные виды печати и богатый набор функций и инструментов для работы с данными.

Недостатками программы являются отсутствие справки .

**Список использованных источников**

1. Томас К., Каролин Б., Анна С. Базы данных. Проектирование, реализация и сопровождение. Теория и практика, Второе издание исправленное и дополненное, Вильяис, М. – 2000г.

# 2. Джен Харрингтон Проектирование объектно-реляционных баз данных, ДМК издательство, М. – 2000г.

3. Глушаков С.В., Ломотько Д.В. Базы Данных.Учебный курс, издательство АСТ, М. – 2002г.

4. Вендров А.М. Проектирование программного обеспечения экономических информационных систем: учебник, финансы и статистика, М. – 2003г.

5. Малыхина M. П. Базы данных: основы, проектирование, использование, СПб., БВХ-Петербург, 2004г.

6.Кириллов В.В. Структуризованный язык запросов (SQL). СПб.: ИТМО, 1994г.

**Приложение А (обязательное)**

**Описание модулей программы**

Основной модуль программы -MainUnit

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "DataModule.h"

#include "MainUnit.h"

#include "Saw21.h"

#include "Unit2.h"

#include "Unit3.h"

#include "Unit13.h"

#include "Unit14.h"

#include "Unit15.h"

#include "Unit16.h"

#include "Unit17.h"

#include "Unit18.h"

#include "Unit19.h"

#include "Unit20.h"

#include "Unit21.h"

#include "Unit22.h"

#include "Unit25.h"

#include "Unit26.h"

#include "ContractorDiagram.h"

#include "DateAgreemDiagram.h"

#include "ObjectAgreemDiagramm.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma link "CSPIN"

#pragma link "FR\_Class"

#pragma link "FR\_Chart"

#pragma link "FR\_ChBox"

#pragma link "FR\_DCtrl"

#pragma resource "\*.dfm"

TForm1 \*Form1;

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::FormCreate(TObject \*Sender)

{

RadioButton3->Checked=false; /\*Установка видимости компонентов при создании формы \*/

Edit1->Text="by Evgen Corp. version1.0(Demo)";

ToolButton9->Enabled=false;

EditObjAgreem->Visible=false;

// DataModule2->Table1->Active=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N15Click(TObject \*Sender)

{

DataModule2->Table1->IndexFieldNames="Register\_number";/\*Сортировка по регистрационному номеру \*/

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N16Click(TObject \*Sender)

{

DataModule2->Table1->IndexFieldNames="Juridical\_number";/\* Сортировка по юридическому номеру\*/

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N17Click(TObject \*Sender)

{

DataModule2->Table1->IndexFieldNames="Contractor";/\* Сортировка по контрагенту \*/

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N18Click(TObject \*Sender)

{

FormBySort->ShowModal(); /Вызов формы сортировки

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

if(RadioButton3->Checked==true)/\*При выборе включаем фльтр\*/

DataModule2->Table1->Filtered=false;

else

{

if (RadioButton1->Checked==true)

DataModule2->Table1->Filter="Contractor='"+CBContract->Text+"'";/\*Фильтр по контрагенту\*/

else if(RadioButton2->Checked==true)

{

DataModule2->Table1->Filter="Object\_agreement='"+EditObjAgreem->Text+"'";/\*Фильтр по предмету договора\*/

}

else DataModule2->Table1->Filter= "(Contractor='"+CBContract->Text+"') and (Object\_agreement="+(DataModule2->Table1->FieldByName("Object\_agreement")->AsString+EditObjAgreem->Text)+")";/\*Фильтр по контрагенту и предмету договора\*/

}

DataModule2->Table1->Filtered=false;/\*Выключаем фильтр\*/

DataModule2->Table1->Refresh();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N9Click(TObject \*Sender)

{

Form2->ShowModal();//Вызов формы "Добавить договор"

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N10Click(TObject \*Sender)

{

Form3->ShowModal();//Вызов формы "Редактировать договор"

Form3->Position=poOwnerFormCenter;/\*Установка свойств формы(позиция и стиль)\*/

Form3->BorderStyle=bsToolWindow;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N11Click(TObject \*Sender)

{

if(Application->MessageBox("Вы действительно хотите удалить запись?","Подтвердите удаление записи!",MB\_YESNO+MB\_ICONEXCLAMATION)==IDYES)

DataModule2->Table1->Delete();

DataModule2->Table1->Refresh();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::ToolButton13Click(TObject \*Sender)

{

Form1->Close();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::ToolButton5Click(TObject \*Sender)

{

if(DataModule2->Table1->Modified==true)//Сохраняем изменения в базе

DataModule2->Table1->Post();

ShowMessage("Вы сохранили изменнения в таблице!!");

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::DBGrid1DblClick(TObject \*Sender)

{

Form3->ShowModal();//Вызов формы "Редактировать договор"

Form3->Position=poOwnerFormCenter;

Form3->BorderStyle=bsToolWindow;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::ToolButton11Click(TObject \*Sender)

{

DataModule2->Table1->Next();//Переходим на след. запись

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N6Click(TObject \*Sender)

{

if(DataModule2->Table1->Modified==true)//Сохраняем изменения в таблице

DataModule2->Table1->Post();

ShowMessage("Вы сохранили изменнения в таблице!!");

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::ToolButton3Click(TObject \*Sender)

{

Form2->ShowModal();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::ToolButton7Click(TObject \*Sender)

{

ReportAgreem->QuickRep1->Print();//Печать

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N27Click(TObject \*Sender)

{

ReportAgreem->QuickRep1->Preview();//Предварительный просмотр перед //печатью

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::QuickReport1Click(TObject \*Sender)

{

ReportAgreem->QuickRep1->Print();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button2Click(TObject \*Sender)

{

Form14->QuickRep1->Preview();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button3Click(TObject \*Sender)

{

Form14->QuickRep1->Print();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button4Click(TObject \*Sender)

{

Form15->ShowModal();

Form15->Position=poOwnerFormCenter;

Form15->BorderStyle=bsToolWindow;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button6Click(TObject \*Sender)

{

if(Application->MessageBox("Вы действительно хотите удалить запись?","Подтвердите удаление записи!",MB\_YESNO+MB\_ICONEXCLAMATION)==IDYES)

DataModule2->Table2->Delete();

DataModule2->Table2->Refresh();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button5Click(TObject \*Sender)

{

Form16->ShowModal();

Form16->Position=poOwnerFormCenter;

Form16->BorderStyle=bsToolWindow;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::DBGrid2DblClick(TObject \*Sender)

{

Form16->ShowModal();

Form15->Position=poOwnerFormCenter;

Form15->BorderStyle=bsToolWindow;

}

void \_\_fastcall TForm1::N29Click(TObject \*Sender)

{

Form17->ShowModal();

Form15->Position=poOwnerFormCenter;

Form15->BorderStyle=bsToolWindow;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N8Click(TObject \*Sender)

{

Form1->Close();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N110Click(TObject \*Sender)

{

Form18->ShowModal();

Form15->Position=poOwnerFormCenter;

Form15->BorderStyle=bsToolWindow;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N30Click(TObject \*Sender)

{

Form19->ShowModal();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N35Click(TObject \*Sender)

{

Form22->ShowModal();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::MsWord1Click(TObject \*Sender)

{

Form20->ShowModal();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N7Click(TObject \*Sender)

{

DBGrid1->Visible=true;

DBGrid2->Visible=true;

EditObjAgreem->Visible=true;

Edit1->Visible=false;

Memo1->Visible=false;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N13Click(TObject \*Sender)

{

DataModule2->Table1->Filtered=false;

DataModule2->Table1->Active=false;

DataModule2->Table1->Active=true;

RadioButton1->Checked=false;

RadioButton2->Checked=false;

RadioButton3->Checked=false;

RadioButton4->Checked=false;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::MsWord21Click(TObject \*Sender)

{

Form23->ShowModal();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N37Click(TObject \*Sender)

{

DataModule2->Table3->Active=false;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N38Click(TObject \*Sender)

{

DataModule2->Table3->Active=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N39Click(TObject \*Sender)

{

Form25->ShowModal();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N40Click(TObject \*Sender)

{

Form26->ShowModal();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N41Click(TObject \*Sender)

{

Form2->ShowModal();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N42Click(TObject \*Sender)

{

Form3->ShowModal();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N43Click(TObject \*Sender)

{

if(Application->MessageBox("Вы действительно хотите удалить запись?","Подтвердите удаление записи!",MB\_YESNO+MB\_ICONEXCLAMATION)==IDYES)

DataModule2->Table1->Delete();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N44Click(TObject \*Sender)

{

DataModule2->Table1->Filtered=false;

DataModule2->Table1->Active=false;

DataModule2->Table1->Active=true;

RadioButton1->Checked=false;

RadioButton2->Checked=false;

RadioButton3->Checked=false;

RadioButton4->Checked=false;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N45Click(TObject \*Sender)

{

Form25->ShowModal();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N46Click(TObject \*Sender)

{

Form26->ShowModal();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N47Click(TObject \*Sender)

{

ReportAgreem->QuickRep1->Preview();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N48Click(TObject \*Sender)

{

Form20->ShowModal();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N49Click(TObject \*Sender)

{

Form15->ShowModal();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N50Click(TObject \*Sender)

{

Form16->ShowModal();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N51Click(TObject \*Sender)

{

DataModule2->Table2->Delete();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N52Click(TObject \*Sender)

{

Form14->QuickRep1->Preview();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N12Click(TObject \*Sender)

{

Form1->frReport1->ShowReport();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N19Click(TObject \*Sender)

{

FormRep->ShowModal();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N55Click(TObject \*Sender)

{

FormRepAgreem->ShowModal();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N57Click(TObject \*Sender)

{

Form4->ShowModal();

}

//---------------------------------------------------------------------------

Unit20

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Saw12.h"

#include "Unit20.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma link "Word\_2K\_SRVR"

#pragma resource "\*.dfm"

TForm20 \*Form20;

//---------------------------------------------------------------------------

\_\_fastcall TForm20::TForm20(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm20::ToolButton2Click(TObject \*Sender)

{

//Прверка наличия открытого документа

if(WordApplication1->Documents->Count==0)

{

Application->MessageBoxA(PChar("В Word нет открытого документа"),"Varning",MB\_OK+MB\_ICONEXCLAMATION);

Abort();

}

WordDocument1->ConnectTo(WordApplication1->ActiveDocument);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm20::FormCreate(TObject \*Sender)

{

DataModule2->Table1->Open();

//Выключение проверок синтаксиса и грамматики что бы не замедлять работу Winword

WordApplication1->Options->CheckSpellingAsYouType=false;

WordApplication1->Options->CheckGrammarAsYouType=false;

if(WordApplication1->Documents->Count==0)

{

ToolButton3->Enabled=false;

ToolButton4->Enabled=false;

ToolButton6->Enabled=false;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm20::ToolButton5Click(TObject \*Sender)

{

//Открытие нового документа

WordApplication1->Documents->Add(EmptyParam, EmptyParam);

ToolButton3->Enabled=true;

ToolButton4->Enabled=true;

ToolButton5->Enabled=true;

ToolButton6->Enabled=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm20::ToolButton6Click(TObject \*Sender)

{

//Перенос в документ всех записей

TBookmark SavePlace;

//Закладка на текущей записи

SavePlace=DataModule2->Table1->GetBookmark();

DataModule2->Table1->First();

StatusBar1->SimpleText="Идет формирование документа";

//Настройка диаграммы

ProgressBar1->Max=DataModule2->Table1->RecordCount;

ProgressBar1->Position=0;

ProgressBar1->Visible=true;

//Цикл по записям

while(! DataModule2->Table1->Eof)

{

ARecordExecuteClick(Sender);

ProgressBar1->Position=ProgressBar1->Position+1;

DataModule2->Table1->Next();

}

//Возвращение на текущую запись

DataModule2->Table1->GotoBookmark(SavePlace);

//Очистка закладки,полосы состояния и диаграммы

StatusBar1->SimpleText="";

ProgressBar1->Visible=false;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm20::ARecordExecuteClick(TObject \*Sender)

{

TVariant snew="\n";

//Перенос в документ одной записи

//DocumentSearch();

WordApplication1->Selection->InsertAfter(snew);

WordApplication1->Selection->InsertAfter(

TVariant(DataModule2->Table1Register\_number->AsString + ' ' +

DataModule2->Table1Juridical\_number->AsString + ' ' +

DataModule2->Table1Date\_register->AsDateTime + ' ' +

DataModule2->Table1Date\_agreement->AsDateTime + ' ' +

DataModule2->Table1Contractor->AsString + ' ' +

DataModule2->Table1Object\_agreement->AsString + ' ' +

DataModule2->Table1Sum\_agreement->AsInteger+ '\n'));

WordApplication1->Selection->InsertAfter(snew);

if(ToolButton7->Enabled==true)

{

WordApplication1->Selection->InsertAfter(

TVariant (DataModule2->Table1Date\_completion->AsString + "Срок окончания \n"));

WordApplication1->Selection->InsertAfter(snew);

}

if(ToolButton8->Enabled==true)

{

WordApplication1->Selection->InsertAfter(

TVariant (DataModule2->Table1Perfomance->AsString + "Исполнитель\n"));

}

WordFont1->ConnectTo(WordApplication1->Selection->Font);

WordParagraphFormat1->ConnectTo(WordApplication1->Selection->ParagraphFormat);

TVariant Direction = wdCollapseEnd;

WordApplication1->Selection->Collapse(&Direction);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm20::ToolButton3Click(TObject \*Sender)

{

//DocumentSearch();

WordApplication1->Dialogs->Item(wdDialogFileSaveAs)->Show(EmptyParam);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm20::ToolButton1Click(TObject \*Sender)

{

//Открытие и соединение с word, если пользователь случайно закрыл его

WordApplication1->Connect();

//Включение видимости сервера

WordApplication1->Visible;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm20::ApplicationEvents1Hint(TObject \*Sender)

{

//Отображение подсказок в строке состояния

StatusBar1->SimpleText=Application->Hint;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm20::FormDestroy(TObject \*Sender)

{

//Разрываем соединение с базой

DataModule2->Table1->Close();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm20::ToolButton4Click(TObject \*Sender)

{

//Предварительный просмотр документа

//DocumentSearch();

WordDocument1->PrintPreview();

WordApplication1->Visible;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm20::BitBtn1Click(TObject \*Sender)

{

Close();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm20::ToolButton15Click(TObject \*Sender)

{

//Открытие файла

WordApplication1->Visible;

if(WordApplication1->Dialogs->Item(wdDialogFileOpen)->Show(EmptyParam)==-1)

{

ToolButton3->Enabled=false;

ToolButton4->Enabled=false;

ToolButton5->Enabled=false;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm20::CheckBox1Click(TObject \*Sender)

{

if(CheckBox1->Checked==true){

ToolButton6->Enabled=true;

ToolButton7->Enabled=true;

ToolButton8->Enabled=true;}

else {

ToolButton6->Enabled=false;

ToolButton7->Enabled=false;

ToolButton8->Enabled=false; }

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm20::FormShow(TObject \*Sender)

{

CheckBox1->Checked=false;

}

//---------------------------------------------------------------------------

//Поиск в БД

#include <vcl.h>

#pragma hdrstop

#include "Unit25.h"

#include "Saw12.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm25 \*Form25;

//---------------------------------------------------------------------------

\_\_fastcall TForm25::TForm25(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm25::ComboBox1Change(TObject \*Sender)

{

if(ComboBox1->ItemIndex==0){ //Установка видимости компонентов, //связанных с данными

EditReg->Visible=true;

EditJur->Visible=false;

EditDateReg->Visible=false;

EditDateAgreem->Visible=false;

EditObjAgreem->Visible=false;

EditContractor->Visible=false;}

else if(ComboBox1->ItemIndex==1){

EditJur->Visible=true;

EditReg->Visible=false;

EditDateReg->Visible=false;

EditDateAgreem->Visible=false;

EditObjAgreem->Visible=false;

EditContractor->Visible=false;}

else {if(ComboBox1->ItemIndex==2){

EditDateReg->Visible=true;

EditJur->Visible=false;

EditReg->Visible=false;

EditDateAgreem->Visible=false;

EditObjAgreem->Visible=false;

EditContractor->Visible=false;

}

else {if(ComboBox1->ItemIndex==3){

EditDateAgreem->Visible=true;

EditDateReg->Visible=false;

EditJur->Visible=false;

EditReg->Visible=false;

EditObjAgreem->Visible=false;

EditContractor->Visible=false;

}

else {if(ComboBox1->ItemIndex==4){

EditObjAgreem->Visible=true;

EditDateReg->Visible=false;

EditJur->Visible=false;

EditReg->Visible=false;

EditDateAgreem->Visible=false;

EditContractor->Visible=false;

}

else {if(ComboBox1->ItemIndex==5){

EditContractor->Visible=true;

EditDateReg->Visible=false;

EditJur->Visible=false;

EditReg->Visible=false;

EditDateAgreem->Visible=false;

EditObjAgreem->Visible=false;

}

}

}

}

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm25::BitBtn2Click(TObject \*Sender)

{

Form25->Close();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm25::BitBtn1Click(TObject \*Sender)

{

if(ComboBox1->ItemIndex==0){ //Поиск в записи в базе данных

DataModule2->Table1->IndexFieldNames="Register\_number";

DataModule2->Table1->SetKey();

DataModule2->Table1->FieldByName("Register\_number")->AsString=EditReg->Text;

Form25->Close();

if(!DataModule2->Table1->GotoKey())

ShowMessage("Запись не найдена!");}

else {if(ComboBox1->ItemIndex==1){

DataModule2->Table1->IndexFieldNames="Juridical\_number"; //

DataModule2->Table1->SetKey();

DataModule2->Table1->FieldByName("Juridical\_number")->AsInteger=StrToInt(EditJur->Text);

Form25->Close();

if(!DataModule2->Table1->GotoKey())

ShowMessage("Запись не найдена!!");}

else {if(ComboBox1->ItemIndex==2){

DataModule2->Table1->IndexFieldNames="Date\_register";

DataModule2->Table1->SetKey();

DataModule2->Table1->FieldByName("Date\_register")->AsDateTime=StrToDate(EditDateReg->Text); //

Form25->Close();

if(!DataModule2->Table1->GotoKey())

ShowMessage("Запись не найдена");}

else {if(ComboBox1->ItemIndex==3){

DataModule2->Table1->IndexFieldNames="Date\_agreement"; //

DataModule2->Table1->SetKey();

DataModule2->Table1->FieldByName("Date\_agreement")->AsDateTime=StrToDate(EditDateAgreem->Text);

Form25->Close();

if(!DataModule2->Table1->GotoKey())

ShowMessage("Запись не найдена");}

else {if(ComboBox1->ItemIndex==4){

DataModule2->Table1->IndexFieldNames="Object\_agreement"; //

DataModule2->Table1->SetKey();

DataModule2->Table1->FieldByName("Object\_agreement")->AsString=EditObjAgreem->Text;

Form25->Close();

if(!DataModule2->Table1->GotoKey())

ShowMessage("Запись не найдена");}

else {if(ComboBox1->ItemIndex==5){

DataModule2->Table1->IndexFieldNames="Contractor";

DataModule2->Table1->SetKey();

DataModule2->Table1->FieldByName("Contractor")->AsString=EditContractor->Text;

Form25->Close();

if(!DataModule2->Table1->GotoKey())

ShowMessage("Запись не найдена");}

}

}

}

}

}

}

//---------------------------------------------------------------------------

Модуль фильтрации БД

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit26.h"

#include "Saw12.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm26 \*Form26;

//---------------------------------------------------------------------------

\_\_fastcall TForm26::TForm26(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm26::BitBtn2Click(TObject \*Sender)

{

Form26->Close();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm26::ComboBox1Change(TObject \*Sender)

{

if(ComboBox1->ItemIndex==0){

Label2->Visible=true;

Label3->Visible=true;

EditDateRegMin->Visible=true;

EditDateRegMax->Visible=true;

EditSumAgrMin->Visible=false;

EditSumAgrMax->Visible=false;

EditDateCompMin->Visible=false;

EditDateCompMax->Visible=false;

EditDateAgreemMin->Visible=false;

EditDateAgreemMax->Visible=false;}

else { if(ComboBox1->ItemIndex==1){

Label2->Visible=true;

Label3->Visible=true;

EditDateAgreemMin->Visible=true;

EditDateAgreemMax->Visible=true;

EditDateRegMin->Visible=false;

EditDateRegMax->Visible=false;

EditSumAgrMin->Visible=false;

EditSumAgrMax->Visible=false;

EditDateCompMin->Visible=false;

EditDateCompMax->Visible=false;}

else {if(ComboBox1->ItemIndex==2){

Label2->Visible=true;

Label3->Visible=true;

EditDateCompMin->Visible=true;

EditDateCompMax->Visible=true;

EditDateAgreemMin->Visible=false;

EditDateAgreemMax->Visible=false;

EditDateRegMin->Visible=false;

EditDateRegMax->Visible=false;

EditSumAgrMin->Visible=false;

EditSumAgrMax->Visible=false;}

else {if(ComboBox1->ItemIndex==3){

Label2->Visible=true;

Label3->Visible=true;

EditSumAgrMin->Visible=true;

EditSumAgrMax->Visible=true;

EditDateCompMin->Visible=false;

EditDateCompMax->Visible=false;

EditDateAgreemMin->Visible=false;

EditDateAgreemMax->Visible=false;

EditDateRegMin->Visible=false;

EditDateRegMax->Visible=false;

}

}

}

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm26::FormShow(TObject \*Sender)

{

Label2->Visible=false;

Label3->Visible=false;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm26::BitBtn1Click(TObject \*Sender)

{

if(ComboBox1->ItemIndex==0){

DataModule2->Table1->IndexFieldNames="Date\_register";

DataModule2->Table1->SetRangeStart();

DataModule2->Table1->FieldByName("Date\_register")->AsDateTime=StrToDate(EditDateRegMin->Text);

DataModule2->Table1->SetRangeEnd();

DataModule2->Table1->FieldByName("Date\_register")->AsDateTime=StrToDate(EditDateRegMax->Text);

DataModule2->Table1->ApplyRange();

DataModule2->Table1->Refresh();

Form26->Close();}

else {if(ComboBox1->ItemIndex==1){

DataModule2->Table1->IndexFieldNames="Date\_agreement";

DataModule2->Table1->SetRangeStart();

DataModule2->Table1->FieldByName("Date\_agreement")->AsDateTime=StrToDate(EditDateAgreemMin->Text);

DataModule2->Table1->SetRangeEnd();

DataModule2->Table1->FieldByName("Date\_agreement")->AsDateTime=StrToDate(EditDateAgreemMax->Text);

DataModule2->Table1->ApplyRange();

DataModule2->Table1->Refresh();

Form26->Close();}

else {if(ComboBox1->ItemIndex==2){

DataModule2->Table1->IndexFieldNames="Date\_completion";

DataModule2->Table1->SetRangeStart();

DataModule2->Table1->FieldByName("Date\_completion")->AsDateTime=StrToDate(EditDateCompMin->Text);

DataModule2->Table1->SetRangeEnd();

DataModule2->Table1->FieldByName("Date\_completion")->AsDateTime=StrToDate(EditDateCompMax->Text);

DataModule2->Table1->ApplyRange();

DataModule2->Table1->Refresh();

Form26->Close();}

else {if(ComboBox1->ItemIndex==3){

DataModule2->Table1->IndexFieldNames="Sum\_agreement";

DataModule2->Table1->SetRangeStart();

DataModule2->Table1->FieldByName("Sum\_agreement")->AsString=EditSumAgrMin->Text;

DataModule2->Table1->SetRangeEnd();

DataModule2->Table1->FieldByName("Sum\_agreement")->AsString=EditSumAgrMax->Text;

DataModule2->Table1->ApplyRange();

DataModule2->Table1->Refresh();//

Form26->Close();}

}

}

}

}

//---------------------------------------------------------------------------

**Приложение Б (обязательное)**

**Отчеты**

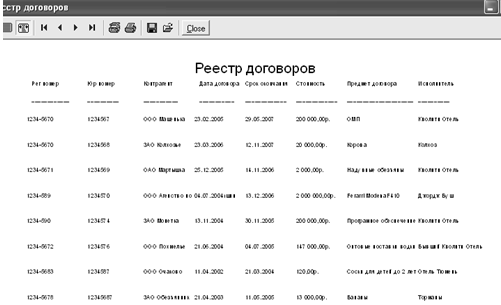


Рисунок А.1 - Отчет по основным договорам

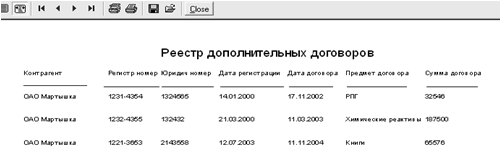


Рисунок A.1 - Отчет по дополнительным договорам