**Пример выполнения индивидуального задания по курсу «Моделирование бизнеса»**

**по теме «Учет заказов предприятия «Озеленитель»»**

**Содержание**

Введение

1. Постановка задачи. Описание и анализ предметной области

2. Функции предметной области задачи. Построение модели IDEF0 для индивидуальной задачи (функциональная модель IDEF0). Функциональные блоки (процессы), подпроцессы, интерфейсные дуги. Глоссарий. Контекстная диаграмма, диаграммы декомпозиции. Процесс построения IDEF0.

3. Перечень документов для задания и документооборот. Функциональная методика потоков данных (диаграмма потоков данных DFD). Потоки данных, процессы (работы) преобразования входных потоков данных в выходные, внешние сущности, накопители данных (хранилища). Процесс построения DFD.

4. Постановка задачи. Описание выходной информации (отчеты, справки, экранные формы). Описание входной информации (документы, справочники).

5. Стоимостный анализ (Activity Based Costing). Центры затрат ABC. Величины затрат на компоненты, рабочая сила, управление и временные характеристики работы – Duration (продолжительность) и Frequency (частота) выполнения.

6. Построение диаграммы декомпозиции в нотации IDEF3. Элементы IDEF3-диаграммы: работы, связи, перекрестки и объекты ссылок. Диаграммы декомпозиции.

6. Диаграмма сущность-связь (ER-диаграмма).

7. Определение информационных связей объектов и построение информационно-логической модели (ИЛМ) данных (модель IDEF1X).

8. Нормализация полученных отношений

9. Язык UML. Диаграммы классов

10. Построение логической и физической структуры базы данных

Выводы

Основные аббревиатуры

Список использованной литературы

**Введение**

**Цель работы**:разработка моделей бизнес-процессов информационной системы, автоматизирующей работу менеджера по работе с контрактами и клиентами (их заключение, сопровождение, расторжение) предприятия «Озеленитель», также предоставление новых возможностей клиентам фирмы (отслеживание статуса длительных и сложных заказов, быстрая связь с менеджером проекта, фирмы).

Данная система повысит качество сервиса для клиентов фирмы. Позволит менеджеру оперативно согласовывать заказы/контракты или заключать новые заказы.

**1. Постановка задачи. Описание и анализ предметной области «Учет заказов предприятия «Озеленитель»»**

**1.1 Описание процессов предметной области**

Муниципальное предприятие города Армавира «Озеленитель» создано на базе тепличного комплекса в феврале 1993 года. МП г. Армавира «Озеленитель» является членом Ассоциации цветоводов и озеленителей России. «Озеленитель» тесно сотрудничает с родственными предприятиями городов и посёлков края – Сочи, Кропоткин, Тихорецка, Краснодара, Успенского, Новокубанска, Туапсе, с предприятиями других краёв и областей – Ростова, Таганрога, Ставрополя, Пятигорска, Нальчика.

Основное направление деятельности «Озеленителя» - производство работ по озеленению города и текущему содержанию городских зеленых насаждений, работы по ландшафтному дизайну. Кроме того, предприятие осуществляет производство более ста наименований цветочной рассады, более ста наименований горшечных растений, выращивание срезочной продукции и изготовление из нее букетов, корзин, венков, композиций, флористические работы.

Основные цеха предприятия - цех зеленого строительства, цех цветоводства. В состав предприятия входит цех по изготовлению малых архитектурных форм из дерева, бетона. Также в состав предприятия входят участок механизации - это 18 единиц техники и 26 человек механизаторов, который помимо выполнения ремонтных работ для нужд предприятия освоил производство тротуарной плитки.

Клиенты предприятия звонят на фирму, согласовывают заказ с менеджером. Они передают ему список того, что хотят заказать, менеджер согласно прайс-листу выставляет счет, а также уточняет то, какие виды продукции не могут быть поставлены, каковы сроки предоставления заказа, другие особенности заказа. В результате клиент получает ответ (письмо, факс, что-то другое) в котором менеджер описывает результат запроса, и если клиент согласен, то он связывается с менеджером и подтверждает заказ. Менеджер предъявляет требование по оплате заказа, после выполнения этих требований фирма приступает к выполнению заказа.

На данный момент такая процедура сопровождения заказов не является оптимальной. Все операции по сопровождению заявки осуществляются в ручном режиме. С ростом объемов заказов увеличивается общее время на обработку заказов менеджером. Имеющаяся на данный момент процедура обработки и заключения заказов не имеет практически никакой автоматизации.

Менеджер помимо мероприятий по сопровождению заявок выполняет ряд других работ. Соответственно рассеивается внимание, необходим значительный промежуток времени для переключения на работы, связанные с заказами. Так как заказы поступают в произвольные моменты времени, менеджеру приходится по нескольку раз переключаться с одной работы на другую, необходимо вспоминать детали каждого заказа, искать записанную ранее информацию по заказу в блокноте или другом источнике. И эти проблемы возникают не только на стадии размещения и согласования заказа.

Ряд других проблем связан с формированием ответа. Для составления ответа на заказ уходит много времени. Необходимо найти позицию в прайсе фирмы, определить цену и вставить ее в ответ. После того как данный документ будет отослан заказчику, менеджеру необходимо проследить оплату заказа, а затем отдать команду на исполнение.

Клиенту фирмы очень проблематично отследить объем выполненных работ по своему заказу. Ему необходимо позвонить менеджеру, сказать какой заказ его интересует и хорошо, если у менеджера данная информация под рукой или в голове. В противном случае менеджеру необходимо время для выяснения статуса заказа и объема выполненных по нему работ. Это опять ведет к увеличению нагрузки на менеджера и выбивает его из спокойного ритма работы.

На предприятии также встречается проблема, когда заказ выполнен, а все сроки по его передачи заказчику прошли, по причине неявки заказчика в установленный срок, и товар может попросту испортиться.

Так как это муниципальное предприятие, оно производит работы для муниципалитета. И опять-таки с информированием муниципальных руководителей о сроках, объемах работ могут возникать проблемы и накладки.

Данные проблемы мы попытаемся решить с помощью внедрения информационной системы (ИС) для отдела заказов на данном предприятии.

В разрабатываемой ИС предполагается автоматизировать работу менеджера по работе с контрактами и клиентами (их заключение, сопровождение, расторжение), также предоставление новых возможностей клиентам фирмы (отслеживание статуса длительных и сложных заказов, быстрая связь с менеджером проекта, фирмы).

Данная система повысит качество сервиса для клиентов фирмы. Позволит менеджеру оперативно согласовывать заказы/контракты или заключать новые заказы.

Применение ИС для решения описанных выше проблем позволит ликвидировать некоторые проблемы, а также сведет к минимуму негативные последствия других проблем на предприятии.

Если затянуть с разработкой и внедрением ИС, то невозможно будет увеличивать экономические показатели предприятия более высокими темпами.

Также если решать эти проблемы не в комплексе, а от случая к случаю, по мере жесткой необходимости в их решении, то процесс автоматизации отдела заказов затянется на неопределенный срок. При таком подходе к решению проблем, возможно появление новых проблем связанных с взаимодействием уже работающих (разработанных и запущенных) подсистем (программ), решающих конкретные узкие задачи. В итоге это приведет к переделке уже реализованных механизмов и созданию полноценной ИС.

**2. Функции предметной области задачи (функциональная модель IDEF0)**

На сегодняшний день проблема автоматизирования ручного труда крайне важна. Поэтому для современного предпринимателя работа без ЭВМ просто невозможна, так как отсутствие таковой системы обработки информации повлечёт за собой огромные затраты рабочего времени, а также сам процесс обработки будет длиться долго, в то время как с помощью автоматизированной системы это можно сделать гораздо быстрее и эффективнее. В настоящее время на данном предприятии документация практически ведется вручную, что очень затрудняет работу.

Исходная информация поступает от заказчиков в форме заказов, контрактов, договоров с приложением спецификаций на необходимую продукцию.

Целью решения задач является обеспечение персонала информацией о наличии заказов, состоянии их выполнения. Цель реализуется путем организации обработки информации и решением задач по поступившим заказам и фактического исполнения. Заказанная продукция поставляется по договорам поставки.

Перечень решаемых задач:

-планирование заказов на продукцию;

-учет выполнения заказов в целом по объему поставок с указанием времени, качества, отклонений от запланированного графика поставок. Здесь предельно решается поисковая задача оперативного контроля по заказам, в т.ч. по отдельным заказам.

Выделены **Функции:**

* учет и анализ посещаемости клиентов;
* формирование и ведение списка клиентов и их заказов на фирме, поддержание актуальности вводимых данных;
* оперативное получение актуальной информации в автоматическом режиме;
* формирование групп пользователей с присвоением каждому пользователю определенных прав доступа к информации;
* создание разнообразных статистических и аналитических отчетов для всех групп пользователей.

**2.1 Функциональная модель IDEF0**

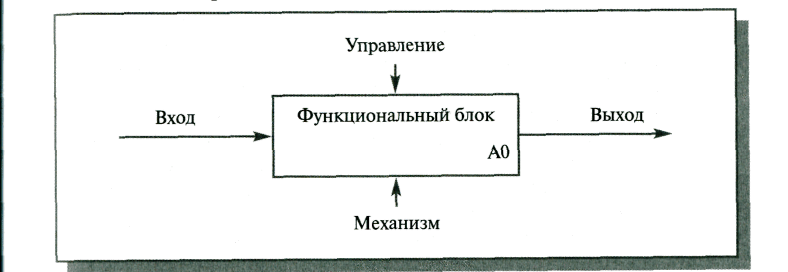
**Методология IDEFO** представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области.

В основе методологии лежат четыре основных понятия: функциональный блок, интерфейсная дуга, декомпозиция, глоссарий.

**Функциональный блок**(Activity Box) представляет собой некоторую конкретную функцию в рамках рассматриваемой системы. По требованиям стандарта название каждого функционального блока должно быть сформулировано в глагольном наклонении.

На диаграмме функциональный блок изображается прямоугольником. Каждая из четырех сторон функционального блока имеет свое определенное значение (роль), при этом:

* верхняя сторона имеет значение «Управление» (Control);
* левая сторона имеет значение «Вход» (Input);
* правая сторона имеет значение «Выход» (Output);
* нижняя сторона имеет значение «Механизм» (Mechanism).

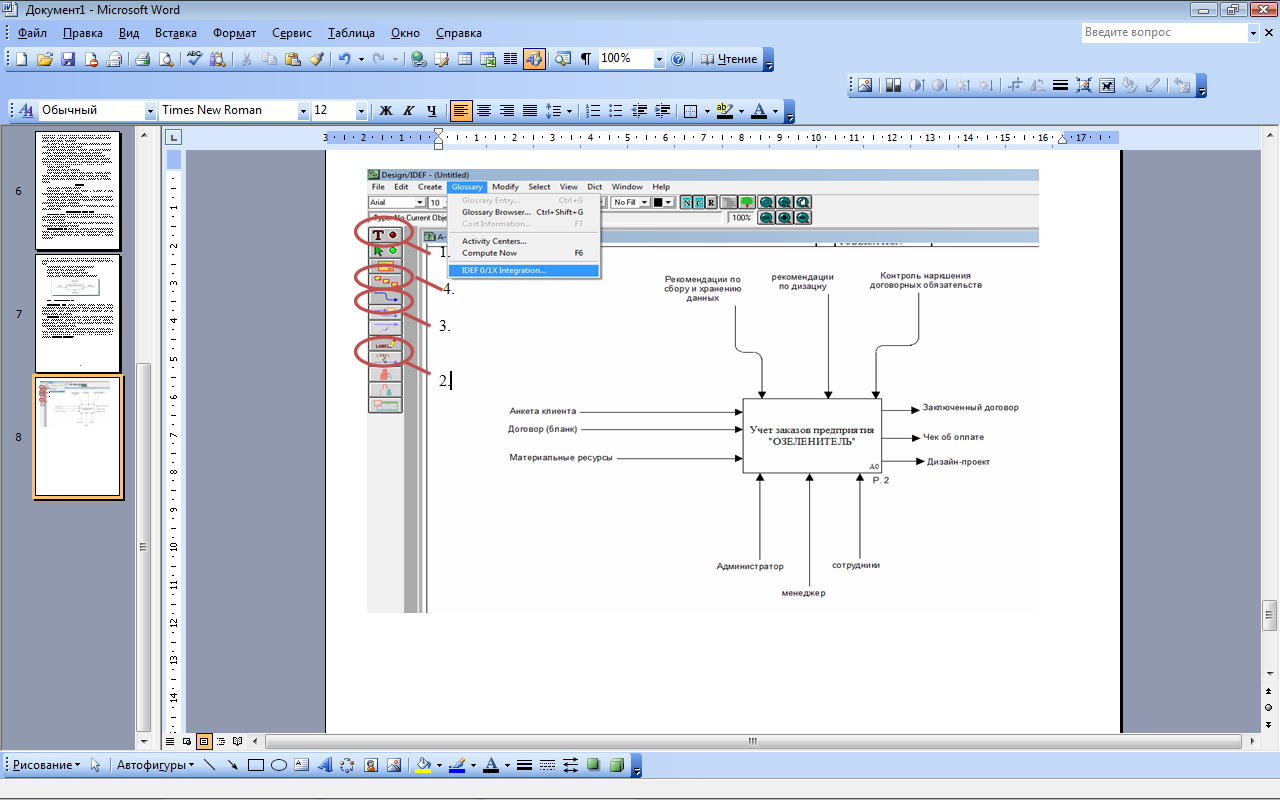


**Рис. 1 Функциональный блок**

**Интерфейсная дуга**(Arrow) отображает элемент системы, который обрабатывается функциональным блоком или оказывает иное влияние на функцию, представленную данным функциональным блоком. Интерфейсные дуги часто называют потоками или стрелками.

**Декомпозиция**(Decomposition) является основным понятием стандарта IDEF0. Принцип декомпозиции применяется при разбиении сложного процесса на составляющие его функции. При этом уровень детализации процесса определяется непосредственно разработчиком модели.

Модель IDEF0 всегда начинается с представления системы как единого целого - одного функционального блока с интерфейсными дугами, простирающимися за пределы рассматриваемой области. Такая диаграмма с одним функциональным блоком называется **контекстной диаграммой**.



**Рис. 2 Контекстная диаграмма**

Для того, чтобы создать контекстную диаграмму выполняем следующие действия:

1. Открываем программу Design IDEF;
2. Выбираем вкладку File/New;
3. Появляется окно Select New Page Type, в Methodology выбираем IDEF0;



1. Появляется функциональный блок А0;
2. На панели инструментов выбираем кнопку с изображением буквы Т (1), вводим текст внутри блока;
3. Для описания всех стрелок выбираем кнопку Label (2);
4. Выбираем кнопку (3) и проводим стрелки от текста к функциональному блоку. В итоге получилась контекстная диаграмма.

Входными данными являются:

* Анкета клиента;
* Договор ( шаблон);
* Материальные ресурсы.

Выходные Данные:

* Заключенный договор;
* Чек об оплате;
* Дизайн - проект;

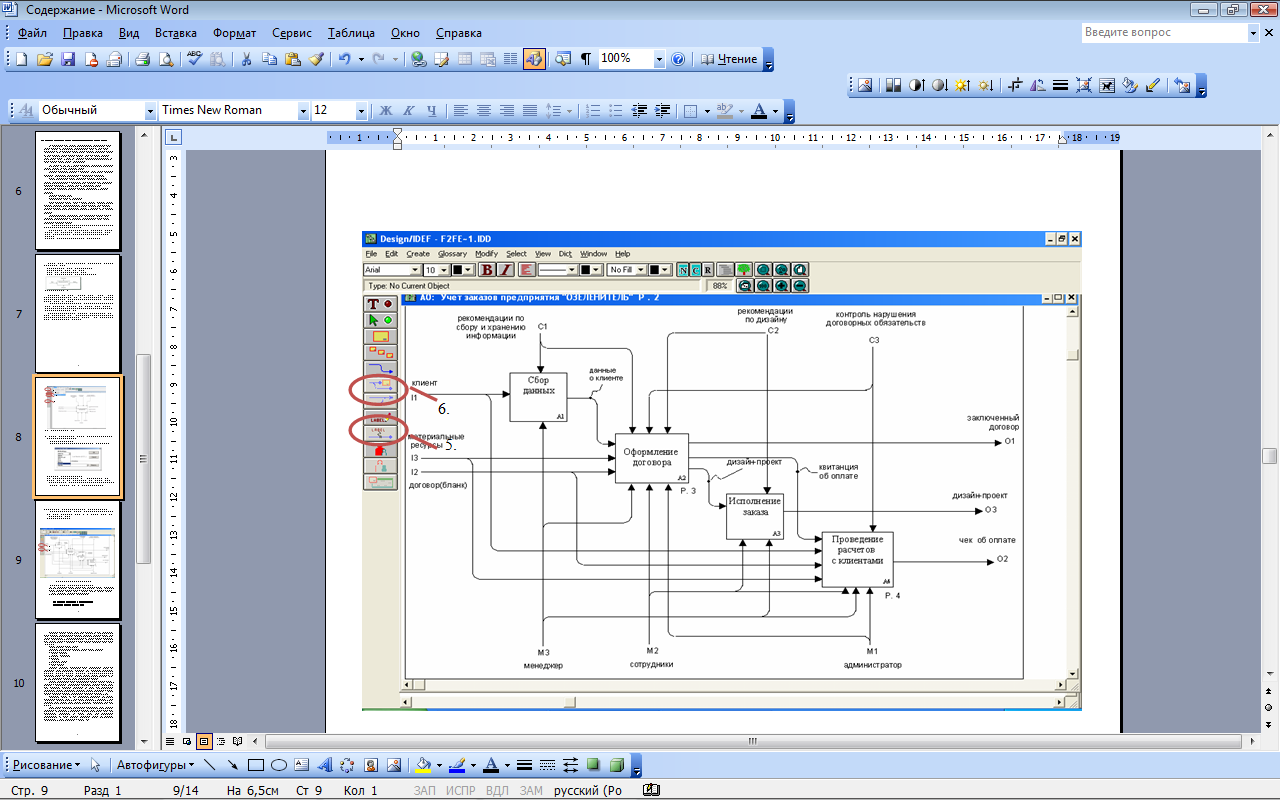
Управление:

* Рекомендации по сбору и хранению данных;
* Рекомендации по дизайну;
* Контроль нарушения договорных обязательств;

Механизм:

* Администратор;
* Сотрудники;
* Менеджер.

1. Переходим к декомпозиции диаграммы:
2. Выбираем вкладку Create/Decompose. Появляется новый лист.
3. На панели инструментов выбираем кнопку (4) и указываем нужное количество блоков;
4. Выполняем пункты 5,6;



**Рис.3 Диаграмма декомпозиции (IDEF0)**

12.Чтобы подписать стрелку, используем кнопку (5).

13.Для разветвления стрелок прежде всего выделяем стрелку и используем кнопку (6), затем подводим курсор к нужному блоку (он начинает мигать);

В итоге получилась **Диаграмма декомпозиции (IDEF0)**

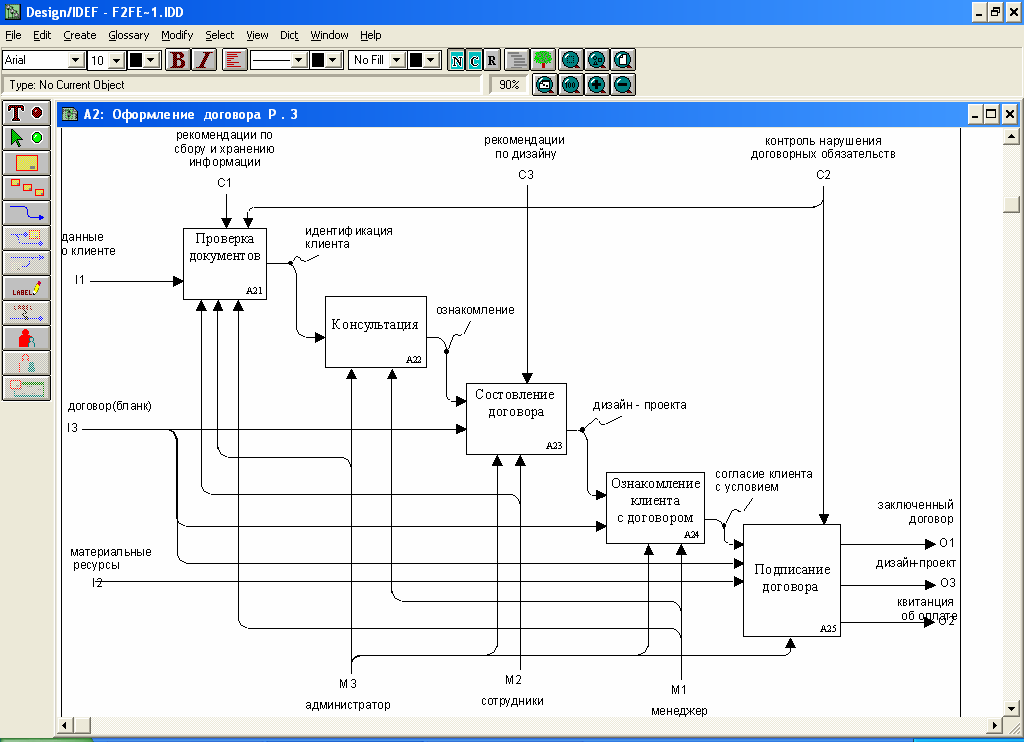
В нее входят 4 процесса:

* Сбор данных;
* Оформление договора;
* Исполнение заказа;
* Проведение расчетов с клиентами.

Два из этих процессов декомпозируем на подпроцессы. Процесс «Оформление договора» разбивается на:

* Проверка документов;
* Консультация;
* Составление договора ;
* Ознакомление клиента с договором;
* Подписание договора.

Выполнив декомпозицию, получим:

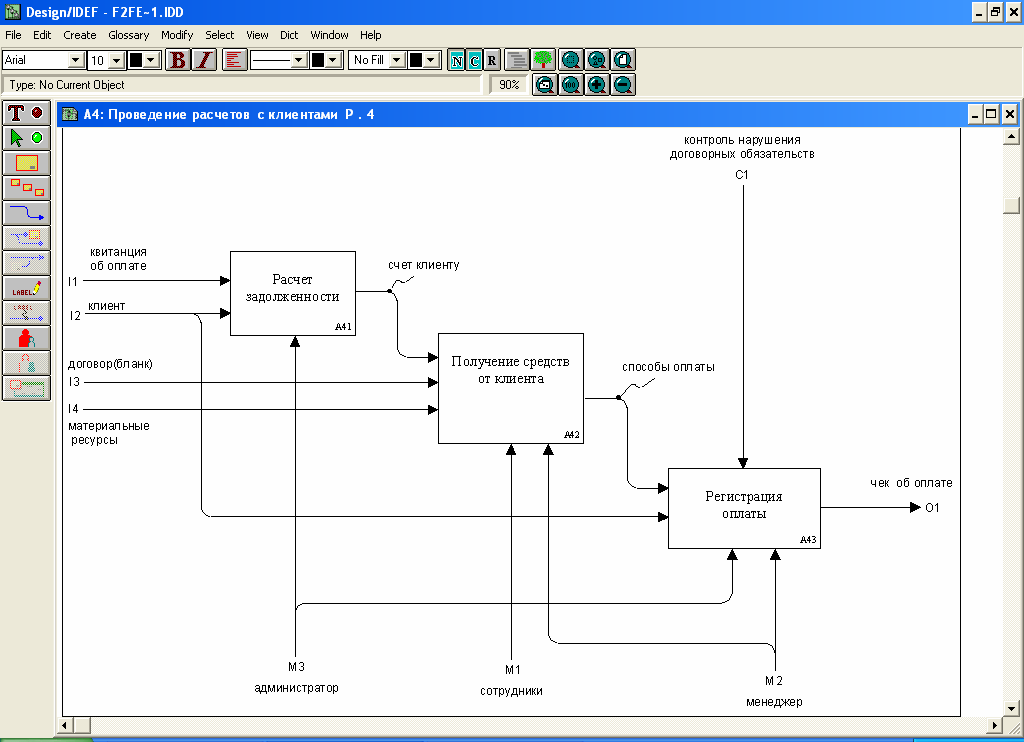


**Рис.4 Диаграмма декомпозиции второго уровня**

**« Оформление договора»**

Процесс «Проведение расчетов с клиентами» разбивается на:

* Расчет задолженности;
* Проведение средств от клиента;
* Регистрация оплаты.



**Рис. 5 Диаграмма декомпозиции второго уровня**

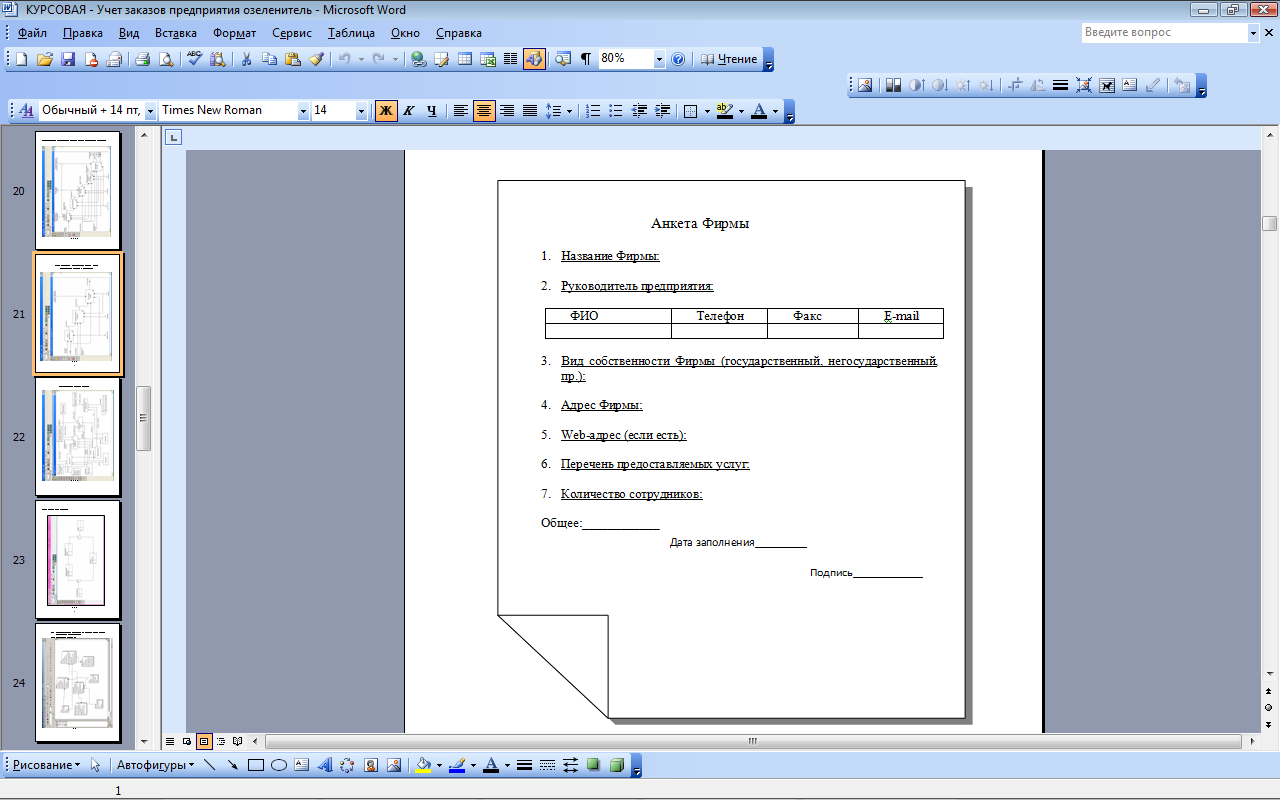
**«Проведение расчетов с клиентами»**

1. **Перечень документов и документооборот (диаграмма потоков данных DFD)**

ИС «Учет заказов предприятия Озеленитель» включает данные о Фирме, клиентах и заказах, о сотрудниках и квитанции об оплате.

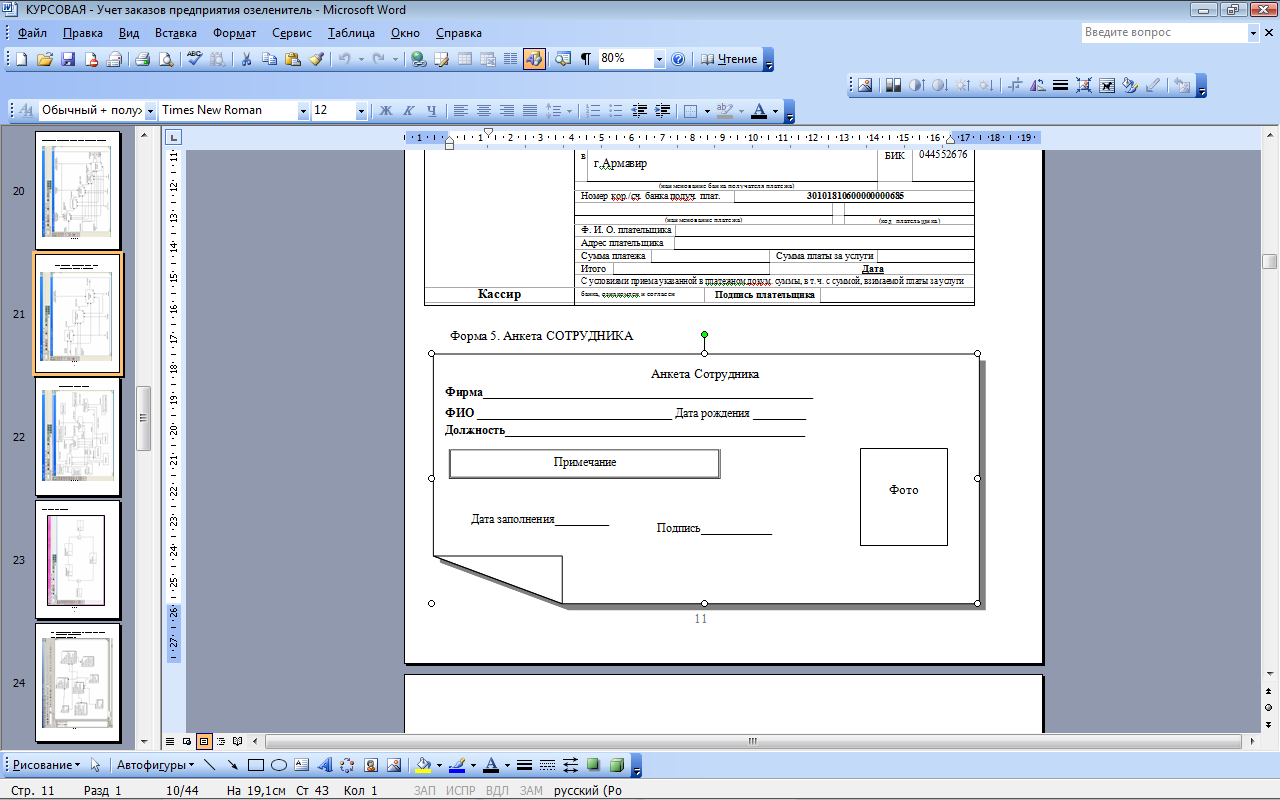
Данные о Фирме помещаются в Анкете Фирмы. Этот документ содержит следующие составные поля:

* Название Фирмы;
* Руководитель предприятия;
* Вид собственности;
* Адрес Фирмы;
* Адреса филиалов;
* Web – адрес;
* Перечень предоставляемых услуг;
* Количество сотрудников.



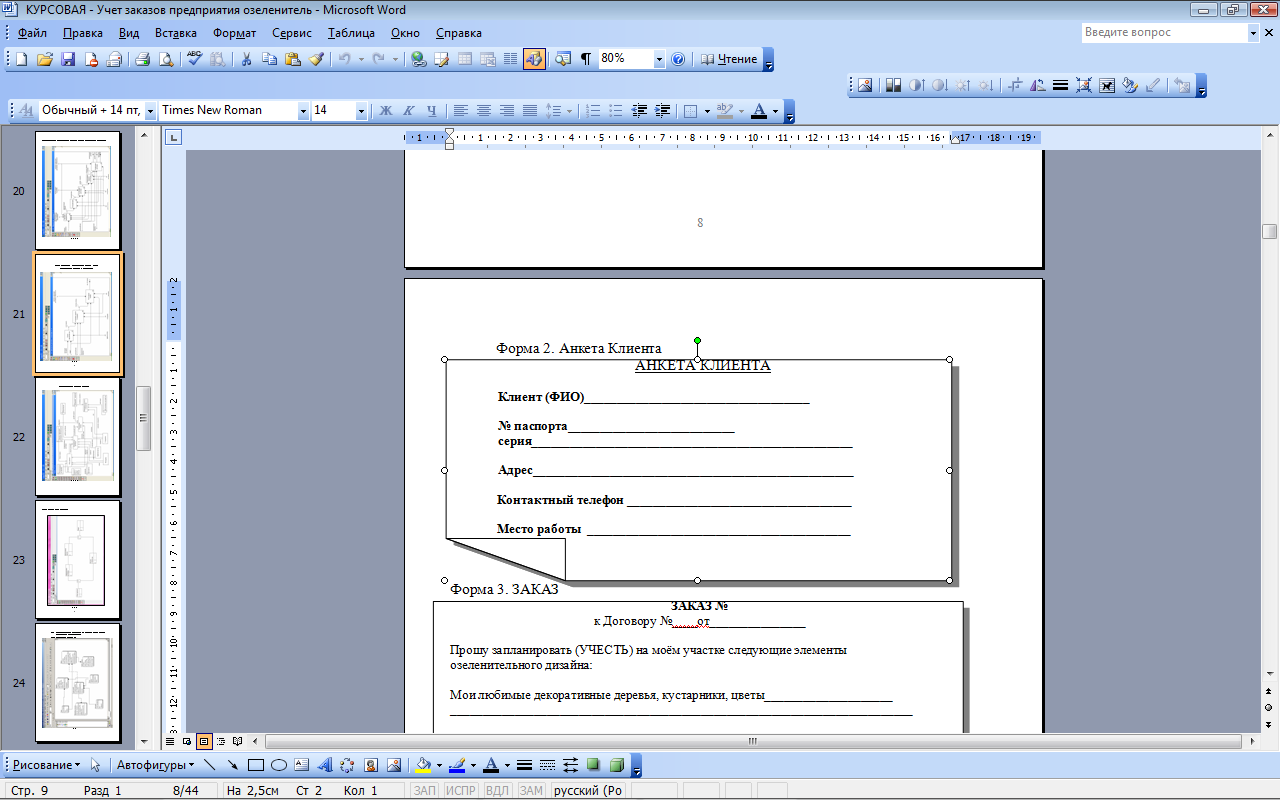
Данные о сотрудниках помещаются в Анкету сотрудников, которые содержат:

* Название фирмы, в которой сотрудник работает;
* Должность;
* ФИО сотрудника;
* Дата рождения;
* Домашний адрес;
* фото.



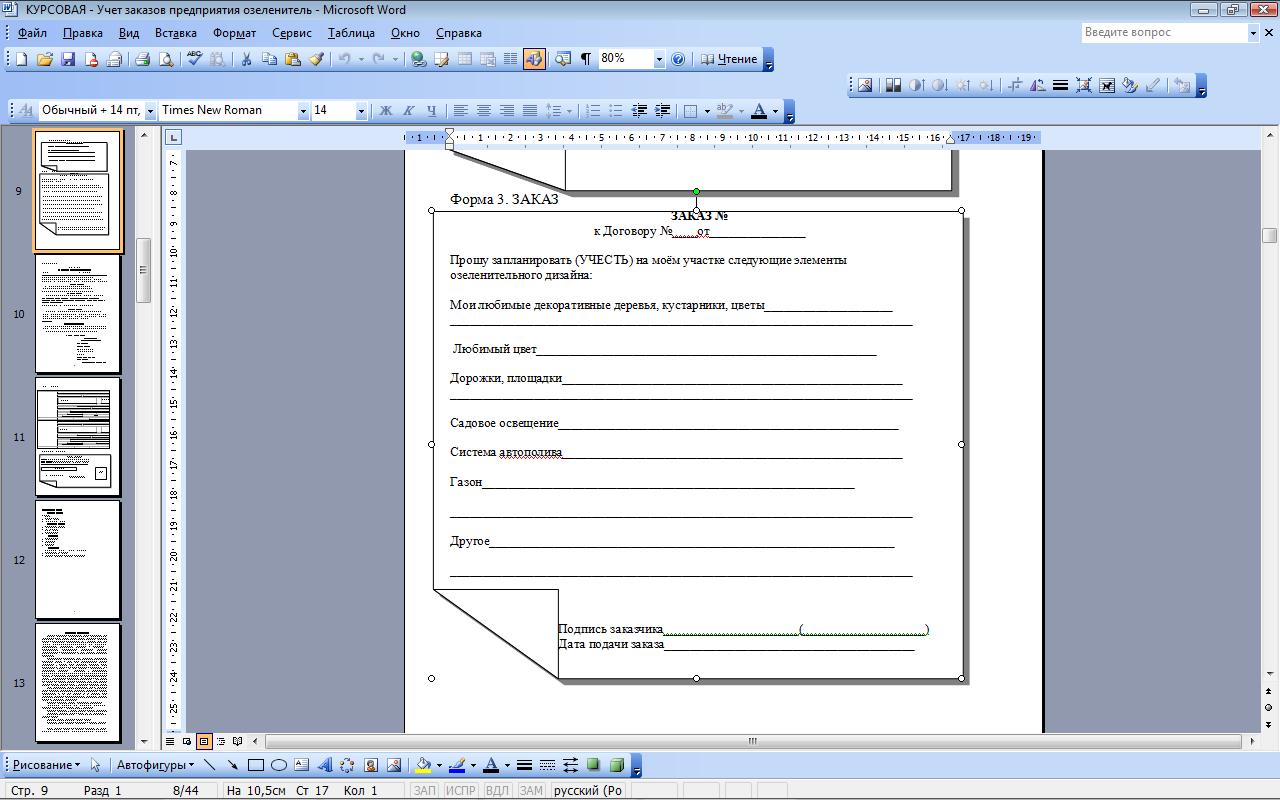
Данные о клиентах помещаются в Анкете клиента, содержащие:

* Клиент ФИО
* № паспорта, серия;
* Адрес;
* Контактный телефон;
* Место работы.



Данные о заказах помещаются в Анкете заказ, содержащей:

* № Договора;
* № заказа
* Пожелание клиента по дизайн – проекту.



Договор, подтверждающий заказ клиента:

Данные о заказах помещаются в Анкете Договор, содержащий:

* № Договора;
* Дата выдачи;
* Название Фирмы:
* Руководитель;
* Предмет договора;
* Условия договора и т.д.

Д О Г О В О Р

на выполнение озеленительных работ (услуг)

г.Армавир от «\_\_» \_\_\_\_\_\_\_\_20­­\_\_\_ г.

именуемый (ая, ое) в дальнейшем Заказчик, в лице \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

с одной стороны, и Муниципальное предприятие «Озеленитель», именуемое в дальнейшем Исполнитель, в лице Генерального директора АМБАРЯН А.А., с другой стороны, действующей на основании Устава, заключили настоящий Договор о нижеследующем:

1. **Предмет договора**

1.1**.** Исполнитель осуществляет \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

в дальнейшем именуется Объект, расположенный по адресу \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**II. Условия договора**

2.1 Заказчик производит оплату Исполнителю поэтапно в соответствии с Приложениями в размере \_\_\_\_\_ от суммы, указанной в каждом Приложении.

**\*За посадочный материал (растения) и материалы (плодородный грунт, песок…)**

**предоплата 100%. Условия оплаты за работы оговариваются в каждом случае отдельно (в основном поэтапная оплата).**

2.2 Сроком начала производства работ считается день поступления оплаты Исполнителю. Срок окончания производства работ указывается отдельно в каждом Приложении.

III. **Особые условия**

3.1 Исполнитель обязан подобрать Заказчику кондиционный посадочный материал в соответствии с параметрами, указанными в Приложениях. При соответствии параметров и состоянии кондиции отказ Заказчика от закупленных (подобранных) и доставленных ему растений возможен лишь до посадки в случае наличия у Заказчика Акта квалифицированной экспертизы о несоответствии растений (экспертиза проводится за счет Заказчика).

\* **Данный пункт действует, когда Заказчик предварительно посадочный материал не осматривает.**

**IV. Дополнительные условия**

4.1.Стороны освобождаются от ответственности по настоящему Договору вследствие действия обстоятельств непреодолимой силы (форс-мажор) в рамках действия данных обстоятельств, что подтверждается документально.

4.2.Настоящий Договор не может быть расторгнут без согласия обеих сторон.

У. Адреса и подписи сторон

ЗАКАЗЧИК

ИСПОЛНИТЕЛЬ

**Муниципальное предприятие**

**«Озеленитель»**

**ИНН 7712023040**

**ОГРН 1027700515347**

**125414, г. Армавир**

**ул.Мира, д. 16, оф. 34**

**р/с** 40702810300010013005

к/с 30101810000000000676

в ОАО АРКБ «Росбизнесбанк»

г.Армавир

БИК 044552676

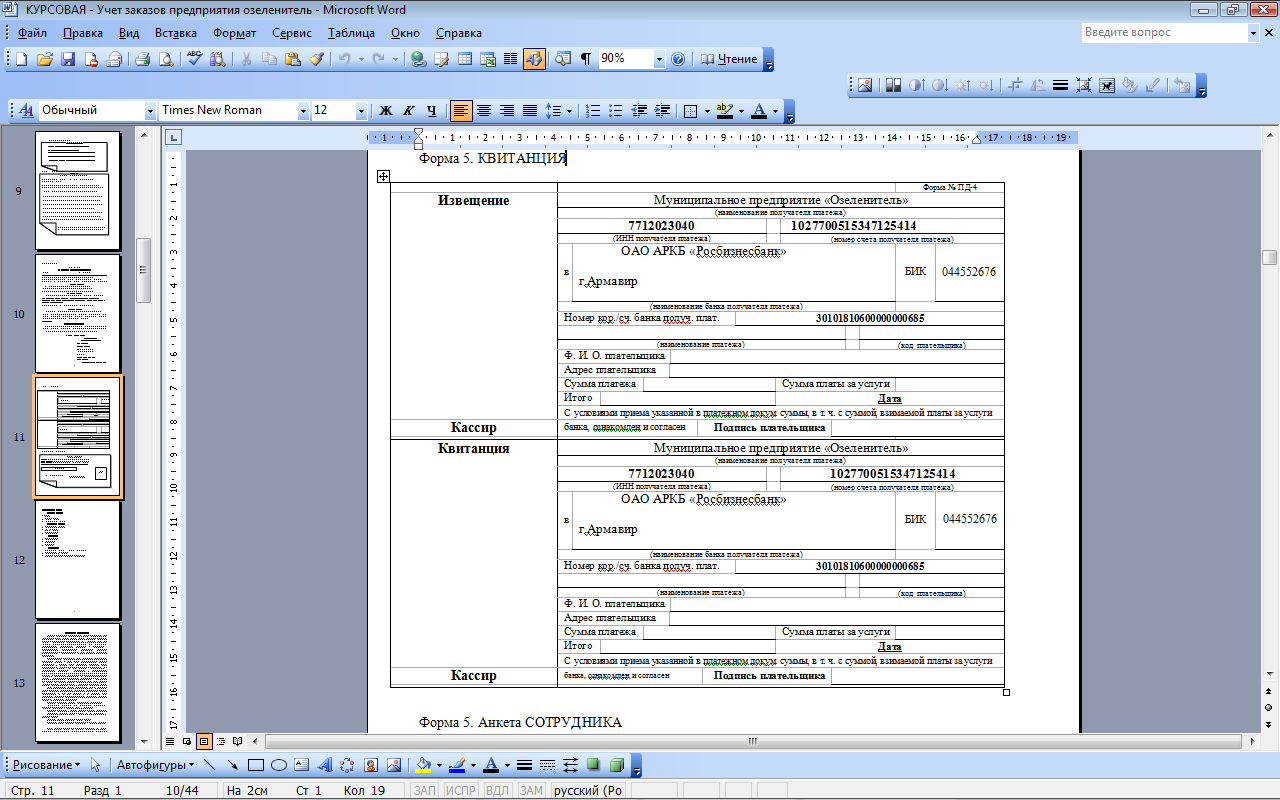
**ОКВЭД 01.41.2; 93.05; 74.84**

**ОКПО 17257803**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Амбарян А.А.)**

Данные о квитанции помещаются в Анкету квитанции, которые содержат:

* наименование получателя платежа;
* ИНН получателя платежа;
* наименование банка получателя платежа;
* Ф. И. О. плательщика;
* Адрес плательщика;
* Сумма платежа и т.д.



* 1. **Диаграмма потоков данных DFD**

Диаграммы потоков данных (Data Flow Diagrams — DFD) представляют собой иерархию функциональных процессов, связанных потоками данных. Цель такого представления — продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.

***Состав диаграмм потоков данных.***

Основными компонентами диаграмм потоков данных являются:

• внешние сущности;

• системы и подсистемы;

• процессы;

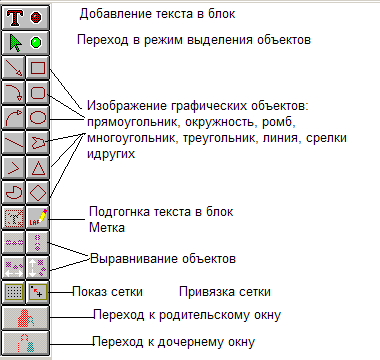
• накопители данных;

• потоки данных.

Внешняя сущность представляет собой материальный объект или физическое лицо, являющиеся источником или приемником информации, например, заказчики, персонал, поставщики, клиенты, склад. Определение некоторого объекта или системы в качестве внешней сущности указывает на то, что она находится за пределами границ анализируемой системы.

Процесс представляет собой преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом. Физически процесс может быть реализован различными способами: это может быть подразделение организации (отдел), выполняющее обработку входных документов и выпуск отчетов, программа, аппаратно реализованное логическое устройство и т.д.

Накопитель данных — это абстрактное устройство для хранения информации, которую можно в любой момент поместить в накопитель и через некоторое время извлечь, причем способы помещения и извлечения могут быть любыми.

Поток данных определяет информацию, передаваемую через некоторое соединение от источника к приемнику. Реальный поток данных может быть информацией, передаваемой по кабелю между двумя устройствами, пересылаемыми по почте письмами, магнитными лентами или дискетами, переносимыми с одного компьютера на другой и т.д.

**Рис. 6 Инструменты для создания DFD-модели**

1. Обучение на факультете

- процессы (работы) преобразования входных потоков данных в выходные

- накопители данных (хранилища).

ГОС

- внешние сущности

Абитуриент

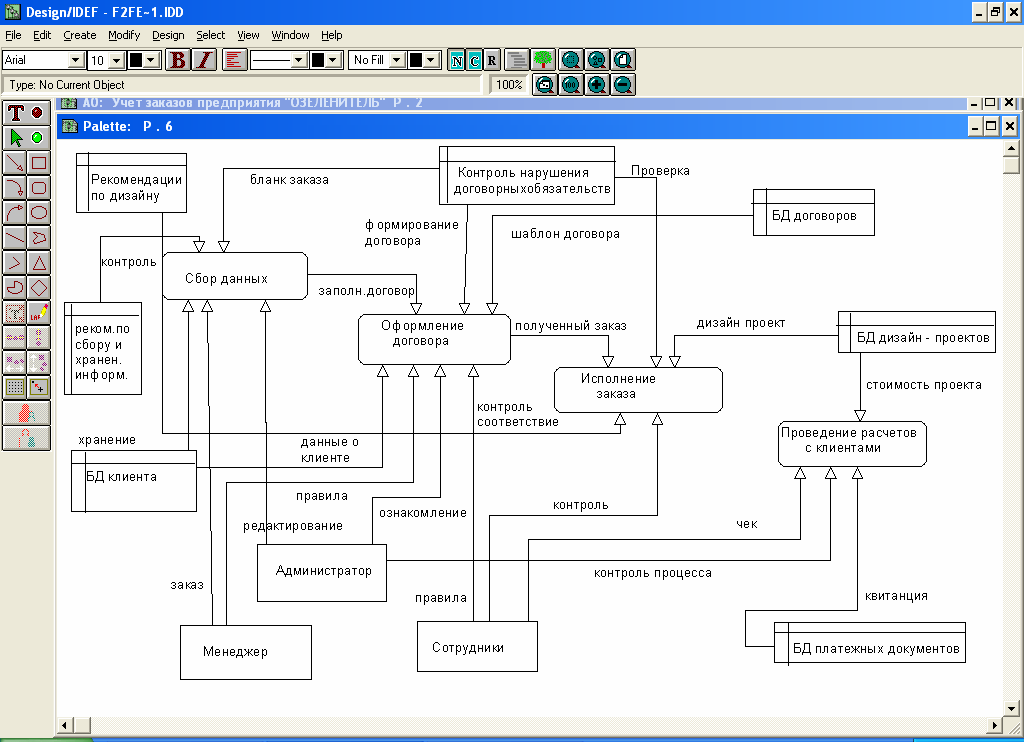
- потоки данных

1) Выбираем вкладку Create/New Page;

2) Появляется окно Select New Page Type, в Methodology выбираем Palette;

3) Используя кнопки панели инструментов, создаем:

* процессы (работы) преобразования входных потоков данных в выходные;
* накопители данных (хранилища).
* внешние сущности
* потоки данных



**Рис. 7 Диаграмма потоков данных (DFD)**

1. **Постановка задачи. Описание выходной информации (отчеты, справки, экранные формы). Описание входной информации (документы, справочники).**

**Цель работы** – создание ИС, позволяющей вести учет заказов предприятии «Озеленитель».

**Задачи ИС «Анализ пропуска занятий»:**

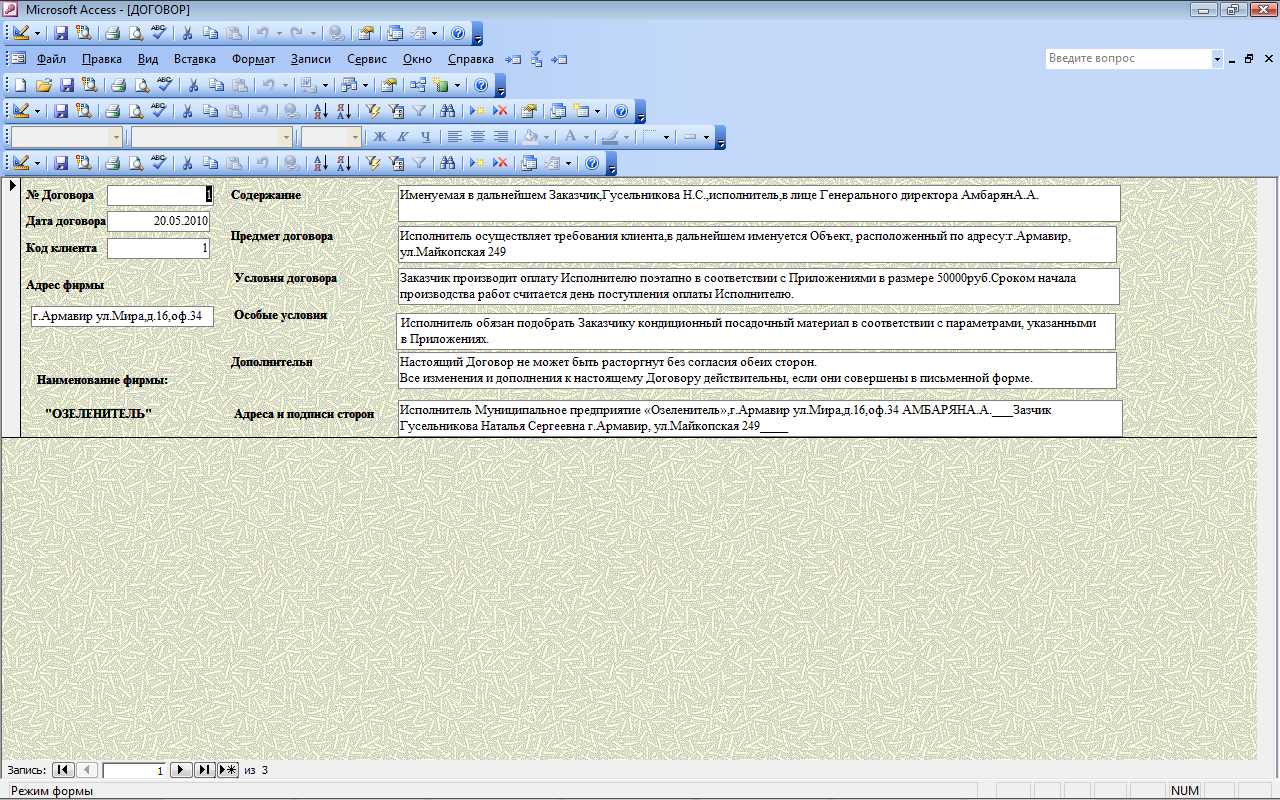
* Построение бизнес – процессов;
* Исследование документов и документооборота;
* Построение структуры БД, объектной модели в виде классов;
* Осуществление реализации в конкретной СУБД.

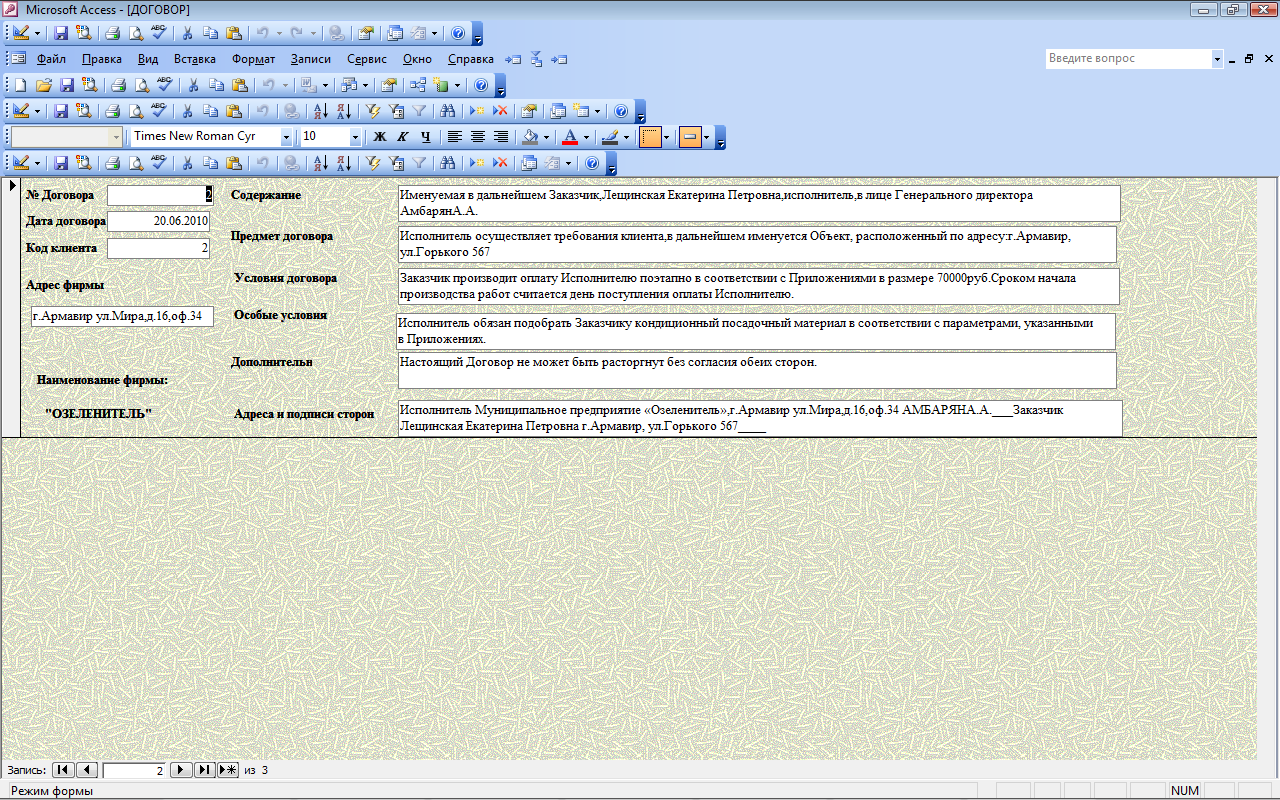
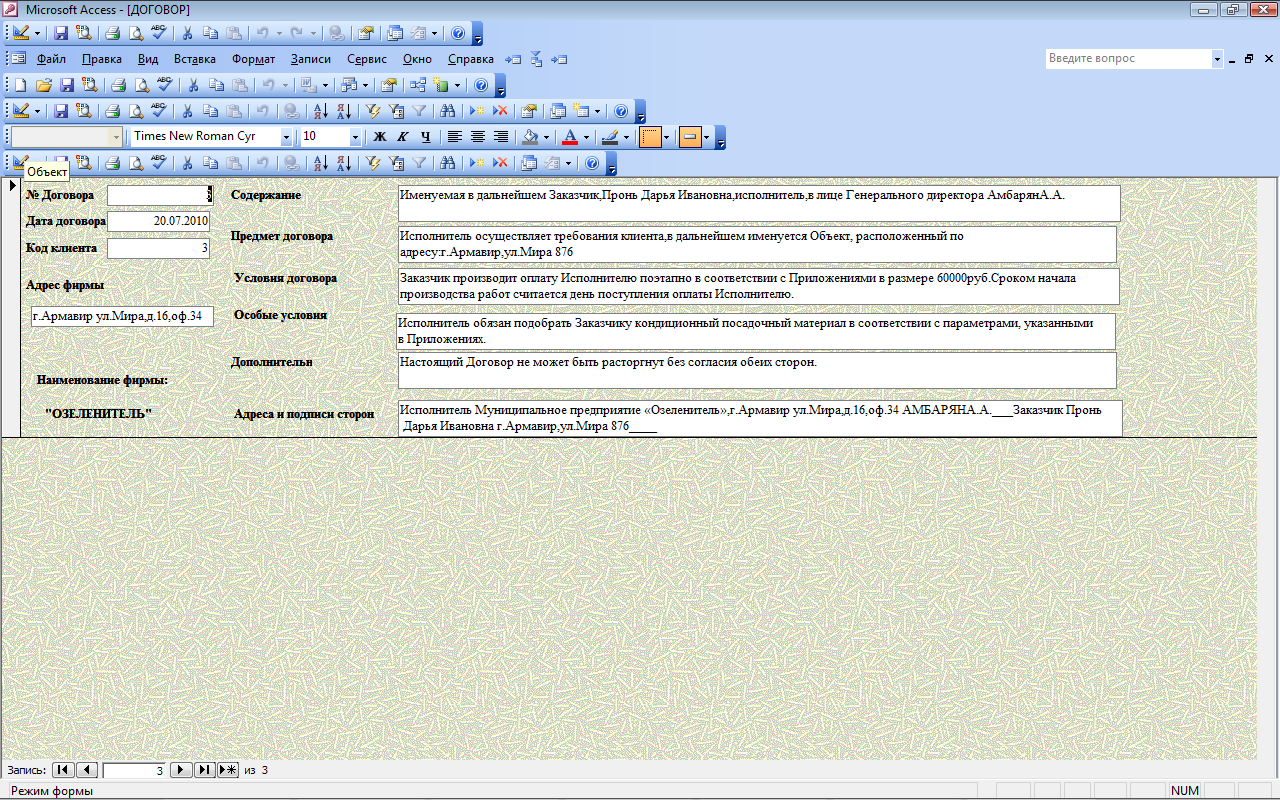
**Описание выходной информации (отчеты, справки, экранные формы)**

Здесь описываем выходную информацию задачи для пользователя. В результате решения задачи выходным документом будет Договор со списком клиентов. В нем содержится:

1. № Договора;
2. Дата Договора;
3. Код клиента;
4. Адрес фирмы;
5. Наименование фирмы;
6. Содержание фирмы;
7. Предмет договора;
8. Условия договора;
9. Особые условия;
10. Дополнительные условия;
11. Адреса и подписи сторон.

Теперь приведем формы выходных документов:

**Таблица 1. Список Договоров**

****

**Таблица 2. Описание выходных сообщений**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Название выходного сообщения | Идентификатор сообщения | Форма представления и требования к ней | Периодичность выдачи | Срок выдачи и допустимое время задержки | Кол-во экземпляров | Количество строк в документе (массиве) | Пользователи информации |
| Список договоров | договор | В виде документа, со всеми видами условий | По запросу | В течении 10 мин. Задержка до 2 часов | 2 | 500 | Руководитель,менеджер, клиент. |

**Таблица 3. Описание структурных единиц выходных сообщений**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название структурной единицы информации (поля) | Идентификатор | Тип данных | Длина | Идентификатор исходного сообщения, к которому принадлежит |
| договор | № договора | Числовой | 20 | Список договоров |

**4.1 Описание входной информации (документы, справочники)**

Здесь содержатся описание форм входных сообщений с информацией, на основе которой может быть решена задача и получены выходные документы (сообщения).

Здесь содержатся описание форм входных сообщений с информацией, на основе которой может быть решена задача и получены выходные документы (сообщения).

**Таблица 4. Справочник по клиентам**

|  |  |  |  |
| --- | --- | --- | --- |
| Код клиента | ФИО | Адрес | Контактный телефон |
| 1 | Гусельникова Наталья  Сергеевна | г.Армавир, ул.Майкопская 249 | 12547 |
| 2 | Лещинская Екатерина Петровна | г.Армавир, ул.Горького 567 | 65652 |
| 3 | Пронь  Дарья  Ивановна | г.Армавир,ул.Мира 876 | 12540 |

Оперативные данные, характеризующие клиентов, находятся в первичных документах. Таким документом является Анкета клиента. Для решения задачи данные с анкета клиента (Код клиента, ФИО, Адрес, Контактный телефон.

**Таблица 5. Перечень и описание входных сообщений**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название входного сообщения | Идентификатор | Форма сообщения и требования к ней | Срок и частота поступления | Источник поступления |
| Анкета клиента | Код клиента | В виде анкеты | По запросу | Руководства  фирмы |

**Таблица 6. Перечень и описание структурных единиц информации входных сообщений**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название структурной единицы | Идентификатор | Тип | Максимальная  разрядность в знаках | Вид контроля | Идентификатор сообщения, к которому принадлежит |
| Анкета  клиента | Код клиента | Числовой | 20 | Менеджер | Справочник по клиентам |

В этом же пункте определяется информация, выдаваемая по запросам пользователя. Для получения отчета с результатами принятых клиентов, необходимо обеспечить диалог с пользователем для ввода кода изделия. Такая входная информация вводится с помощью клавиатуры в ответ на сообщение-запрос на экране в процессе решения задачи.

**6. Диаграмма сущность-связь (ER-диаграмма)**

Для создания базы данных, структура которой не зависит от конкретных информационных потребностей и позволяет выполнить любые запросы пользователей, в индивидуальном задании следует разработать диаграмму инфологической модели «сущность-связь» (ER-диаграмма).

Общий подход к проблеме семантического моделирования характеризуется четырьмя основными этапами:

1. Прежде всего выявляется некоторое множество семантических концепций, понятий, называемых «сущностями» которые могут быть полезны при неформальном обсуждении реального мира. Сущности могут быть классифицированы по разным типам сущностей и тогда все сущности определенного типа обладают некоторыми общими свойствами. Каждая сущность обладает некоторым особым свойством, предназначенным для ее идентификации. Каждая сущность может быть связана с другими сущностями посредством некоторых связей.

2. Далее определяется набор соответствующих символических, формальных объектов, которые могут использоваться для представления описанных семантических концепций.

3. Определяется набор формальных общих правил целостности, предназначенных для работы с такими формальными объектами.

4. Также определяется набор формальных операторов, предназначенных для манипулирования формальными объектами.

Необходимо подчеркнуть, что правила целостности и операторы являются такой же частью модели данных, как и объекты.

Базовыми понятиями ER-диаграмм являются:

**Сущность** (Entity) — реальный или абстрактный объект, явление или процесс, имеющий существенное значение для рассматриваемой предметной области.

Каждая сущность должна обладать уникальным идентификатором (ами), благодаря чему каждый **экземпляр** сущности однозначно идентифицируется и отличается от всех других экземпляров данного типа сущности. Каждая сущность должна обладать свойствами:

* иметь уникальное имя;
* обладать одним или несколькими атрибутами, которые принадлежат сущности;
* обладать одним или несколькими идентифицирующими атрибутами, однозначно идентифицирующими каждый экземпляр сущности.

Сущности обладают некоторыми свойствами. Все сущности одного и того же типа обладают некоторыми общими свойствами. **Атрибут** — любая характеристика сущности, значимая для рассматриваемой предметной области и предназначенная для классификации, идентификации, классификации, количественной характеристики или выражения состояния сущности.

Каждая сущность может обладать любым количеством связей с другими сущностями модели.

**Связь** (Relationship)— поименованная ассоциация между двумя сущностями, значимая для рассматриваемой предметной области. Связь — это ассоциация между сущностями, при которой каждый экземпляр одной сущности ассоциирован с произвольным количеством экземпляров второй сущности, и наоборот. Тип связи рассматривается между типами сущностей, а конкретный экземпляр связи рассматриваемого типа существует между конкретными экземплярами рассматриваемых типов сущностей. Связь может быть обязательной, возможной, условной. Чаще всего встречаются связи между двумя типами сущностей (бинарные связи), хотя в модели могут быть выделены связи между любым количеством сущностей (тернарные, … n-арные).

**Мощность** связи представляет собой отношение количества экземпляров одной сущности к соответствующему количеству экземпляров другой сущности. Связи в модели «сущность-связь» могут иметь мощность:

**«один-к-одному»** (1:1), **«один-ко-многим»** (1:M) и **«многие-ко-многим»** (M:N).

Связь 1:1 (**«один-к-одному»**) – это связь между двумя типами сущностей A и B, при которой каждому экземпляру сущности A соответствует один и только один экземпляр сущности B.

Связь 1:M (**«один-ко-многим»**) – это связь между двумя типами сущностей A и B, при которой одному экземпляру сущности A может соответствовать ноль, один или несколько экземпляров сущности B, но каждому экземпляру сущности B соответствует только один экземпляр сущности A.

Связь M:N (**«многие-ко-многим»**) – это связь между двумя типами сущностей A и B, при которой каждому экземпляру сущности A может соответствовать ноль, один или несколько экземпляров сущности B и наоборот.

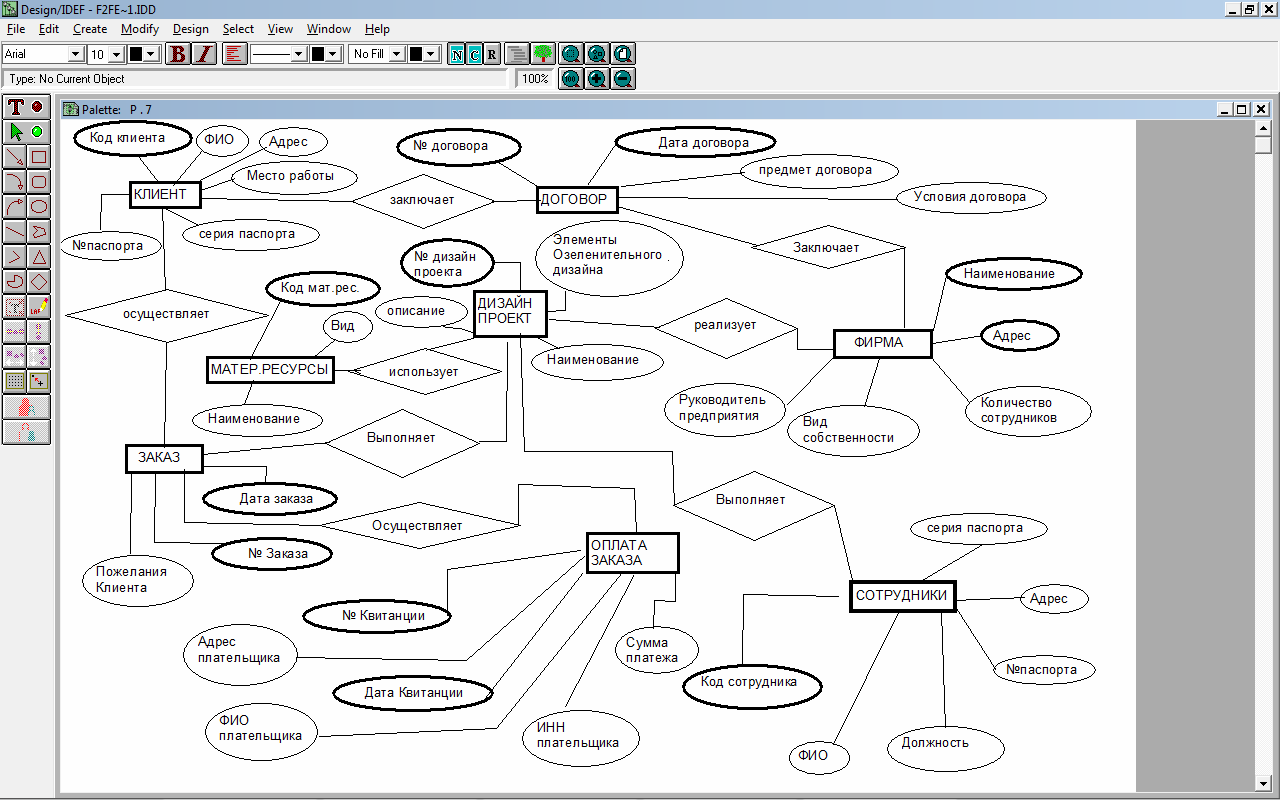
На первом этапе моделирования пытаются выделить множество семантических концепций, полезных для описания предметной области (части реального мира, имеющей интерес для проектировщика ИС). Некоторые из этих концепций, а именно – сущности, свойства, связи и подтипы представлены в Табл.7.

**Таблица 7**

|  |  |  |
| --- | --- | --- |
| Понятие | Неформальное определение | Примеры |
| СУЩНОСТЬ (Entity) | Некоторый отличимый объект, явление, процесс | Заказчик, изделие, поставка  Работник, отдел, человек  Заказ на поставку, серия заказов |
| СВОЙСТВО (Property) | Элемент данных, описывающий характеристику сущности | Номер поставщика, количество поставки  Отдел работника, оклад работника  Дата заказа |
| СВЯЗЬ (Relationship) | Связь, служащая для обеспечения взаимодействия между двумя или более другими сущностями | Поставка (поставщик-деталь)  Должность (работник–отдел) |
| ПОДТИП (Subtpye) | Сущность типа Y является подтипом сущности типа X тогда, когда каждый экземпляр сущности типа Y обязательно является экземпляром сущности типа X | «Работник» является подтипом сущности «Человек» |

Цель моделирования данных состоит в обеспечении разработчика ИС концептуальной схемой базы данных в форме одной или нескольких моделей, которые относительно легко могут быть отображены в любую систему баз данных. Наиболее распространенным средством моделирования данных являются модели "сущность-связь" (ER-модели). ER- модель включает аналоги всех семантических объектов, представленных в Табл.11.

Рассмотрим ER-диаграмму, представленной на рис. 8.

****

**Рис. 8 - ER-диаграмма**

**7. Разработка информационного обеспечения задачи. Определение информационных связей объектов и построение информационно-логической модели (ИЛМ) данных (модель IDEF1X).**

**7.1. Информационный анализ входной информации, необходимой для реализации задачи и выделения информационных объектов предметной области**

Рассмотрим информацию, которая помещается в документе Клиент.

В заголовочной части Клиент приводятся реквизиты (Код клиента, ФИО, Адрес, № анкеты и т.д.), которые являются общими для всего документа. Причем каждое из них имеет единое значение, поскольку один документ Анкета всегда привязан к одному Клиенту. Среди этих реквизитов особую роль играет код клиента. Поскольку код клиента уникальный, то есть его значение не может повторяться.

Табличная часть Анкеты клиента содержит реквизиты, которые имеют множество значений в соответствующих столбцах, поскольку Анкета может содержать несколько Клиентов.

На основе проведенного анализа устанавливают функциональные зависимости реквизитов документа Анкеты клиентов и отображают их в таблице 8.

**Таблица 8. Функциональные зависимости реквизитов Анкеты клиента**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Наименование реквизитов документа Анкета клиента | Имя реквизита | Функциональные зависимости | | | |
| № Анкеты | ANK |  | |  | |
| Код клиента | KK |  |  | |
| ФИО клиента | FIOKL |  |  | |
| Адрес клиента | AD |  |  | |
| Место работы | MR |  |  | |
| № паспорта | PAS |  |  | |
| Серия паспорта | SP |  |  | |
| Контактный телефон | KT |  |  | |
|  |  |  |  | |
|  |  |  |  | |

**Таблица 9.**  **Соответствие описательных и ключевых реквизитов**

|  |  |  |  |
| --- | --- | --- | --- |
| Описательные реквизиты | Ключевые реквизиты | Вид ключа | Название информационного объекта, который включает реквизит |
| 1 | 2 | 3 | 4 |
| KK  FIOKL  AD  MR  PAS  SP  KT | ANK  KK  KK  ANK  KK  KK  KK | первичный  первичный  первичный  первичный  первичный  первичный  первичный | Анкета  Код клиента  Код клиента  Анкета  Код клиента  Код клиента  Код клиента |

Для выделения информационных объектов необходимо сгруппировать реквизиты, которые имеют одинаковые ключи, в один информационный объект (ИO).

**Таблица 10. Реквизитный состав информационных объектов**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Реквизиты ИO | Признак ключа | | Имя ИO | Название ИO | Семантика (описание) |
| ANK | Первичный | | ANKETA | Анкета | Данные о анкете |
| KK  MR |  | |
| KK | Первичный | | KLIENT | Клиент | Общие сведения о клиенте |
| FIOKL  AD  PAS  SP  KT |  |  |

Составляется список информационных единиц (атрибутов или элементов данных) проектируемой ИС. Это необходимо для того, чтобы:

а) иметь единый перечень информационных единиц;

б) исключить синонимы;

в) определить важные, возможно, ключевые элементы данных;

г) определить атрибуты, реально отсутствующие в документах, но необходимые с точки зрения практического применения в базах данных;

д) определить спецификации элементов данных (тип, диапазон возможных значений, ограничения на принимаемые значения).

**7.2. Определение информационных связей объектов и построение ИЛМ данных**

DEF1X является методом для разработки реляционных баз данных и использует условный синтаксис, специально разработанный для удобного построения концептуальной схемы. Концептуальной схемой мы называем универсальное представление структуры данных в рамках коммерческого предприятия, независимое от конечной реализации базы данных и аппаратной платформы. Будучи статическим методом разработки, IDEF1X изначально не предназначен для динамического анализа по принципу "AS IS", тем не менее, он иногда применяется в этом качестве, как альтернатива методу IDEF1. Использование метода IDEF1X наиболее целесообразно для построения логической структуры базы данных после того, как все информационные ресурсы исследованы (скажем с помощью метода IDEF1) и решение о внедрении реляционной базы данных, как части корпоративной информационной системы, было принято. Однако не стоит забывать, что средства моделирования IDEF1X специально разработаны для построения реляционных информационных систем, и если существует необходимость проектирования другой системы, скажем объектно-ориентированной, то лучше избрать другие методы моделирования.



**Рис.9 Инструменты для создания IDEF1X-модели**

Для того, чтобы создать диаграмму **IDEF1X,** проделаем следующие действия:

1. Выберем вкладку Create/New Page

2. Создадим объекты-сущностей*.* Чтобы создать новую сущность в IDEF1X –модели.

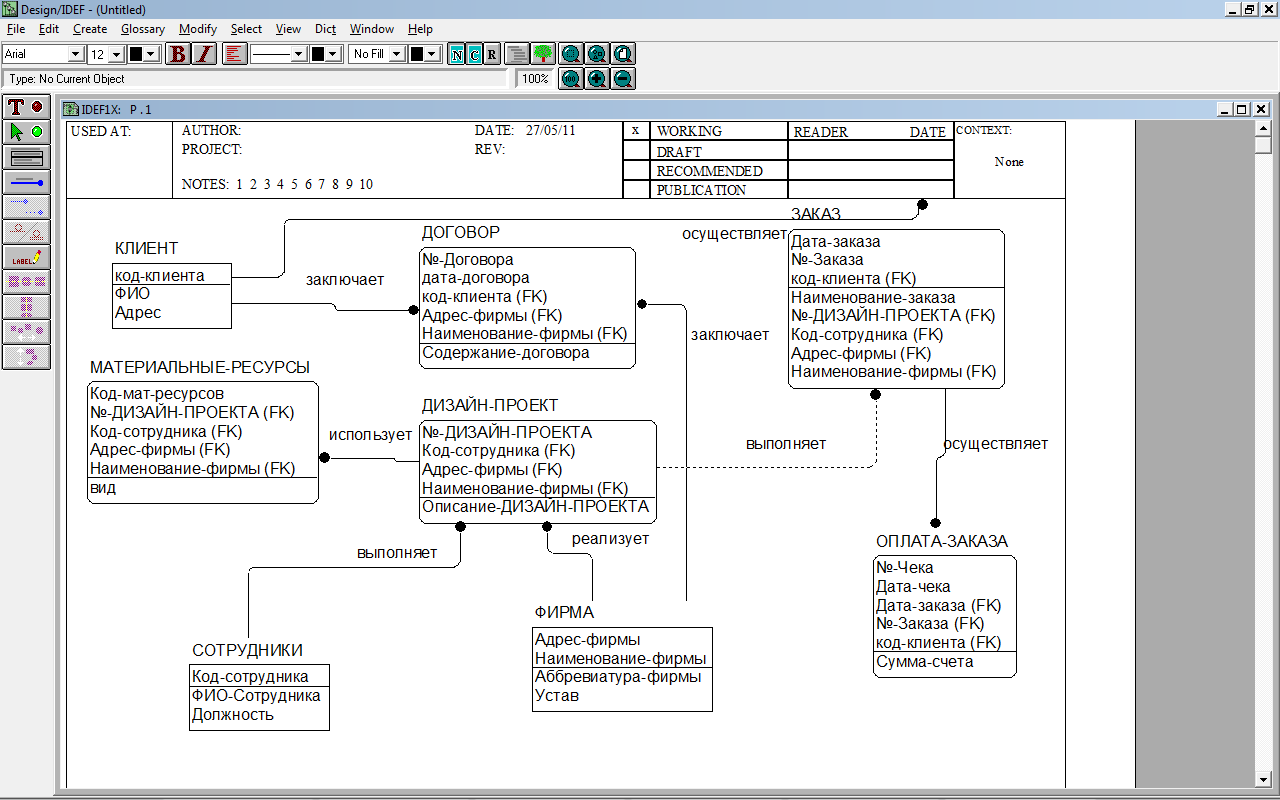
3. Выбираем Entityв меню Createили нажмите кнопку  на функциональной панели инструментов;

4. Обозначим с помощью мыши прямоугольный блок на странице диаграммы;



**Рис.10 Диалоговое окно «Define Entity»**

5. Заполните поля в появившемся окне «Define Entity» (рис. 6): Name–это уникальное имя, с помощью которого сущность будет распознаваться в модели; Aliases – псевдоним под которым сущность может быть известна; Definition– определение сущности, которое обычно используется в предметной области; Attributes– список атрибутов данной сущности. Чтобы ввести список атрибутов, нужно нажать на клавишу Add.



**Рис. 11 Модель IDEF1X**

Определение связей между информационными объектами и типа отношений, которыми они характеризуются, рассмотрим для предметной области «Учет заказов предприятия Озеленитель».

Связи между объектами Дизайн-проект и Заказ характеризуются отношениями один ко многим, поскольку Дизайн-проект имеет много Заказов.

Выполняя пункты 3,4,5, создаем остальные сущности.

**8. Нормализация полученных отношений**

Процесс преобразования базы данных к виду, отвечающему нормальным формам, называется **нормализацией**. Нормализация предназначена для приведения структуры базы данных к виду, обеспечивающему минимальную избыточность, то есть нормализация не имеет целью уменьшение или увеличение производительности работы или же уменьшение или увеличение объёма [БД](http://ru.wikipedia.org/wiki/%D0%91%D0%94). Конечной целью нормализации является уменьшение потенциальной противоречивости хранимой в [БД](http://ru.wikipedia.org/wiki/%D0%91%D0%94) информации.

Итак, проведем нормализацию с таблицей Договор:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № договора | Дата выдачи | Код клиента | ФИО клиента | Адрес клиента | Место работы | Предмет договора | Условия договора |
| 1 | 20.05.10 | 1 | Гусельникова | г.Армавир, ул.Майкопская 249 | Продавец-консультант в мазазине "Лавина" | осуществление требования клиента | производит оплату в размере 50000руб |
| 2 | 20.06.10 | 2 | Лещинская | г.Армавир, ул.Горького 567 | Безработная | осуществление требования клиента | производит оплату в размере 70000руб. |
| 3 | 20.07.10 | 3 | Пронь | г.Армавир,ул.Мира 876 | Безработная | осуществление требования клиента | производит оплату в размере 60000руб. |

Отношения(связи):

№ договора →Дата выдачи, Предмет договора, Условия договора

№ договора→→Код клиента

Код клиента→ФИО клиента

Код клиента→→Адрес клиента, Место работы

№ договора, Код клиента→Адрес клиента, Место работы, Предмет договора, Условия договора.

1. Приведем к **первой нормальной форме**

**Первая нормальная форма (1NF)** — базовая [нормальная форма](http://ru.wikipedia.org/wiki/%D0%9D%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%84%D0%BE%D1%80%D0%BC%D0%B0) [отношения](http://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BD%D0%BE%D1%88%D0%B5%D0%BD%D0%B8%D0%B5_%28%D1%80%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C%29) в [реляционной модели данных](http://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85).

Отношение находится в первой нормальной форме тогда и только тогда, когда в любом допустимом значении отношения каждый его [кортеж](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D1%82%D0%B5%D0%B6) содержит только одно значение для каждого из атрибутов.

В данном случае наша таблица Договор уже приведена к **1NF.**

1. Далее переходим ко второй нормальной форме (2NF)

Отношение находится во второй нормальной форме, если оно находится в [первой нормальной форме](http://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B2%D0%B0%D1%8F_%D0%BD%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%84%D0%BE%D1%80%D0%BC%D0%B0), и при этом любой его атрибут, не входящий в состав [потенциального ключа](http://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D0%B5%D0%BD%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BB%D1%8E%D1%87), [функционально полно](http://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%B7%D0%B0%D0%B2%D0%B8%D1%81%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D1%8C_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29) зависит от каждого потенциального ключа.

Учитывая все отношения, приведенные выше, выбираем первичные ключи: № договора и Код клиента, т. к. все кортежи так или иначе зависят именно от них.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № договора | Дата выдачи | Код клиента | ФИО клиента | Адрес клиента | Место работы | Предмет договора | Условия договора |
| 1 | 20.05.10 | 1 | Гусельникова | г.Армавир, ул.Майкопская 249 | Продавец-консультант в мазазине "Лавина" | осуществление требования клиента | производит оплату в размере 50000руб |
| 2 | 20.06.10 | 2 | Лещинская | г.Армавир, ул.Горького 567 | Безработная | осуществление требования клиента | производит оплату в размере 70000руб. |
| 3 | 20.07.10 | 3 | Пронь | г.Армавир,ул.Мира 876 | Безработная | осуществление требования клиента | производит оплату в размере 60000руб. |

Атрибуты Дата выдачи, Предмет договора, Условия договора и ФИО клиента зависят от части ключа № договора и Кода клиента, поэтому отношения нужно декомпозировать (разбить). В результате этого получаются две таблицы:

|  |  |  |  |
| --- | --- | --- | --- |
| № договора | Дата выдачи | Предмет договора | Условия договора |
| 1 | 20.05.10 | осуществление требования клиента | производит оплату в размере 50000руб |
| 2 | 20.06.10 | осуществление требования клиента | производит оплату в размере 70000руб. |
| 3 | 20.07.10 | осуществление требования клиента | производит оплату в размере 60000руб. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № договора | Код клиента | ФИО клиента | Предмет договора | Условия договора |
| 1 | 1 | Гусельникова | осуществление требования клиента | производит оплату в размере 50000руб |
| 2 | 2 | Лещинская | осуществление требования клиента | производит оплату в размере 70000руб. |
| 3 | 3 | Пронь | осуществление требования клиента | производит оплату в размере 60000руб. |

Теперь здесь все атрибуты, включая ФИО клиента, полно зависят то всего ключа.

### Переходим к третьей нормальной форме (3NF)

Согласно определению Кодда, таблица находится в 3NF тогда и только тогда, когда выполняются следующие условия:

* Отношение находится во [**второй нормальной форме**](http://ru.wikipedia.org/wiki/%D0%92%D1%82%D0%BE%D1%80%D0%B0%D1%8F_%D0%BD%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%84%D0%BE%D1%80%D0%BC%D0%B0);
* Каждый неключевой атрибут отношения находится в нетранзитивной (то есть прямой) зависимости от потенциального ключа.

В результате приведения к 3НФ получаются две таблицы:

Договоры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № договора | Код клиента | ФИО клиента | Предмет договора | Условия договора |
| 1 | 1 | Гусельникова | осуществление требования клиента | производит оплату в размере 50000руб |
| 2 | 2 | Лещинская | осуществление требования клиента | производит оплату в размере 70000руб. |
| 3 | 3 | Пронь | осуществление требования клиента | производит оплату в размере 60000руб. |

Клиенты

|  |  |
| --- | --- |
| Код клиента | ФИО клиента |
| 1 | Гусельникова |
| 2 | Лещинская |
| 3 | Пронь |

**9. Язык UML. Диаграммы классов**

Диаграммы классов являются центральным звеном методологии объектно-ориентированного анализа и проектирования. Диаграмма класса показывает классы и их отношения, тем самым представляя логический аспект проекта. На стадии анализа диаграммы классов используются, чтобы выделить общие роли и обязанности объектов (сущностей), обеспечивающих требуемое поведение системы, на стадии проектирования – чтобы передать структуру классов, формирующих архитектуру системы. Каждый класс должен иметь имя. Имя каждого класса должно быть уникально в содержащем его проекте. Диаграмма классов определяет этапы объектов системы и различные статистические связи, которые существует между ними. Имеется два основных вида статистических связей:

– ассоциации (например, менеджер может вести несколько проектов);

– подтипы (работник является разновидностью личности).

На диаграммах классов также изображаются атрибуты классов, операции и ограничения, которые накладываются на связи между объектами.

Технологический процесс создания диаграммы классов:

1. Подготовка:

a. В навигаторе модели открываю Logical View.

b. Там же открываю Main.

c. Даю имя диаграмме классов.

i. В контекстном меню для Main выбраю команду Rename.

ii. Ввожу имя диаграммы классов.

2. Создание класса:

a. Нажимаю кнопку создания класса.

b. В окне диаграммы классов указываю место класса.

c. Щелчком вызываю изображение класса.

d. Ввожу имя класса:

i. не повторяющееся с именами субъектов диаграммы прецедентов

ii. Являющиеся субъектами, их необходимо привести в стандартный для класса вид командой Format/Stereotype Display.

3. Оформляю класс:

a. В контекстном меню класса выбираю команду New Attribute.

b. Ввожу имя атрибута.

c. Активизирую класс, щелкнуть по значку атрибута.

d. В списке выбраю требуемый значок атрибута:

public (default) protected private implemented

e. В контекстном меню класса выбраю команду New Operation.

f. Ввожу имя операции.

g. Повторяю п.п. 2e, iii, iv для операции.

4. Создаю ассоциации:

a. Нажимаю кнопку создания ассоциации.

b. Рисую стрелку от одного класса к другому.

c. Отрегулирую размещение классов в диаграмме.

5. Оформляю ассоциацию:

a. В контекстном меню ассоциации выбираю команду Multiplicity.

b. В списке выбираю требуемый вид ассоциации

1 – обязательная однозначная;

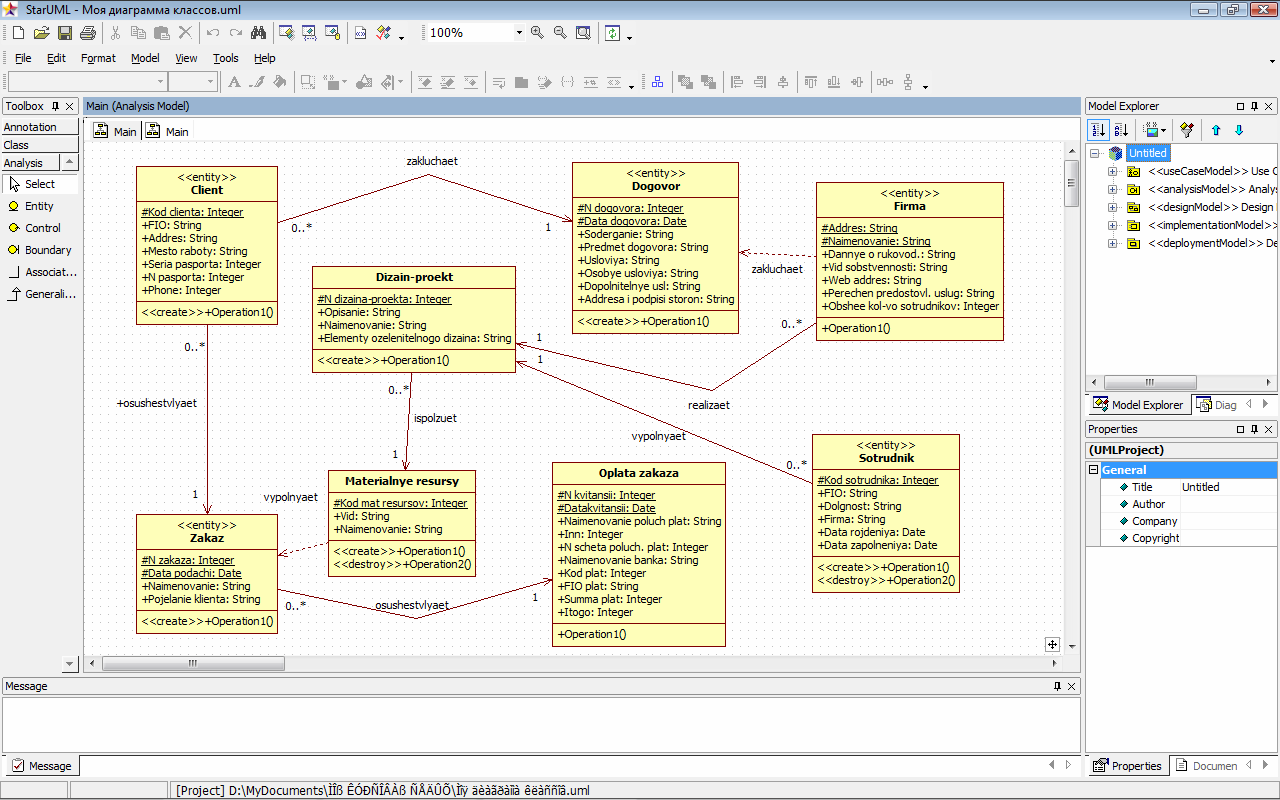
0 .. \* – Zero or More, необязательная многозначная;

1 .. \* – One or More, обязательная многозначная;

0 .. 1 – Zero or One, необязательная однозначная;

c. В контекстном меню ассоциации выбираю команду Navigable, убрав "галочку".

В результате получаем диаграмму классов, представленную на **рис. 12**:



**Рис. 12 Диаграмма классов**

1. **Определение логической и физической структуры базы данных**

**10.1 Логическое проектирование БД**

Логическая структура реляционной СУБД является адекватным отображением полученной информационно – логической модели, не требующим дополнительных преобразований. Каждый информационный объект модели данных отображается соответствующей реляционной таблицей. Структура реляционной таблицы определяется реквизитным составом соответствующего информационного объекта. Ключевые реквизиты объекта образуют уникальный ключ реляционной таблицы. Для каждого столбца задается формат и размер данных. Строки (записи) таблицы соответствуют экземплярам объекта и формируются при загрузке таблицы.

Связи между объектами модели данных реализуются одинаковыми реквизитами – ключами связи в соответствующих таблицах. При этом ключом связи всегда является уникальный ключ главной таблицы. Ключом связи в подчиненной таблице является либо некоторая часть уникального ключа в ней, либо поле, не входящее в состав первичного ключа Ключ связи в подчиненной таблице называют внешним ключом*.*

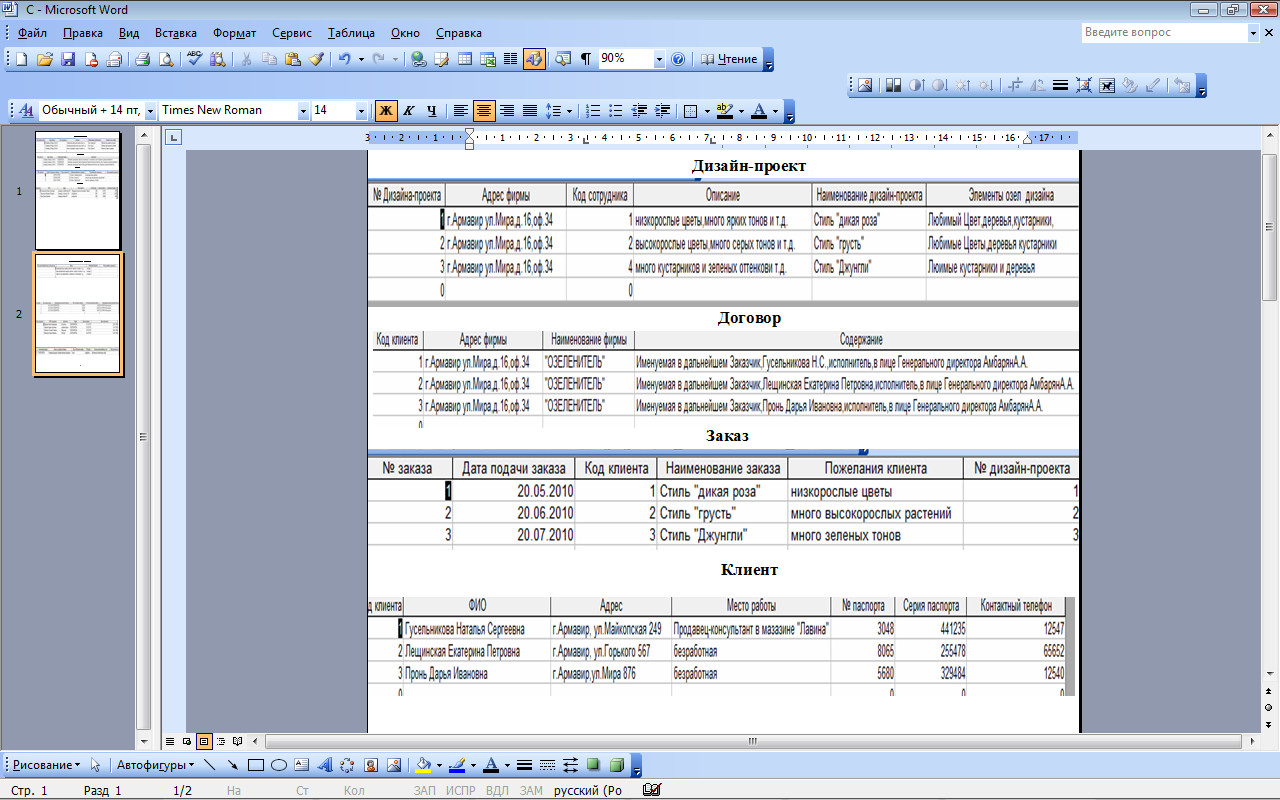
В СУБД может быть создана схема данных, наглядно отображающая структуру базы данных. Определение одно – многозначных связей в этой схеме должно осуществляться в соответствии с построенной моделью данных. Внешний вид схемы данных практически совпадает с графическим представлением информационно – логической модели.

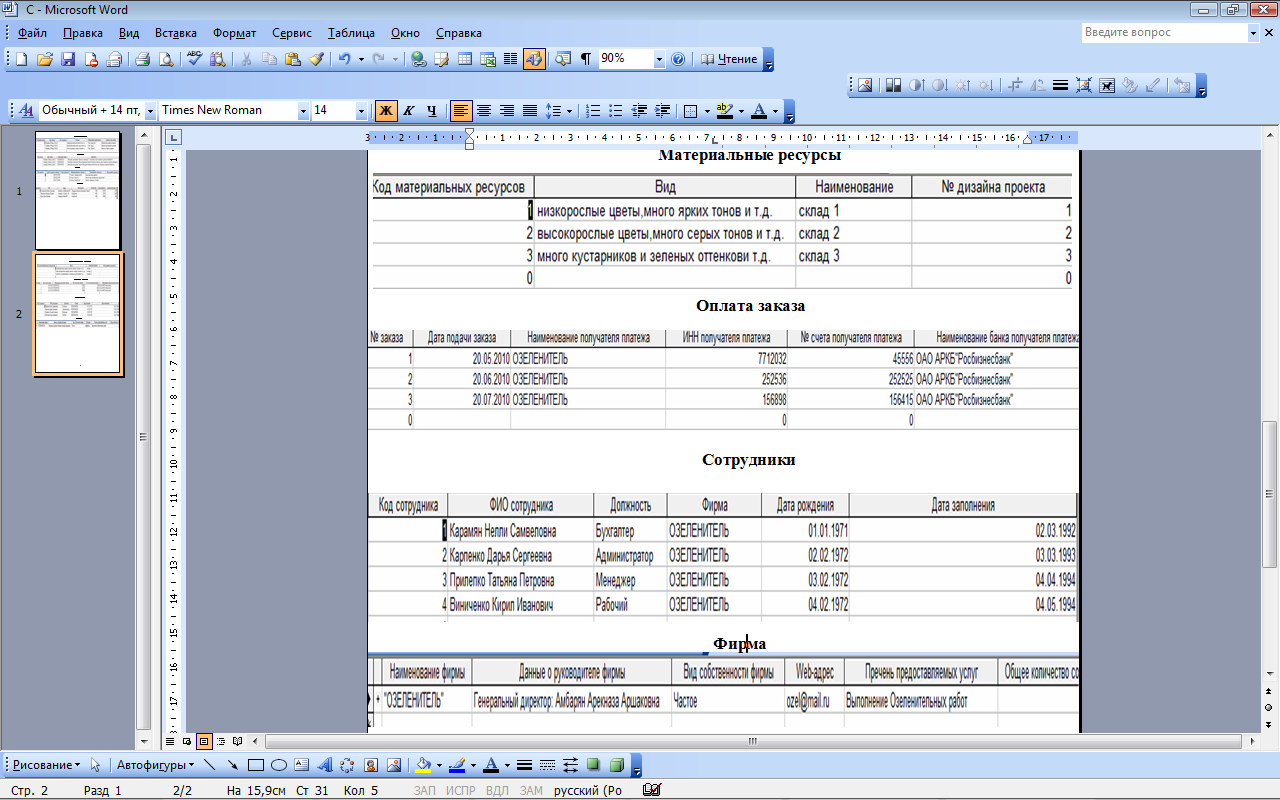
Используем ER-диаграмму для формирования набора предварительных отношений с указанием предполагаемого первичного ключа для каждого отношения. Правила формирования отношений следующие.

1) Каждая сущность преобразуется в отношение. Именем отношения является имя сущности, а набор элементов данных сущности становится набором атрибутов отношения. Идентифицирующие атрибуты сущности становятся первичным ключом отношения.

2) Особенность реляционной модели данных состоит в том, что связь между сущностями формирует также отношение. При этом именуется отношение тем же именем, что и связь ER-диаграммы. Атрибутный состав отношения определяется, как правило, набором идентифицирующих атрибутов каждой участвующей в связи сущности, а также собственными атрибутами связи. Ключ нового отношения может быть составным (набор ключей участвующих в связи сущностей), либо новым.

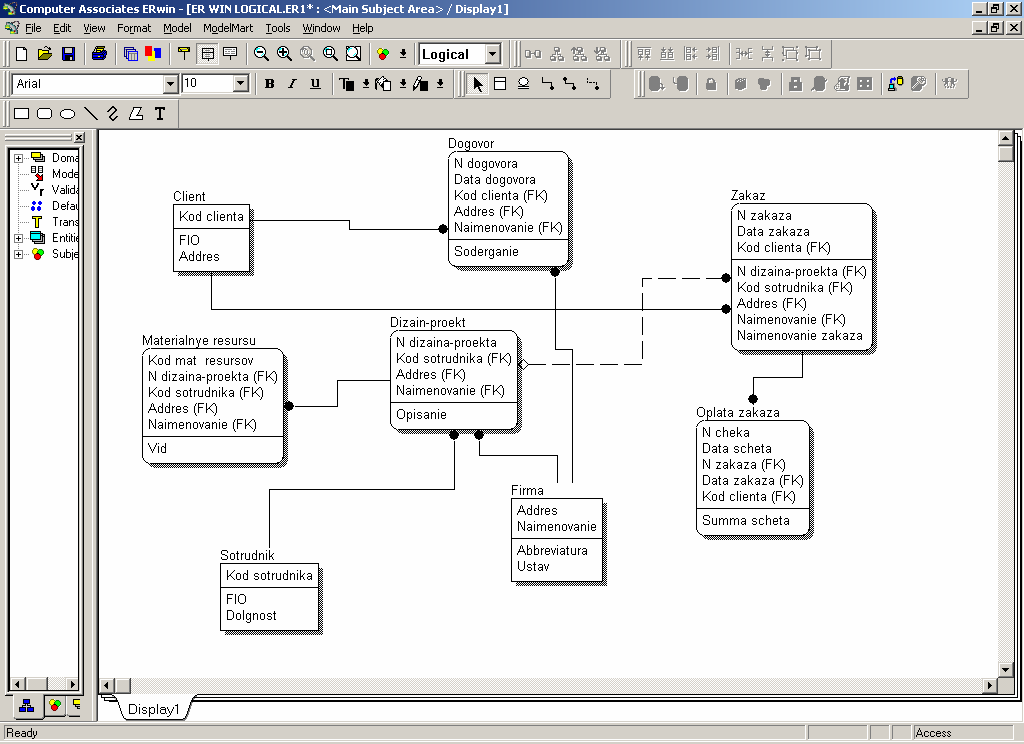
3) Связи между сильными и слабыми сущностями, а также связи подтип- супертип моделируются отношением, в котором атрибутный состав определяется набором ключевых атрибутов сильной сущности и всеми атрибутами слабой сущности. Именуется отношение также, как именуется слабая сущность и может имеет ключ – комбинацию ключевых атрибутов обеих сущностей.





1. Создаем в ERwin новую модель, задавая в окне Create Model-> New Model Type тип Logical/Physical.
2. Разрабатываем логическую модель уровня «Диаграмма сущность-связь», совместимую с функциональной моделью, описанной в Практикуме «Создание функциональной модели с помощью AllFusion PM». Описываем сущности, атрибуты, ключи. Специфицируйте связи, задавая их свойства, тип, мощность.

Логическая структура реляционной базы данных представлена на **рис. 13**



**Рис. 13 Логическая структура реляционной базы данных**

**10.2 Разработка физической структуры базы данных**

Физическое моделирование БД – это способ размещения информации на машинных носителях. Правила перехода от логической структуры к физической заключается в следующем:

- каждая реляционная таблица превращается в таблицу БД;

- каждый столбец таблицы – в поле таблицы;

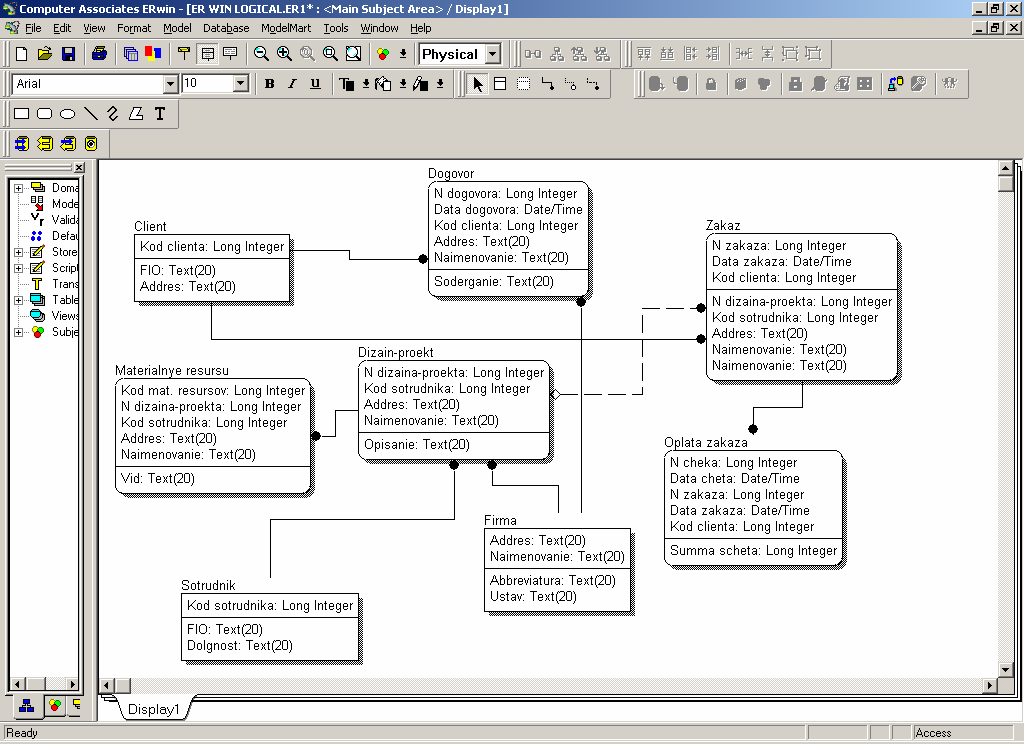
- каждая строка таблицы – в запись таблицы.

В процессе физического проектирования реляционной БД необходимо:

- присвоить имена таблицам;

- присвоить имена полям таблиц.

Таблицы базы данных заполняются на основе входных (первичных) документов.

Перейдите на уровень физической модели, выбрав необходимый сервер.

**Рис. 14 Физическая структура базы данных**

**Выводы**

Целью работы было создание моделей бизнеса ИС «Учет заказов предприятия «Озеленитель». Была исследована деятельность предприятия «Озеленитель» и достигнута поставленная цель.

В процессе проделанной работы, получен практический опыт по организации плана работ над проектом, ознакомлен с CASE - средствами, помогающими создать план работ по проекту, распределенный во времени.

Проводя анализ предметной области, а также предлагая подход к решению проблем предметной области, ознакомился с основными методами по созданию моделей методологией IDEF0 и выполнил следующие задачи:

1. Построение функциональной модели бизнес – процессов;
2. Исследование документов и документооборота;
3. Построение структуры БД, объектной модели в системы в виде классов;
4. Подготовка к реализации ИС в конкретной СУБД;
5. Описание проделанной работы.

**Основные аббревиатуры**

В данной работе использовались следующие термины с соответствующими определениями и сокращениями, установленные нормативными документами.

* Информационная система – система сбора, передачи, обработки и хранения информации.
* Проект – проектно - конструкторская и техническая документация, в которой представлено описание проектных решений по созданию и эксплуатации информационной системы в конкретной программно – технической среде.
* CASE - средства проектирования БД – это программные продукты, позволяющие автоматизировать, документировать, выполнить визуализацию процесса разработки проекта ПО (информационной системы или базы данных).
* БД − базы данных.
* СУБД – система управления базами данных.
* ИС – информационная система.

**Список использованных источников**

1. Вендров А.М. CASE – технологии. Современные методы и средства проектирования информационных систем. – М.: Финансы и статистика,1998.
2. Калянов Г.Н. CASE. Структурный системный анализ (автоматизация и применение). – М.: Лори, 1996.
3. Новоженов Ю.В. Объектно-ориентированные технологии разработки сложных программных систем. М., 1997
4. Панащук С.А. Разработка информационных систем с использованием CASE-системы Silverrun. "СУБД", 1997.
5. Горин С.В., Тандоев А.Ю. Применение CASE-средства Erwin 2.0 для информационного моделирования в системах обработки данных. "СУБД", 1999
6. Составление и оформление служебных документов. Практическое пособие для коммерческих фирм, общественных организаций и государственных структур./Под ред. проф. Т.В. Кузнецовой, изд. 2-е.
7. Григорьев Ю.А., Ревунков Г.И. Банки данных: Учеб. для вузов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002.- 320 с.
8. Орлов С.А., Технология разработки программного обеспечения: Учебник - СПб.: Питер, 2002.- 464с.:ил.
9. Маклаков С.В. BRwin и ERwin. CASE- средства разработки информационных систем. – М.: "ДИАЛОГ- МИФИ ",1999, 256с.
10. Презентации, конспекты и другие материалы по предмету «Моделирование бизнеса. CASE- средства проектирования БД».