

Структурная оптимизация. Методология SADT (IDEF0 и IDEF3)

Основы работы с графическим редактором MS Visio

Принцип программы – использование готовых образцов – **шаблонов**. Именно наличие обширной коллекции готовых образцов – галереи шаблонов **diagram gallery** позволяет легко работать в MS Visio.

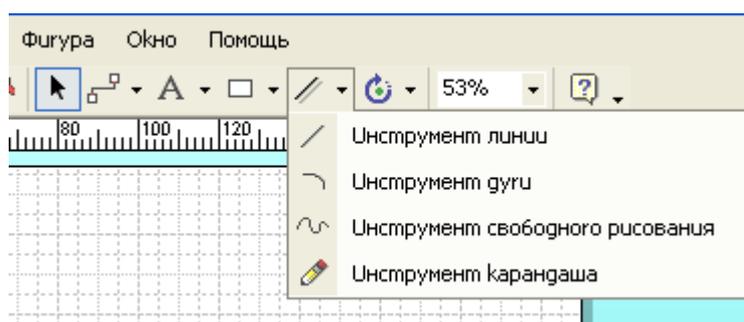
Благодаря использованию готовых шаблонов при создании рисунков, пользователь Visio освобождается от кропотливого труда рисования сложных фигур. Ему остается только найти необходимые шаблоны.

Окно шаблонов по умолчанию располагается слева от окна рисунков и имеет заголовок **Shapes**. Содержимое окна зависит от выбранных трафаретов, но обязательно в окне шаблонов присутствует строка заголовка и панель поиска шаблонов **Search for Shapes**.

Если вы не планируете использования шаблонов, то окно шаблонов можно закрыть, нажав на крестик в правом верхнем углу окна шаблонов.

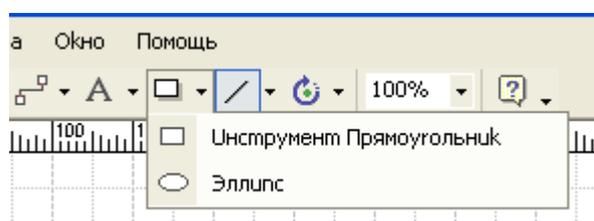
1.1. Простые рисунки на основе графических примитивов

Графические примитивы MS Visio, входящие в группу команд «Инструменты рисования», располагаются на панели Стандартная. К одномерным инструментам относят отрезок, дугу, кривую и карандаш.



Инструменты рисования для создания одномерных объектов

Инструменты для создания плоских (двумерных) примитивов.



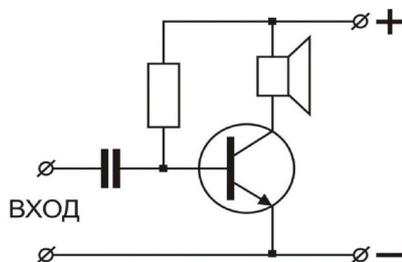
Инструменты рисования для создания плоских объектов

Если на имеющуюся плоскую фигуру положить новую плоскую фигуру, то она перекроет первую. Если вам нужны 2 или несколько одинаковых фигуры, то их можно скопировать. Для копирования можно воспользоваться стандартным способом через главное меню или контекстное меню. Но более удобным здесь будет способ копирования

при перетаскивании. Для этого, выберите команду «Инструмент указателя»  со стандартной панели инструментов. Указатель курсора мыши примет вид белой стрелки. Далее подведите курсор к тому объекту, копию которого вы хотите получить и одинарным щелчком выделите его. Далее, нажав на клавиатуре кнопку CTRL и, не отпуская ее, нажмите левую кнопку мыши и начинайте перетаскивать исходный объект. Вы увидите, что объект остается на месте, а его копия начинает перемещаться. Таким образом, можно быстро создать несколько одинаковых объектов. Если же нужно много одинаковых объектов, то после создания 3-4 копий, можно их выделить все и копировать теперь всю группу.

Практическое задание 1

Для начала построим простой рисунок, показанный на рис. 1.3. Жирные линии, обозначающие конденсатор и n-p-n переход транзистора можно построить как узкие прямоугольники, залитые черным цветом.



Практическое задание 1

1.2. Форматирование рисунков

Форматирование рисунка предполагает изменение его параметров, заданных средой по умолчанию. Форматирование можно выполнить с помощью команд главного меню (меню **Формат**) или команд контекстного меню. Редактирование объектов лучше выполнять в режиме «Инструмент указателя» - обычно в этом режиме меньше всего ошибок. В любом случае, при выполнении ошибочного действия можно воспользоваться командой отката. Откат можно выполнить с помощью команды **Редактирование – шаг назад** главного меню или соответствующей кнопки пиктографического меню .

В меню расположены команды копирования и перемещения объекта (вырезание, копирование, вставка), а также команды **Просмотр**, **Формат**, **Фигура**.

Рассмотрим команды, расположенные в группе **Формат**.

Команда **Текст** применяется при форматировании текста. При выборе команды **Линия...** раскрывается диалоговое окно с опциями.

Опция **Образец** позволяет выбрать тип линии – сплошная, пунктирная, штриховая, штрих-пунктирная. Вы можете попробовать изменить тип линии, посмотреть как изменился рисунок, а потом выполнить откат.

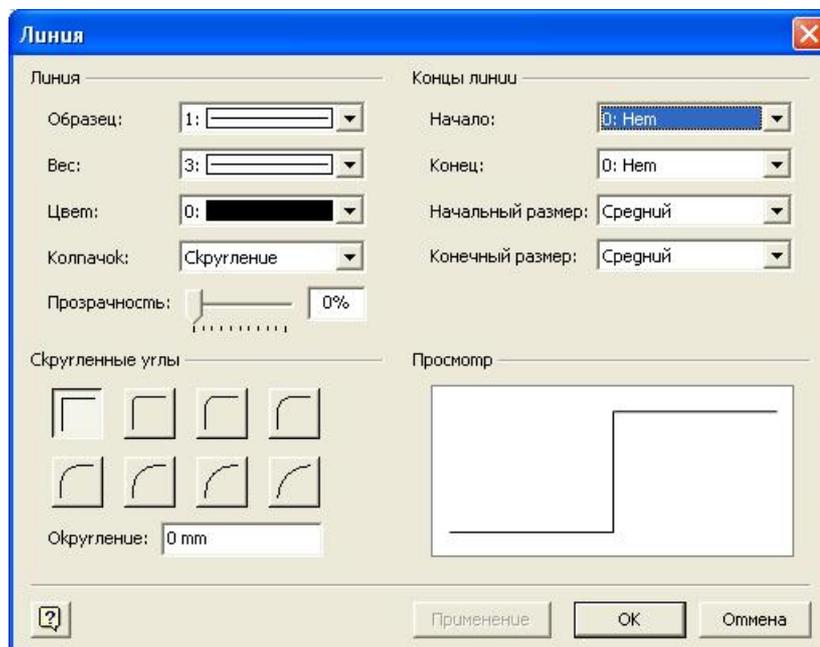
Опция **Вес** позволяет изменять толщину линии, которая задается в пунктах.

Опция **Цвет** позволяет выбрать цвет линии. Если среди 23 имеющихся цветов нет подходящего, выберите команду **Больше цветов**.

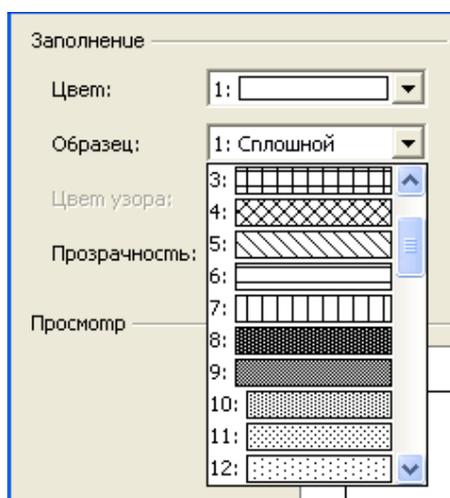
Опция **Прозрачность** позволяет сделать линию более бледной.

Скругленные углы – соответственно позволяет сделать углы прямоугольника более или менее скругленными.

Группа команд **Концы линии** позволяет добавить в конец линии стрелку нужной формы и изменить ее размер. Эти команды применяют для отрезков, когда строят различные схемы или диаграммы.



Диалоговое окно «Линия»



Выпадающий список для выбора стиля заполнения

Следующей группой в команде **Формат** контекстного меню является **Заполнение**.

При выборе опции открывается соответствующий диалог.

Цвет заполнения по умолчанию – белый. Вы можете изменить его на другой.

Образец позволяет изменить стиль заполнения путем выбора из выпадающего списка.

Прозрачность соответственно позволяет задать определенную степень прозрачности заполнения.

Подобные же операции можно выполнить и с тенью фигуры, если ее добавить.

Следующей после команды **Формат** являются команды группы **Фигура**.

Команда **Группа** позволяет сгруппировать несколько фигур. Например, схему из задания 1 можно увеличить или повернуть на несколько градусов, если предварительно сгруппировать все объекты. Можно сгруппировать команды, образующие один сложный объект, например, транзистор. Тогда его можно отдельно переместить или скопировать в другую схему.

Команда **Перемещение вперед** позволяет выдвинуть фигуру с заднего плана на передний. **Перемещение назад**, наоборот, задвигает фигуру на задний план.

Поворот влево и поворот вправо, соответственно, поворачивают фигуру на 90 градусов против или по направлению хода часовой стрелки.

Переворот по вертикали и переворот по горизонтали производят зеркальное отображение фигуры сверху-вниз или слева-направо.

1.3. Вставка текста

Чтобы добавить текст на лист рисунка, нужно выбрать команду **Инструмент текста**, расположенную на панели стандартная пиктографического меню. После ввода текста его можно редактировать, используя соответствующие команду, выбранные из главного или пиктографического меню.

1.4. Окно просмотра и позиции

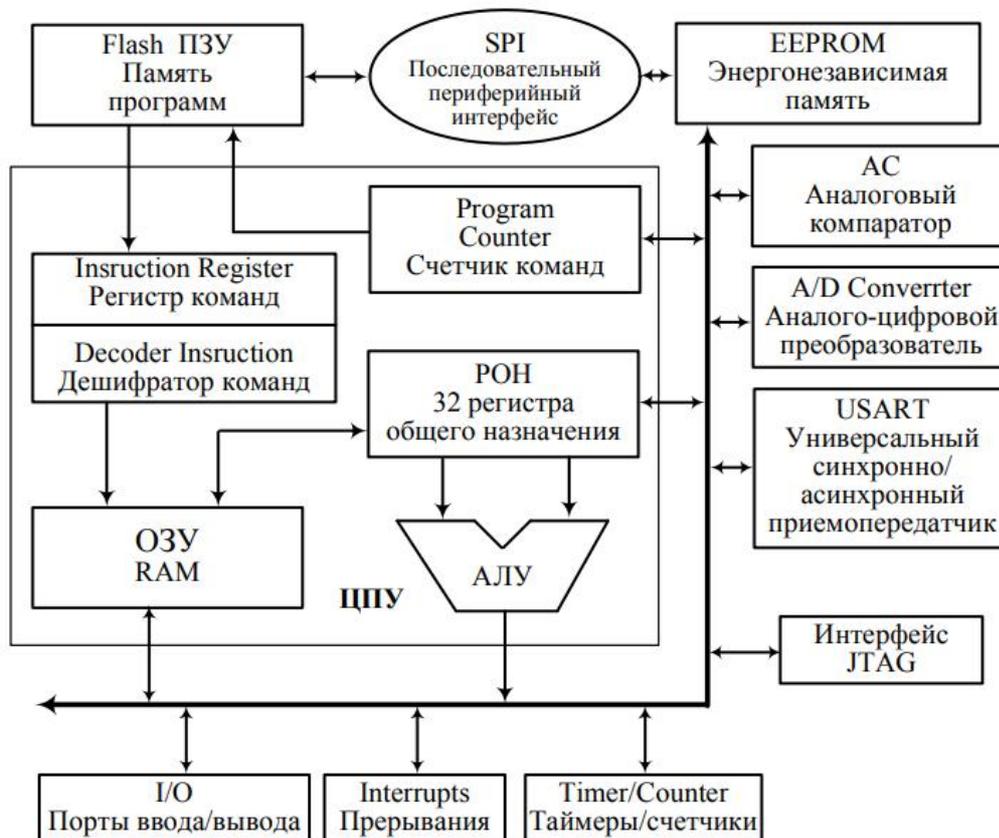
Начало X	100 mm	Позиция размер... ✕
Начало Y	120 mm	
Конец X	140 mm	
Конец Y	120 mm	
Длина	40 mm	
Угол	0 deg	
Высота	0 mm	

Если выполнить команду из главного меню **Просмотр – окно размера и позиции**, то откроется окно с координатами и размерами фигур. С помощью этого инструмента можно создавать и настраивать точные изображения с точностью до долей миллиметра и градуса.

Окно просмотра и позиции

Практическое задание 2

Постройте структурную схему микроконтроллера AVR.



Практическое задание 2

1.5. Рисование фигуры с несколькими сегментами

В простейшем случае каждая фигура состоит из одной части – сегмента. Чтобы нарисовать фигуру с более чем одним сегментом, начинайте каждый новый сегмент в одной из конечных точек или из угла предыдущего сегмента. Можно удалить нарисованный сегмент выбором команды Откат в меню Редактирование. Можно также поправить, добавить, и удалить сегмент после того, как вы дорисуете фигуру.

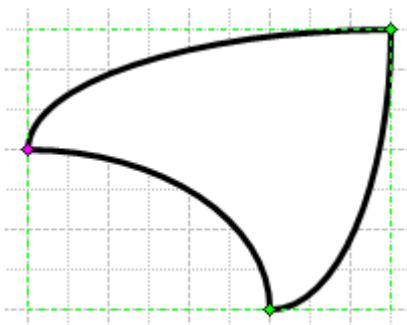
Чтобы нарисовать фигуру с несколькими сегментами:

1. В меню инструментов рисования выберите прямую или дугу.
2. Укажите точку, где вы хотите начать фигуру.
3. Тяните, чтобы нарисовать первый сегмент, затем отпустите кнопку мыши.
4. Укажите один из концов первого сегмента и тяните, чтобы нарисовать следующий сегмент. Когда закончите рисование, отпустите кнопку мыши.



Фигура, состоящая из двух сегментов – дуг

Будьте осторожны, чтобы не щелкнуть по конечной точке первого сегмента. Если вы выберете конечную точку, она станет малиновой и фигура замкнется и станет сплошной плоской фигурой, для которой можно использовать команды заполнения и заливки. Это можно заметить потому признаку, что сетка рисунка под новой фигурой скрывается. При перемещении конечной точки сегмента Visio изменяет размеры, вместо того, чтобы начать рисовать новый.



Малиновая конечная точка первого сегмента

Соединение и фрагментирование фигур

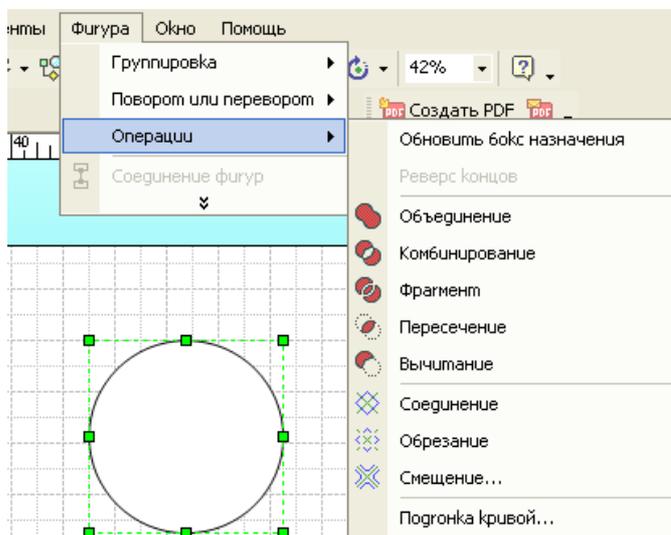
Можно создавать сложные фигуры, рисуя простые компоненты и используя затем команду **Операции**. Есть несколько видов операций.

Комбинирование – объединяет фигуры, чтобы создать фигуры с отверстиями в них типа рамки для картины.

Фрагмент – делит единую фигуру на меньшие части. Создает новую законченную фигуру там, где накладываются две или больше сложных фигур. Создает новую

законченную фигуру в области, ограниченной двумя или более линиями, дугами или кривыми.

Пересечение – создает новую фигуру из области пересечения двух или более фигур.



Вычитание – создает новую фигуру вычитая одну или несколько фигур из существующей фигуры.

Объединение – объединяет две или более фигуры, создавая новую фигуру по их периметрам.

При использовании команд новая фигура наследует формат и текст первой фигуры. Формат и тексты остальных фигур теряются.

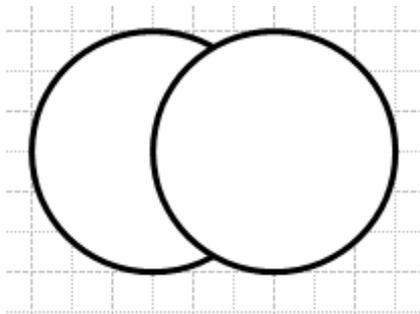
Команда Операции над фигурами

Пересечение фигур

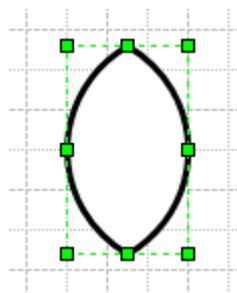
При пересечении двух или более фигур Visio создает описание для новой фигуры и удаляет описания первоначальных.

Для пересечения фигур:

1. Разместите пересекаемые фигуры на странице рисунка. Пусть это будут два одинаковых круга, полученные путем копирования.



Две фигуры, подготовленные для пересечения



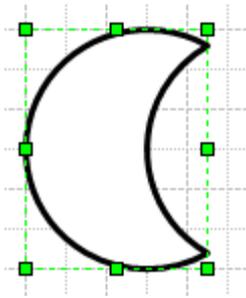
2. Выделите фигуры. Чтобы выделить вторую фигуру и при этом не снялось выделение первой фигуры, удерживайте кнопку Shift.

3. В меню Фигура выберите команду Операции, затем выберите Пересечение.

Результат пересечения двух фигур

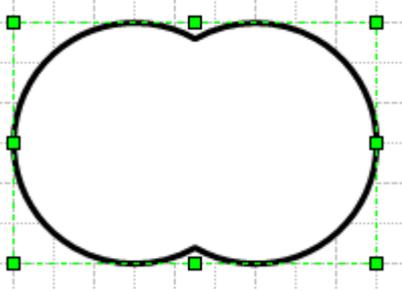
Вычитание фигур

Наложите одну фигуру на другую. Выделите первую, затем вторую фигуры.



Выберите команду Вычитание из меню Фигура – Операции.
Перекрывающийся сегмент второй фигуры “вычитается” из первой.

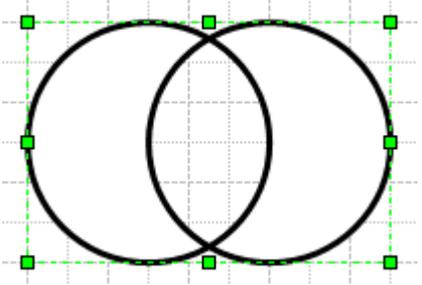
Результат вычитания двух фигур



Объединение двух фигур

Выполняется выбором команды Объединение.
Результат объединения для двух кругов.

Результат объединения двух фигур



Комбинирование фигур

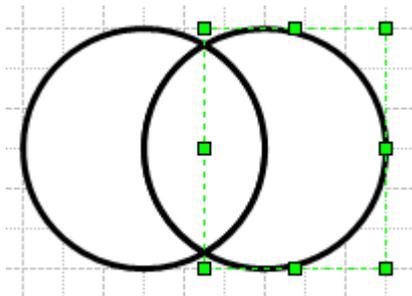
Выполняется выбором команды Комбинирование.
Результат комбинирования исходных кругов.

Обратите внимание на то, что в последнем случае область сплошной части фигуры получилась в форме рамки, а внутренняя часть стала представлять собой отверстие в единой фигуре.

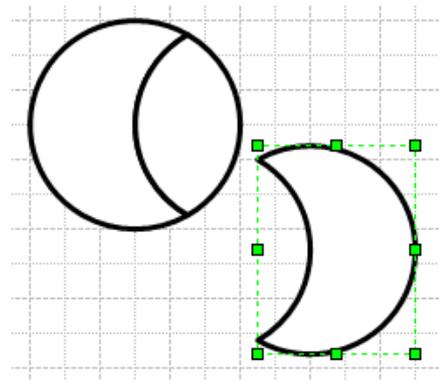
Результат комбинирования двух фигур

Фрагментирование фигур

Выполняется выбором команды Фрагмент. Если начать перемещать выделенный фрагмент вы увидите результат – два разных фрагмента.



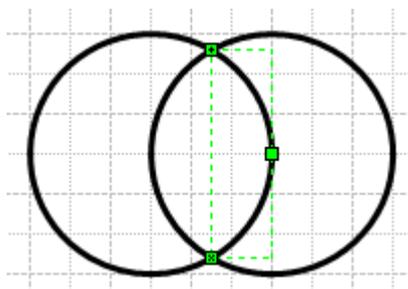
Результат фрагментирования двух фигур



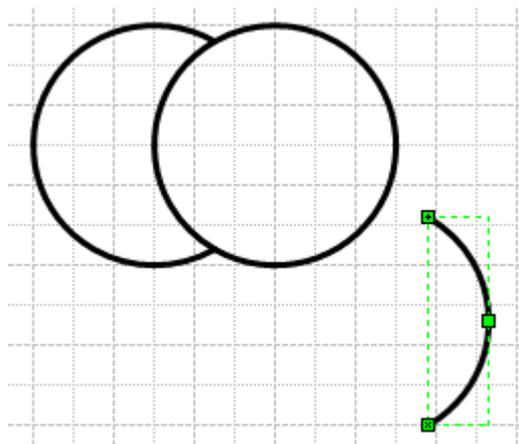
Перемещение одного из фрагментов

Обрезание фигур

Выполняется выбором команды Обрезание. Если начать перемещать выделенный фрагмент, вы увидите, что обрезанный фрагмент – дуга отделился от основной фигуры.



Обрезание фигур



Перемещение фрагмента

1.5. Работа с шаблонами MS Visio

Принцип организации галереи шаблонов

Хранение в галерее шаблонов хорошо структурировано и построено по иерархическому принципу.

Stencil – трафарет. Шаблоны объединяются по тематикам и размещаются в трафарете. Все трафареты в Visio поименованы: basic shapes (основные шаблоны), office equipment (офисное оборудование) и др.

Template – образец. Трафареты, в зависимости от тематики, объединяются в образцы.

Shape – шаблон – является единицей хранения в галерее. Шаблон Visio обладает вполне определенными свойствами, позволяющими работать с шаблоном легко. Их ещё называют «умными шаблонами». Те шаблоны, которые хранятся в галерее, называются **основными**. После перемещения шаблона на лист рисунка, он становится **копией (экземпляром)**.

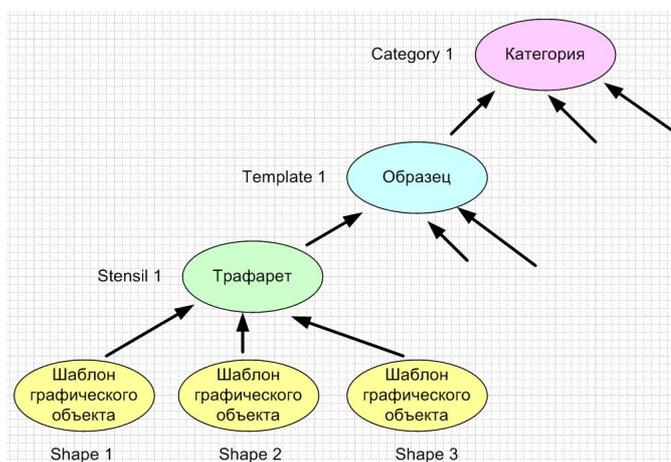
Образец отличается от трафаретов и шаблонов тем, что в нем хранятся не только основные шаблоны и трафареты, но и присущие им такие специфические свойства, как настройки страницы, установки принтера, стили, цветовые гаммы.

Трафареты объединяются в **категию** - **Category**. У них соответствующие названия.

Таким образом, чтобы найти нужный шаблон, нужно:

1) определить, к какой категории относится рисунок

2) выбрать соответствующий образец (template) внутри категории. В результате получится доступ ко всем



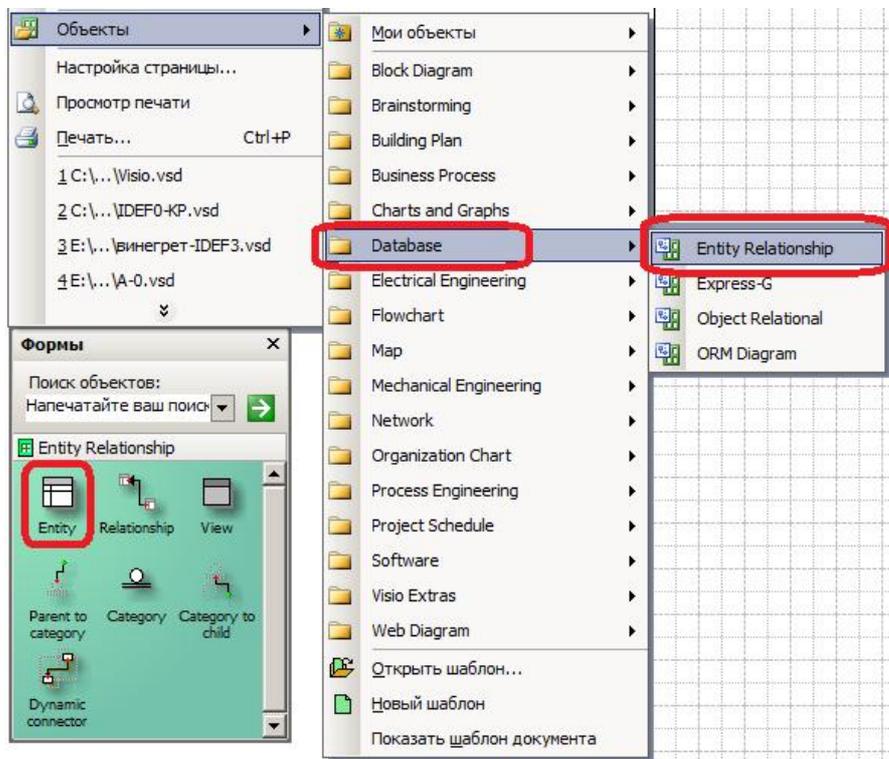
трафаретам, включенным в конкретный образец.

3) указать трафарет (stencil), содержащий нужный шаблон.

В качестве примера на рисунке показано расположение шаблона Entity (Сущность):

- Трафарет – Entity RelationShip;

- Категория – DataBase.



Трафареты категории DataBase

2. Функциональный анализ

Общие сведения

Технология структурного анализа и проектирования SADT (Structured Analysis and Design Technique) представляет собой графический язык и набор процедур для анализа систем. Она позволяет описать систему, систематизировать и структурировать информацию об ее функционировании и прийти тем самым к полному пониманию системы до ее воплощения в жизнь. SADT — это способ уменьшить количество дорогостоящих ошибок за счет структуризации на ранних этапах создания системы, улучшения контактов между пользователями и разработчиками системы.

Методология SADT поддерживает создание множества моделей для более точного описания сложной системы и включает в себя функциональные модели, модели данных и т.д.

2.1. Специфика модели

Рассмотрим функциональное проектирование SADT, стандартизованное и опубликованное под названием IDEF0.

IDEF0-модель состоит из набора диаграмм. Каждая диаграмма содержит блоки и дуги. Блоки изображают функции моделируемой системы, они помечаются глаголами или глагольными оборотами. Дуги связывают блоки вместе и отображают взаимосвязи между ними. Изображение блока со всеми элементами показано на рис. 2.1.



Рис. 2.1. Блок A0 на контекстной диаграмме A-0

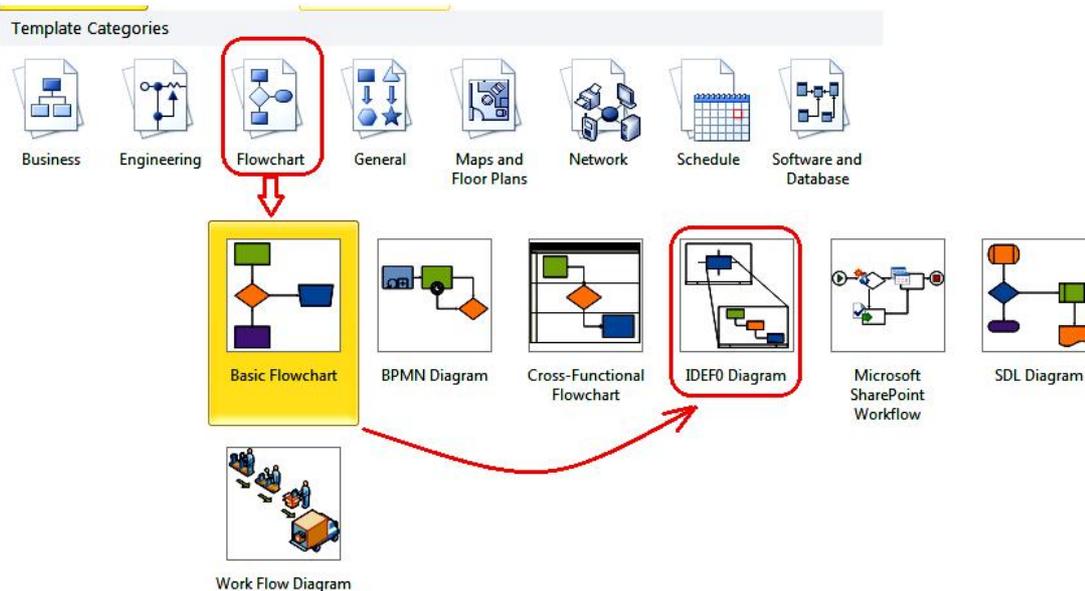
Левая сторона блока предназначена для входов, правая — для выходов, верхняя — для управления, нижняя — для механизмов. Входы преобразуются в выходы, управление ограничивает или предписывает условия выполнения преобразований, механизмы показывают, кто (или что) выполняет функции.

Обобщающая диаграмма верхнего уровня (контекстная диаграмма) является в своем роде уникальной, поскольку содержит один-единственный блок (см. рис. 2.1). Все остальные диаграммы IDEF0-модели обычно содержат от 3 до 6 блоков, при этом каждая такая диаграмма должна детализировать некоторый блок уже имеющейся в модели диаграммы.

Иногда по названию дуг, входящих и выходящих из блоков диаграммы, модель IDEF0 называют ICOM- диаграммой (Input, Control, Output, Mechanism).

2.2. Пример решения задачи функционального анализа процессов системы

Рассмотрим в качестве примера решение задачи анализа процессов, имеющих место во время курсового проектирования по дисциплине «Проектирование информационных систем».



Для графического отображения диаграмм на экране монитора и последующей распечатки на принтере воспользуемся возможностями графического редактора MS Visio. Этот редактор включает в свой набор инструментов шаблоны для разработки различных моделей, в том числе и ICOM-диаграмм. При загрузке редактора выберите группу шаблонов FlowChart (схемы потоков) и уже из этой группы шаблон IDEF0 diagram.

После загрузки программы измените ориентацию страницы с принятой по умолчанию книжной ориентации на альбомную. Вызвав контекстное меню на ярлычке страницы, измените ее название на А-0. Также с помощью контекстного меню добавьте второй лист с именем А0.

Диаграмма А-0

С помощью мыши перенесите графическое изображение блока диаграммы (элемент Activity Box) с панели шаблонов Shapes на рабочее пространство страницы (листа для размещения рисунка). Откроется диалог с окном главных свойств этого блока, в котором нужно ввести имя Курсовое проектирование (рис. 2.2). В результате этих действий будет создан блок А0 и размещен на листе чертежа.

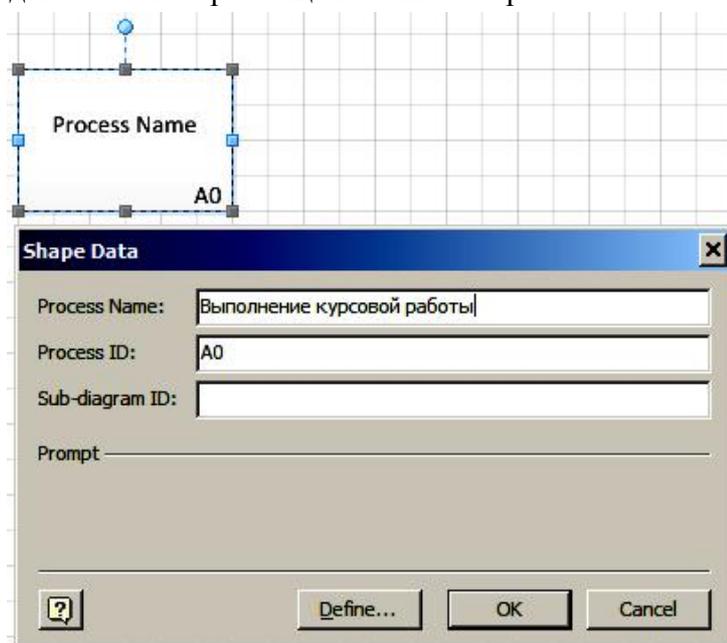
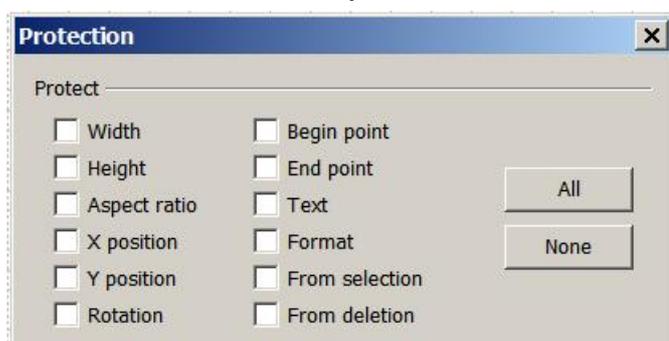


Рис. 2.2. Диалог для ввода имени блока

Возможно, что по умолчанию включена защита блока от изменения. Тогда Вы не



сможете ни изменить размеры блока, например стандартным способом с помощью ручек (grips), ни с помощью окна свойств. Отключим защиту. Вызвав из контекстного меню команду Format (Формат) - Protection (Защита), сбросим все флажки в открывшемся диалоге (рис. 2.3).

Рис. 2.3. Диалог задания элементов защиты блока

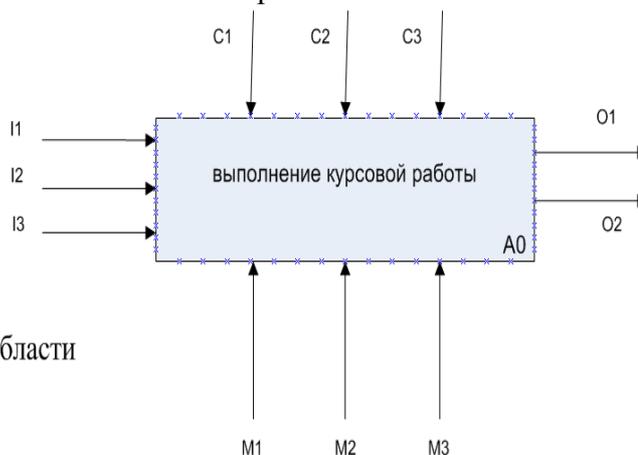
Далее выбрав шаблон IDEF0-connector (соединителя), постройте диаграмму, показанную на рис. 2.4.

Чтобы добавить текст в нужную точку рисунка, используйте команду Text tool. Пиктограмма команды на рис. 2.5 обведена кружочком. К тексту можно применить стандартные команды форматирования: задание размера, гарнитуры, цвета, выравнивания и пр. (рис. 2.5).

Чтобы выполнить курсовой проект, нужно на вход процесса получить задание с темой курсового проекта, необходимую и достаточную информацию о предметной области, различные справочные материалы, обязательные при проектировании.

Таким образом, на первом этапе мы определились, что модель имеет 3 входа: I1, I2, I3.

Важно, что входы должны обязательно располагаться СЛЕВА от блока.



- I1 – задание к курсовой работе
- I2 – информация о предметной области
- I3 – справочные материалы
- O1 – курсовая работа
- O2 – пояснительная записка
- C1 – требования к уровню и объему курсовой работы
- C2 – требования к оформлению пояснительной записки
- C3 – учебный план
- M1 – технические средства
- M2 – программные средства
- M3 – студент

Рис. 2.4. Диаграмма А-0 с блоком А0

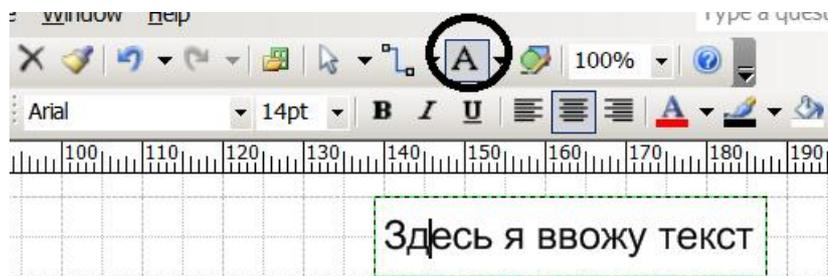


Рис. 2.5. Ввод и форматирование текста

Важно помнить, что моделирование по определению представляет собой итерационный процесс, который нередко требует возвращения на один из ранних этапов. Поэтому Вас, как бизнес-аналитиков, не должен пугать тот факт, что сделанные на первом этапе решения, возможно придется корректировать, в частности добавлять или изменять, например, входы в процесс А0.

На выходе из системы мы должны получить результат процесса:
 а) готовый курсовой проект в форме информационной системы;
 б) пояснительную записку с описанием проектирования объемом 25 – 30 с. Формата А4.

Обозначим выходы как О1 и О2.

Выходы всегда располагаются СПРАВА от блока.

Теперь рассмотрим элементы управления процессом. Курсовое проектирование выполняется строго в соответствии с требованиями, принятыми на кафедре к любому проекту и к проекту по дисциплине в частности. Кроме того, существуют требования к оформлению пояснительной записки, основанные на нормах ГОСТ к текстовым документам или нормативах ЕСКД (Единой системы конструкторской документации). Ну и, очевидно, курсовая работа должна соответствовать положениям учебного и календарного планов для вашей специальности (направления).

Таким образом, мы определились с дугами, входящими в блок СВЕРХУ: С1, С2, С3.

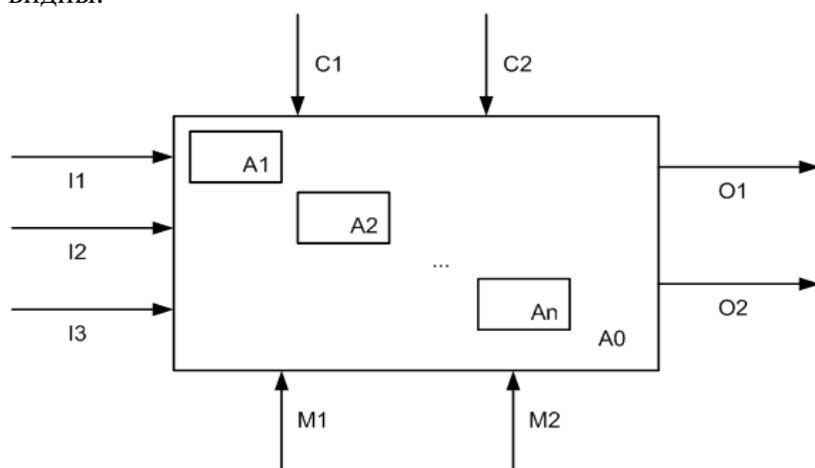
Механизмы для реализации процесса принято располагать СНИЗУ входящими в блок А0. Механизмы – это те инструменты, посредством которых будет реализован основной процесс в системе. В нашем случае к механизмам следует отнести: техническое и программное обеспечения и самого студента, соответственно М1, М2 и М3.

Таким образом, контекстная диаграмма IDEF0, или диаграмма верхнего уровня готова.

Диаграмма А0

Перейдем на второй лист, который мы переименовали в лист А0.

Диаграмма А0 является диаграммой-потомком по отношению к диаграмме А—0, а диаграмма А—0, в свою очередь, является родительской диаграммой для А0. Для стыковки диаграммы-потомка с родительской диаграммой используются ICOM-коды (Input, Control, Output, Mechanism): дуги входов помечаются метками — I1, I2.....дуги выходов — О1, О2,...., дуги управления — С1, С2, ..., дуги механизмов — М1, М2, ... Никаких других входов, выходов, дуг управления или механизмов на диаграмме А0, кроме тех, что присутствовали на А-0, появиться не может. Но дуги могут иметь разветвления и слияния внутри диаграмм. Кроме того, внутри диаграммы А0 появляются внутренние дуги, соединяющие подблоки, и которые, очевидно, на диаграмме А-0 не видны.



Если же в процессе детализирующего анализа определяется новая внешняя дуга, которой не было на А-0, то следует вернуться к диаграмме А-0 и уточнить ее в свете новых фактов. Важно помнить, что соответствие между родительскими и дочерними диаграммами должно быть полным.

Рис. 2.6. Детализация основного процесса

Произведем декомпозицию блока А0 на несколько подпроцессов. Декомпозиция производится в результате детализации основного процесса. Нужно выяснить, какие задачи последовательно нужно решить, чтобы успешно выполнить основной процесс (рис. 2.6).

В нашем случае, чтобы выполнить курсовое проектирование нужно, очевидно:

- 1) выполнить полноценный анализ предметной области;
- 2) на основе проведенного анализа выполнить проектирование системы;
- 3) проект реализовать в работающую систему;
- 4) оформить и распечатать пояснительную записку к проекту.

Таким образом, блок А0 должен быть детализирован на 4 подблока: А1, А2, А3, А4. Их принято размещать последовательно слева-направо лесенкой смещая вниз (рис. 2.7).

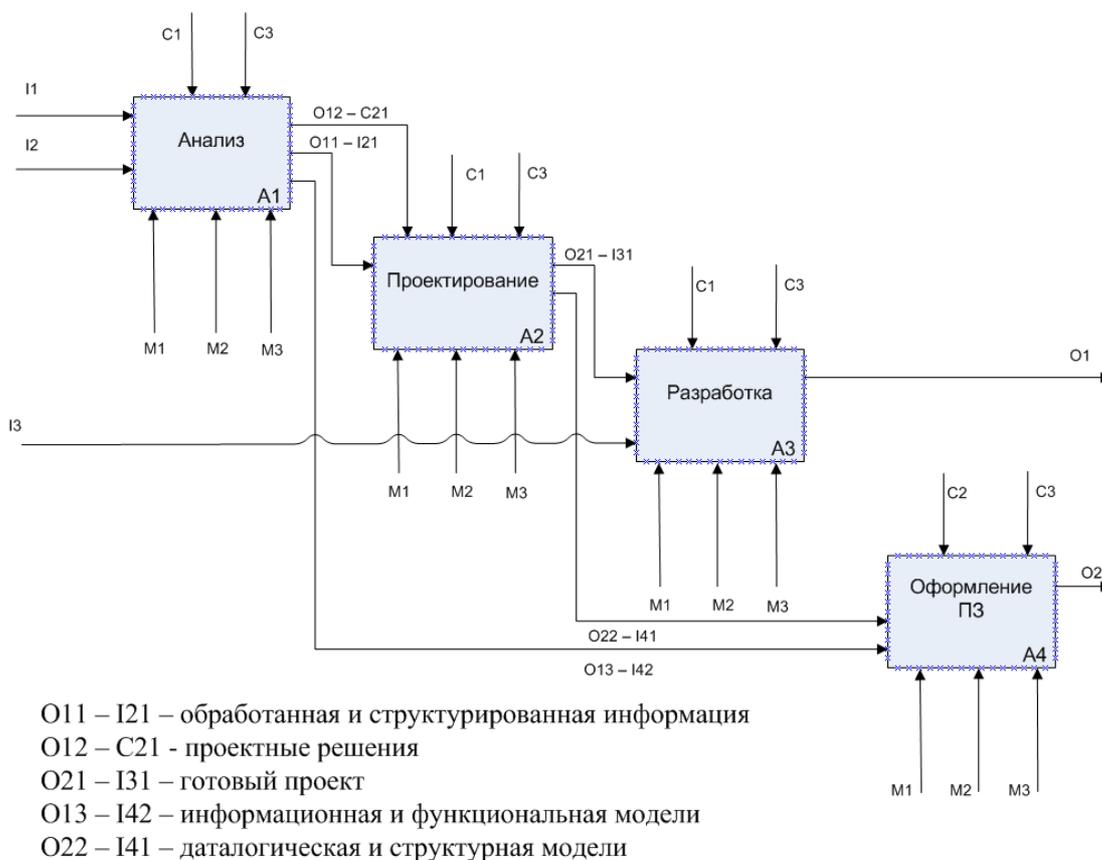


Рис. 2.7. Диаграмма второго уровня А0, детализирующая процесс А0

Используя шаблоны MS Visio, постройте диаграмму А0 на листе А0 так, как показано на рис. 2.7.

Блоки связываются друг с другом внутренними дугами.

На основе анализа предметной области мы получаем 3 результата:

- обработанную и структурированную в форме таблиц, диаграмм информацию о предметной области;
- проектные решения в форме обоснований по видам обеспечений информационной системы, которые нужно реализовать на этапе проектирования;
- функциональную и информационную модели, которые будут использоваться для разработки базы данных и подсистемы управления, а также всех приложений на этапе построения информационной системы. Обозначим их O11, O12, O13. Но одновременно

выходы блока А1 являются входами других блоков. Поэтому этим дугам можно давать двойные обозначения: О11-І21, О12-С21, О13-І42 (рис. 2.7).

Обработанная информация с блока А1 поступает в качестве исходной на блок А2. А вот проектные решения, принятые на этапе анализа, являются уже управляющей нормативной базой при решении задачи проектирования и соответственно подаются на блок А2 как управляющая дуга. Кроме того, чертежи и рисунки моделей, которые были разработаны на этапе анализа, будут использоваться при оформлении пояснительной записки и поэтому являются входами блока А4.

Анализируя процессы подобным образом мы определяемся со всеми остальными внутренними дугами – выходами блоков А2, А3, А4 (рис. 2.7).

Продолжая процесс анализа можно детализировать внутренние блоки А1, А2, А3, А4. Каждая детализация дает свою одноименную диаграмму.

Диаграмма А1

Процесс полноценного анализа предметной области является сложной многоаспектной задачей, составляющей примерно 40% от всего проекта и, соответственно, для лучшего понимания сам требует детализации. На основе декомпозиции этой задачи на 3 – 6 подзадач разрабатывается диаграмма А1 (рис. 2.8).

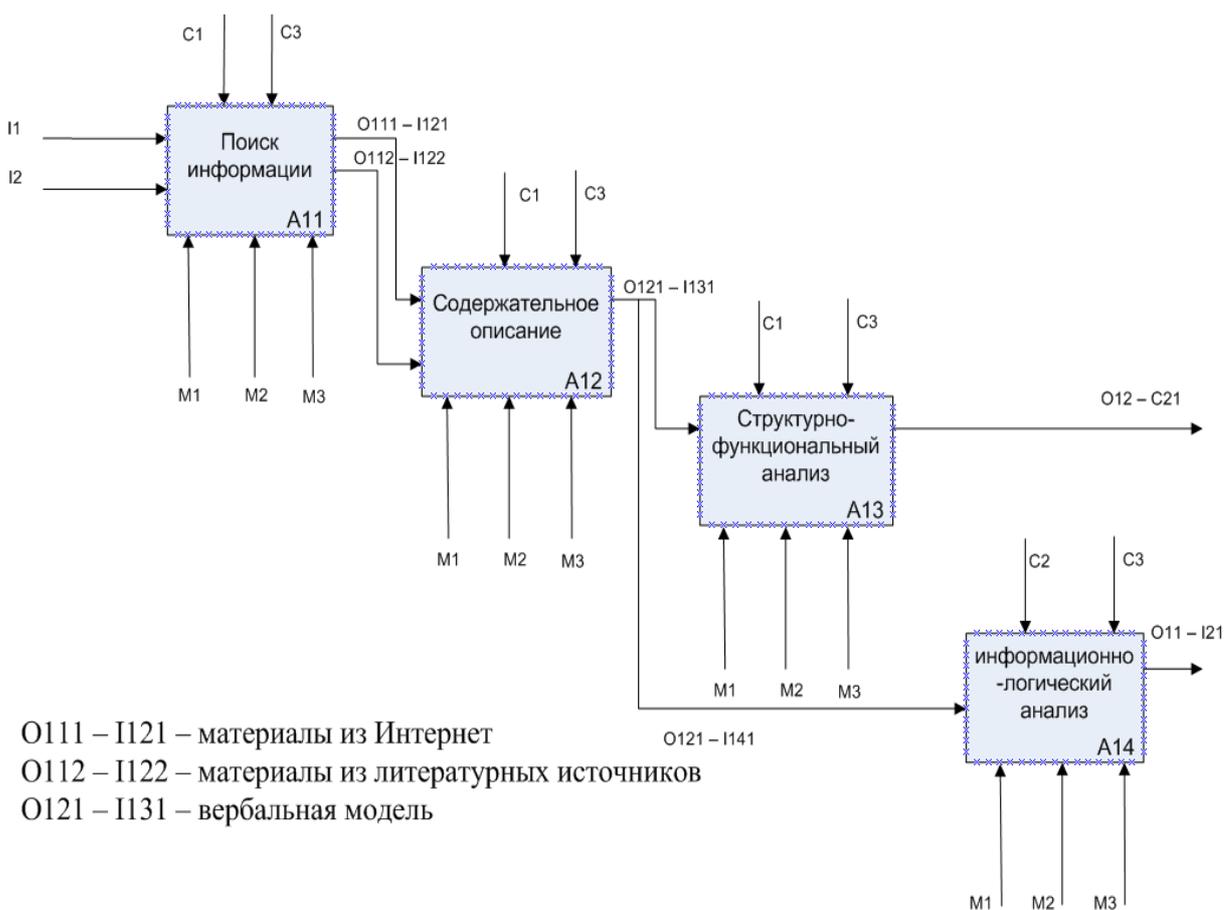


Рис. 2.8. Диаграмма А1

Анализ предметной области с целью разработки информационной системы на основе использования технологии структурного проектирования включает в себя, как правило 4 подпроцесса:

- поиск информации в литературных источниках, сети Интернет, на предприятии во время курсовой и преддипломной практик;
- содержательное описание предметной области, результатом которого становится вербальная модель – модель описания, не использующая формализованных методов;
- функционально-структурный анализ на основе стандартных технологий, использующихся в структурном подходе к проектированию - SADT, Гейна-Сарсона, Йордана-Демарко, ГОСТ 19.701-90 и др.;
- информационно-логический анализ, результатом которого становится информационно-логическая модель предметной области в форме диаграмм сущность-связь (ER-модель).

Для размещения диаграммы A1 используйте следующую страницу MS Visio, на которой выполните построения, показанные на рис. 2.8.

Диаграмма A2

Действуя аналогичным образом, выполним детализацию блока A2. Как известно, проектирование считается одним из наиболее сложных видов деятельности человека и проектирование информационных систем не является исключением. Поэтому для лучшего понимания процесса и недопущения ошибок разберем его способом декомпозиции на более простые подпроцессы. Результат анализа в форме диаграммы A2 показан на рис. 2.9.

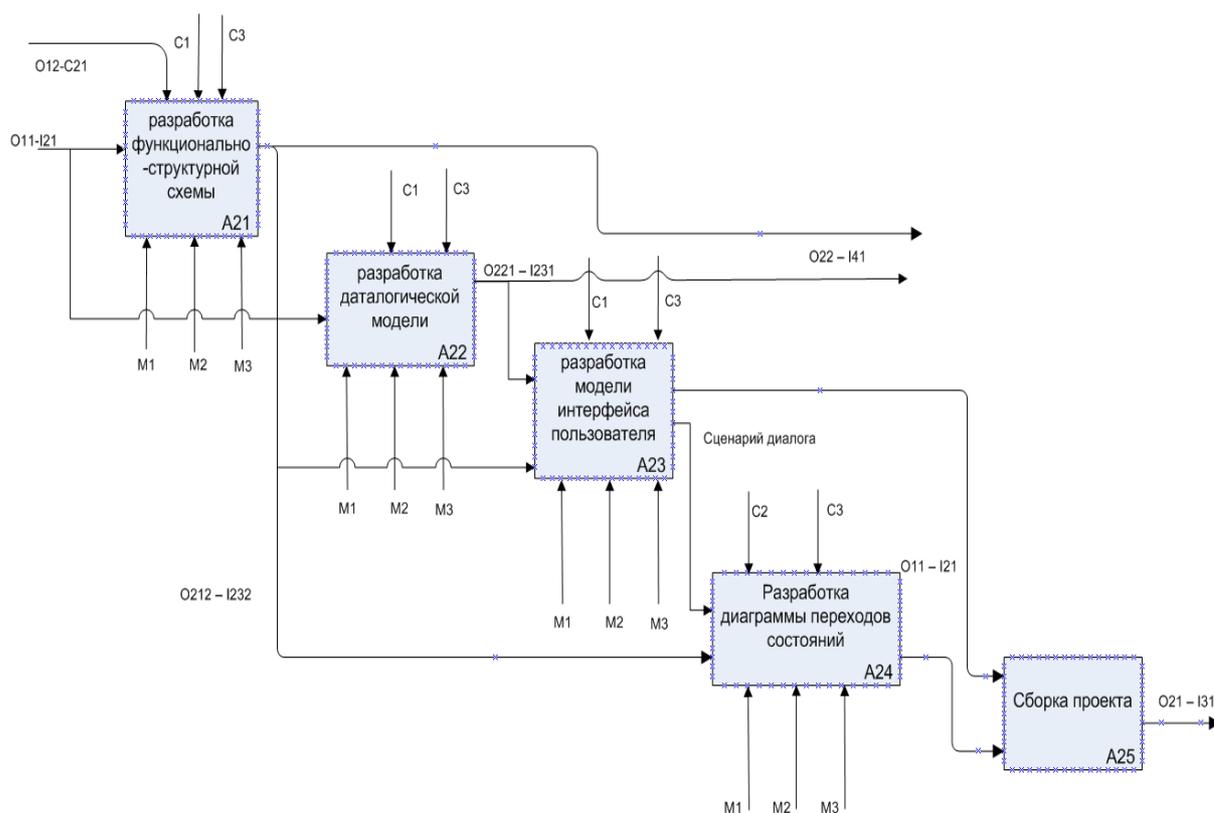


Рис. 2.9. Диаграмма A2

Для размещения диаграммы добавьте новый лист в файл MS Visio.

Описание подпроцессов и спецификацию выходов-входов блоков выполните самостоятельно.

Диаграмма А3 и А4

Построение диаграмм А3 и А4 выполните самостоятельно.

Диаграммы третьего уровня

Нередко оказывается, что и процессы на втором уровне детализации, например блок А14 или А25, достаточно сложны по структуре и поэтому должны быть тоже детализированы. Таким образом, при дальнейшем моделировании могут появиться диаграммы А14, А22 или А25. Как правило декомпозиция блоков прекращается тогда, когда внутренние процессы блока становятся в полной мере понятны и их реализация становится очевидна.

3. Функционально-структурный анализ способом разработки IDEF3-модели

3.1. Основные элементы модели

Специфика модели

В общем случае, процесс – это упорядоченная последовательность действий.

Процессная модель IDEF3 *позволяет*:

- Отразить последовательность процессов
- Показать логику взаимодействия элементов системы.

Цель IDEF3 - дать возможность аналитикам описать ситуацию, когда процессы выполняются в определенной последовательности, а также описать объекты, участвующие совместно в одном процессе.

Основными элементами IDEF3-модели являются:

- 1) единицы работ;
- 2) связи;
- 3) перекрестки;
- 4) объекты ссылок.

Единица работ (UOW, Unit of Work) является центральным компонентом модели.

Имя работы	
номер	

Номер работы является уникальным, присваивается при ее создании и никогда не меняется. Имя работы - словосочетание с отглагольным существительным, изображающим действие (выполнение, изготовление, ...) или Инфинитив глагола (изготовить продукцию)

Связи показывают взаимоотношения работ.

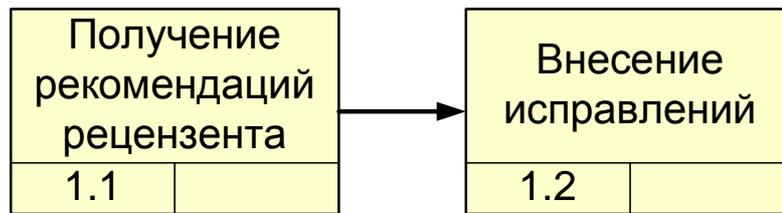
Связи однонаправлены и обычно диаграммы рисуют таким образом, чтобы связи были направлены слева-направо

Различают 3 типа связей:

- Старшая стрелка
- Стрелка отношений
- Поток объектов.

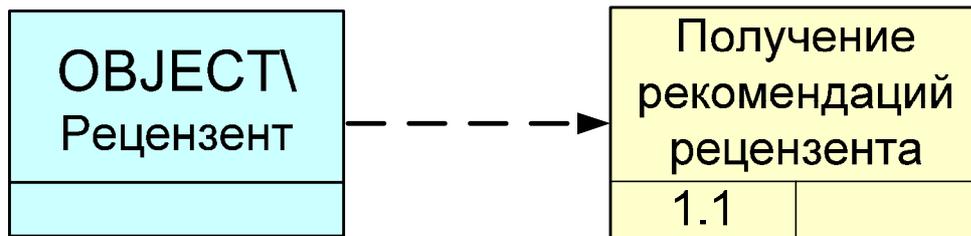
Старшая стрелка: связь типа «временное предшествование» - Precedence

Соединяет единицы работ. Показывает, что работа-источник должна быть закончена прежде, чем начнется работа-цель. Рисуеться в форме простой стрелки.



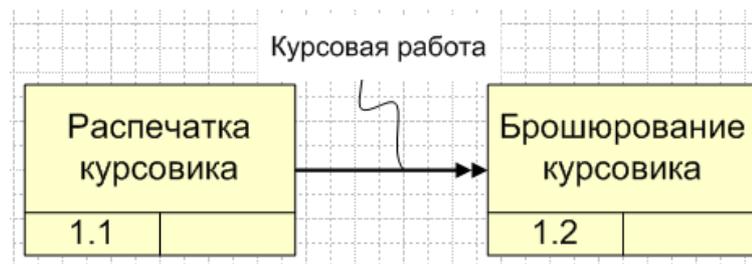
Стрелка отношений: связь типа нечеткое отношение - Relational

Изображается в виде пунктирной линии, используется для изображения связи между единицами работ, а также между единицами работ и объектами ссылок.



Поток объектов: Object Flow

Применяется для описания того факта, что объект используется в двух и более единицах работ, например, когда объект порождается в одной работе и используется в другой.



Перекрестки

Используются для отображения логики взаимодействия стрелок при их слиянии или разветвлении, для отображения множества событий, которые могут или должны быть завершены перед началом следующей работы.

Различают перекрестки для слияния и разветвления стрелок.

Перекрестки не могут быть одновременно использованы для слияния и разветвления стрелок. Все перекрестки на диаграммах нумеруются, каждый номер имеет префикс J.

В отличие от других методологий (IDEF0, DFD) стрелки могут сливаться или разветвляться только через перекрестки.

Правила расположения перекрестков

1. Каждому перекрестку для слияния должен предшествовать перекресток для разветвления.
2. Перекресток для слияния «И» не может следовать за перекрестком для разветвления типа синхронного или асинхронного «ИЛИ»
3. Перекресток для слияния «И» не может следовать за перекрестком типа исключительного «ИЛИ»

4. Перекресток для слияния типа исключительного «ИЛИ» не может следовать за перекрестком для разветвления типа «И»
5. Перекресток, имеющий одну стрелку на одной стороне, должен иметь более одной стрелки на другой.

Обозначение	Тип перекрестка	Логика функционирования	
		Разветвление	Слияние
	<i>Асинхронный «И»</i>	Каждая из исходящих ветвей активируется	Все входящие ветви должны быть завершены
	<i>Синхронный «И»</i>	Все исходящие ветви активируются одновременно	Все входящие ветви должны завершиться одновременно
	<i>Асинхронный «ИЛИ»</i>	Активируется хотя бы одна из исходящих ветвей	Должна завершиться хотя бы одна из ветвей
	<i>Синхронный «ИЛИ»</i>	Если активируются две и более ветвей, то они активируются одновременно	Если должны завершиться две и более ветвей, то они завершаются одновременно
	<i>Исключающее «ИЛИ»</i>	Активируется только одна исходящая ветвь	Должна завершиться только одна входящая ветвь

Объект ссылок

Объект ссылок выражает идею, концепцию данных, которые нельзя связать со стрелкой, перекрестком, работой; используется при построении диаграммы для привлечения внимания пользователя к каким-либо важным аспектам модели.



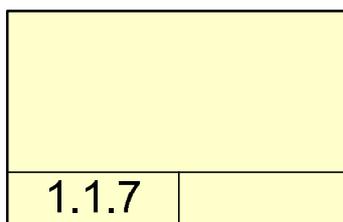
Детализация работ

В IDEF3 декомпозиция используется для **детализации** работ.

Методология IDEF3 позволяет декомпозировать работу **множественно**, т.е. работа может иметь множество дочерних работ. Это позволяет в одной модели описать альтернативные потоки.

Возможность множественной декомпозиции предъявляет дополнительные требования к нумерации работ.

Номер работы, например 1.1.7, состоит из номера родительской работы 1, версии декомпозиции 1 и собственного номера работы на текущей диаграмме 7.



3.2. Создание шаблона объектов модели IDEF3

При разработке модели IDEF0 мы воспользовались готовыми шаблонов объектов, которые предоставлены MS Visio. Если в библиотеке шаблонов отсутствуют шаблоны IDEF3, то перед началом построения модели нужно самостоятельно его разработать.

Для этого сами создаем файл шаблона с именем IDEF3. Если файлы рисунков MS Visio имеют расширение *.vsd (visio drawing), то файлы шаблонов - расширение *.vss (visio stencil).

На листе чертежа с помощью инструментов панели Picture (Рисунок) создаем объект Работа – UOW (Union Of Work) – такой, как показано на рис. 3.1. Используя операцию Группирование, расположенной на вкладке Разработчик Ленты MS Visio, превращаем несколько линий и надписей в единую группу. Теперь можно сделать заливку этого объекта. Для заливки используйте светлые тона, например, светло-желтый цвет.

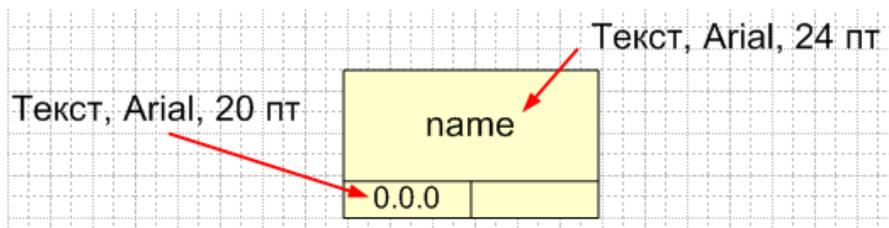


Рис. 3.1. Шаблон объекта Работа

С помощью мыши перенесите созданный объект влево - в область расположения элементов нового шаблона. По умолчанию он получит имя Master.0. С помощью контекстного меню переименуйте его в UOW (рис. 3.2).

Таким же образом создайте и сохраните шаблон объекта ссылок (рис. 3.3). Для него цвет заливки выберите светло-голубой. После переноса его в область объектов шаблона дайте новое имя – Object.

Теперь можно приступить к созданию шаблонов перекрестков. Цвет заливки для них определите как светло-зеленый (рис. 3.4). Для обозначения имен логических операций используйте надпись с размером текста в 32 пт. Перенесите созданные объекты в область шаблона, переименуйте их в соответствии с теми операциями, которые они выполняют, например, A-AND, S-AND, A-OR, S-OR, XOR.

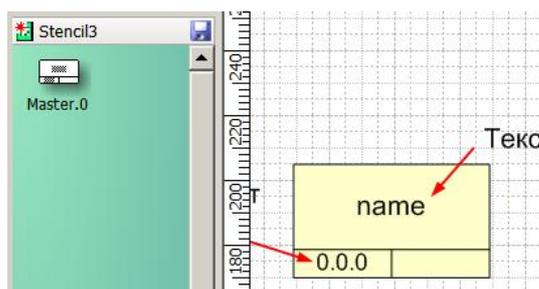


Рис. 3.2. Создание нового объекта шаблона

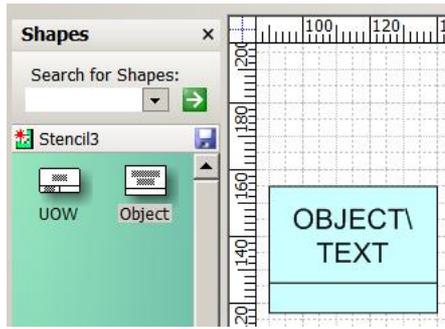


Рис. 3.3. Объект ссылок



Рис. 3.4. Перекрестки

Четвертым типом объектов в модели IDEF3 являются стрелки. Создайте, используя графический примитив Динамический соединитель (Connector), сделайте 3 шаблона: стрелка старшинства (темно-синяя сплошная), стрелка отношений (черная пунктирная), стрелка потока объектов (красная со сдвоенным указателем) – рис. 3.5.

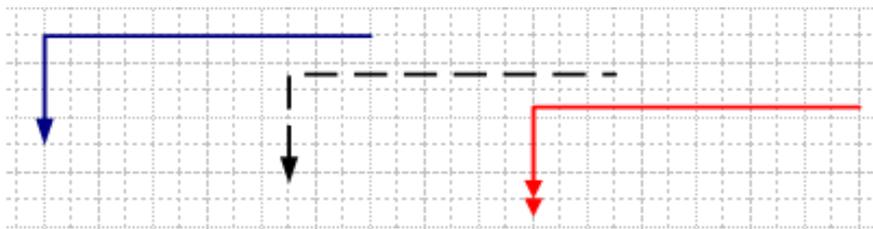


Рис. 3.5. Стрелки для модели IDEF3

Для создания двойной стрелки отредактируйте обычную стрелки с помощью контекстного меню Формат – Линия (рис. 3.6).

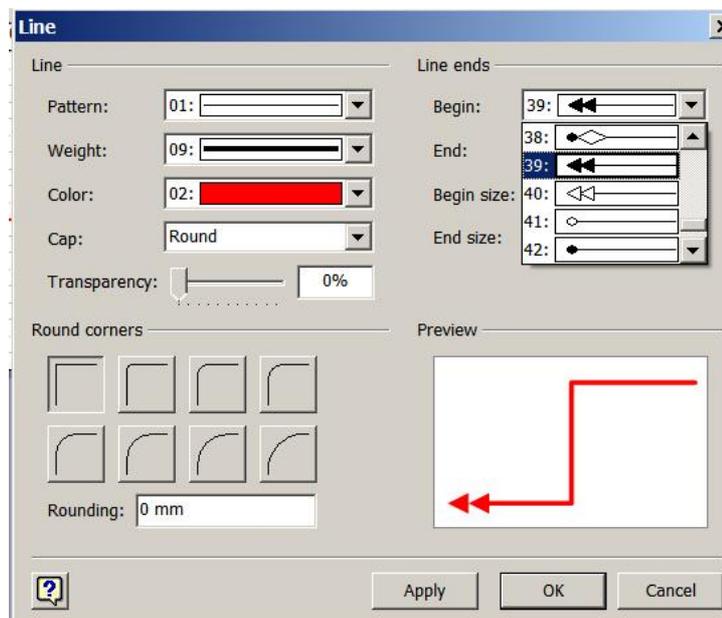


Рис. 3.6. Редактирование линии

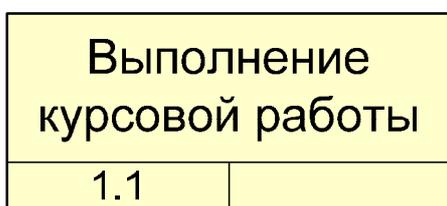
Сохраните стрелки в области шаблонов, задав им соответствующие имена Precedence, Relational, ObjectFlow.

Шаблоны готовы. Их нужно сохранить в файле, задав ему имя IDEF3.vss. Сохранение можно сделать с помощью контекстного меню для всего набора шаблона. Файл шаблона сохраняется в папке My Shapes Вашей учетной записи Windows.

После сохранения все закройте. Причем файл рисунка, с которым Вы работали, сохранять не нужно. В следующий раз Вы можете использовать сохраненный шаблон, выбрав при создании файла рисунка пункт Шаблоны на моем компьютере. С ним можно работать как с обычным шаблоном MS Visio.

3.3. Пример построения IDEF3-диаграммы

Рассмотрим на примере построения модели процесса «**Выполнение курсовой работы**»



Лист 1. Диаграмма 1.1

Начнем с построения контекстной диаграммы.

В новом файле чертежа создайте несколько новых страниц. На странице 1.1 в верхней ее части разместите контекстную диаграмму 1.1, показанную на рис. 3.7.

Рис. 3.7. Контекстная диаграмма 1.1

На том же листе, используя созданные ранее шаблоны, построим детализирующую модель, выполнив декомпозицию контекстной диаграммы, показанную на рис. 3.7.

1. Начнем со списка работ, которые нужно выполнить, чтобы решить задачу **Выполнение курсовой работы**:

- 1) получение задания;
- 2) подбор нужной литературы для решения задачи;
- 3) посещение консультаций руководителя курсовой работы;
- 4) выполнение разделов работы;
- 5) оформление пояснительной записки;
- 6) защита курсовой работы.

2. Создание графических объектов для отображения работ.

Эти шесть работ нужно пока произвольно разместить на втором листе MS Visio. Установите ориентацию страницы – альбомная (Landscape).

Каждому объекту задайте текстовую надпись, соответствующую названию работы.

Задайте номер каждой работе, начиная с 1.1.2 (детализация – блок №1, версия детализации - №1, номер текущего блока – 2, ... 7).

3. Все работы выполняются последовательно друг за другом. Определим, какие работы можно или нужно выполнять параллельно. Очевидно, это **Выполнение разделов** и **Посещение консультаций**. Все блоки работ размещаем горизонтально вдоль прямой, кроме тех, которые должны выполняться параллельно (рис. 3.8).

4. Постановка перекрестков.

Разветвление всего одно – после блока 1.1.3 и слияние тоже только одно – перед блоком 1.1.6. Сначала определим тип перекрестка. Можно ли не посещать консультации? Как правило, посещение студентами консультаций обязательно. Также обязательно

выполнение всех разделов курсовой работы. Таким образом, перекрестки типа OR и XOR здесь не подходят. Выбираем логическое умножение AND.

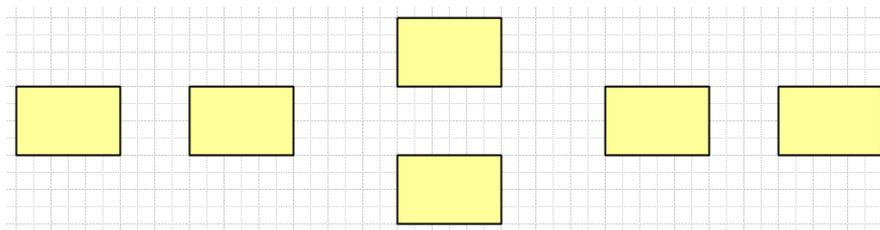


Рис. 3.8. Размещение блоков

Теперь определим, Выполнение разделов курсовой работы и посещение консультаций синхронные процессы или асинхронные. Очевидно, что они асинхронны.

Тогда 2 асинхронных перекрестка выбираем из шаблона и устанавливаем после блока 1.1.3 и перед блоком 1.1.6. Используя надпись, поставьте рядом с перекрестками их обозначения – J1 и J2.

5. Установление связей.

Все работы, кроме параллельных 1.1.4 и 1.1.5, должны выполняться последовательно, строго после завершения предыдущей работы. Таким образом, все связи – стрелки старшинства. Переносим шаблон стрелки старшинства на чертеж и, т.к. в качестве стрелок мы взяли динамический соединитель, нужно захватить связываемые объекты. В момент захвата стрелкой объекта, вокруг него появляется красная рамка. Установите все связи работ между собой так, как показано на рис. 3.9.

6. Добавление объектов ссылок.

В системе рассматриваемых работ есть один внешний объект – преподаватель, он же - руководитель курсовой работы. Он имеет отношение к выполнению двух работ: Получение задания и Посещение консультаций. Обратите внимание, что Преподаватель в соответствие с логикой выполнения курсовой работы является объектом-ссылкой, который не может быть связан с работами, но является важной частью процесса.

Используя соответствующий шаблон, добавьте объект ссылки Преподаватель на диаграмму и соедините его с соответствующими работами пунктирной линией так, как показано на рис. 3.9.

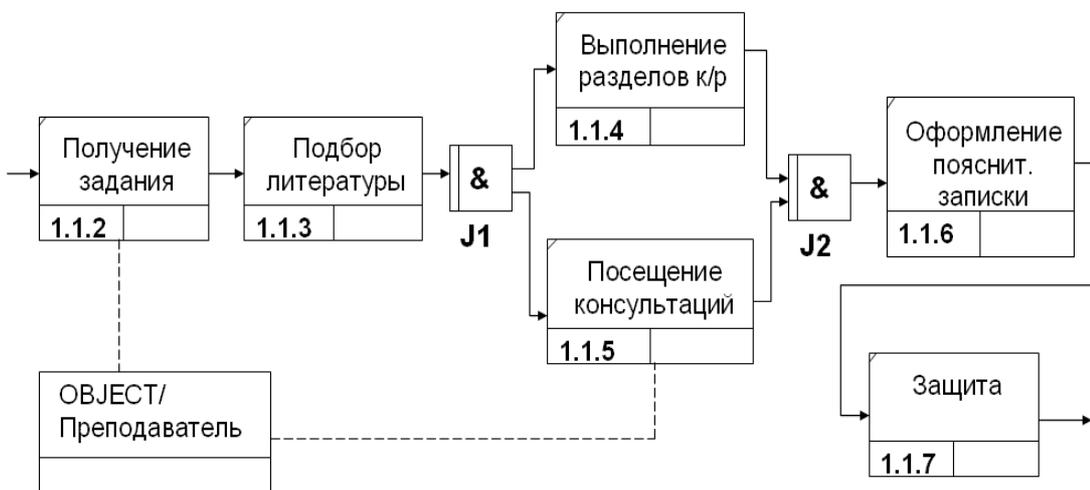


Рис. 3.9. Диаграмма детализации блока 1.1

Лист 2. Следующий уровень детализации

Выполним декомпозицию блока 1.1.4 – Выполнение разделов курсовой работы.

Скопируем блок 1.1.4 с первого листа в верхнюю часть листа 2. Здесь же на втором листе разместим диаграмму, детализирующую этот блок.

1. Начнем со списка работ, которые составляют суть работы Выполнение разделов курсовой работы:

- 1) содержательное описание предметной области;
- 2) формализованный анализ предметной области;
- 3) проектирование базы данных может быть на основе двух альтернативных концепций:
- 4) разработка реляционной БД;
- 5) разработка объектно-ориентированной БД;
- 6) проектирование программных модулей;
- 7) проектирование интерфейса пользователя;
- 8) сборка системы.

2. Графическое отображение всех работ на листе. Эти семь работ разместите произвольно на листе, используя готовые шаблоны.

Задайте номер каждой работе, начиная с 4.1.8 (детализация – блок №4, версия детализации - №1, номер текущего блока – 8, ... 14).

3. Определим последовательность выполнения работ и те работы, которые могут или должны быть выполнены параллельно. Очевидно, что альтернативные работы 10 и 11 должны изображаться параллельно. Кроме того, работы по проектированию программных модулей и интерфейса пользователя можно и даже должно вестись параллельно (рис. 3.10).

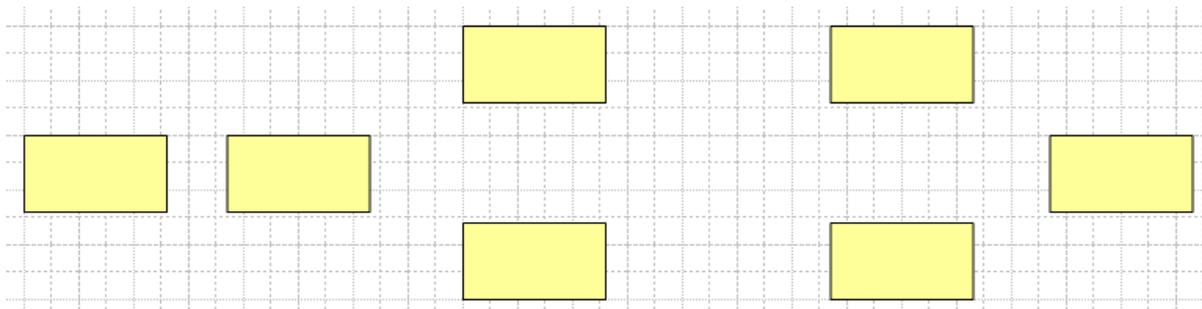


Рис. 3.10. Расположение работ на втором листе

4. Расстановка перекрестков

Ветвление начинается после 9-го блока и завершается после 10-го и 11-го. Затем вновь стрелка разделяется перед 12-ым и 13-ым и соединяется перед 14-ым.

Очевидно, что 12-ая и 13-ая работы не являются альтернативами и должны быть выполнены обе. Причем начинаться и завершаться они должны синхронно – переход к сборке проекта без хотя бы одной из них не возможен. Поэтому в качестве перекрестков придется выбрать синхронное AND.

А вот 10-ая и 11-ая, напротив, являются альтернативными работами и здесь, соответственно, нужно использовать перекресток XOR (вспомним, он бывает только асинхронным).

Нумерация перекрестков: J3, J4, J5, J6.

5. Установление связей. Все связи – стрелки старшинства.

6. Объекты ссылок.

1) К перекрестку J3 нужно подвести объект ссылок типа ELAB (Уточнение) с уточнением: «При структурном подходе в блоке 4.1.2 – реляционная БД, при ОО – ОО БД» (рис. 3.11 а).

2) К блоку Проектирование интерфейса пользователя нужно подвести объект ссылок типа ОБЪЕКТ «Среда визуального программирования» (рис. 3.11 б).

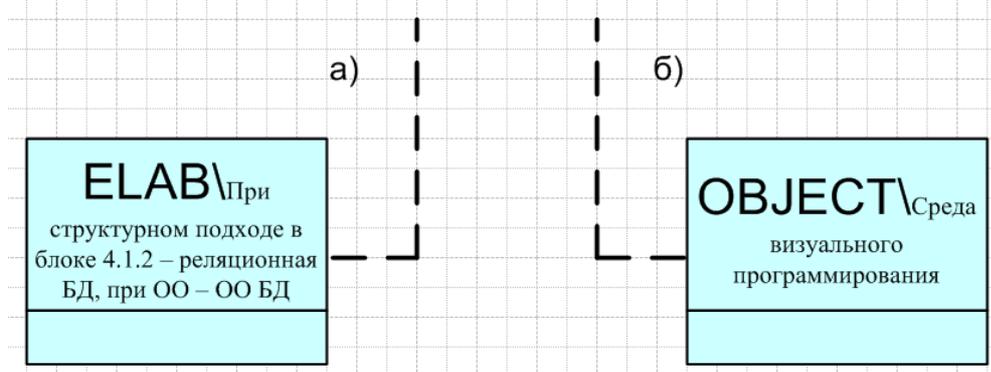


Рис. 3.11. Объекты ссылок

Лист 3. Детализация блока 4.1.9.

Скопируйте блок 4.1.9 со второго листа на третий и разместите его в верхней части.

При детализации этого блока нужно учитывать тот факт, что анализ может быть выполнен на основе двух альтернативных методологий: структурного подхода и объектно-ориентированного. Следовательно, здесь на этом листе должны расположиться 2 ревизии одной работы.

Структурный подход предполагает наличие блоков:

– функциональный анализ – 2 альтернативных варианта:

IDEF0-анализ (9.1.15);

DFD-моделирование (9.1.16).

- информационный анализ – ER-диаграмма (9.1.17).

Объектно-ориентированный анализ предполагает разработку модели, как минимум, из двух диаграмм:

- функциональный анализ – диаграмма прецедентов (9.2.18);

- информационный анализ – диаграмма классов (9.2.19).

С учетом перекрестков должна получиться модель, показанная на рис. 3.12.

Здесь еще можно было бы, наверное уточнить, что на выходе блока 9.1.17 получается готовая ER-модель, которая служит основой для построения реляционной базы данных, а на выходе блока 9.2.19 – диаграмма классов, которая необходима для разработки объектно-ориентированной базы данных. В таком случае, вместо стрелки старшинства из этих блоков должна выходить стрелка объектов с двойным наконечником.

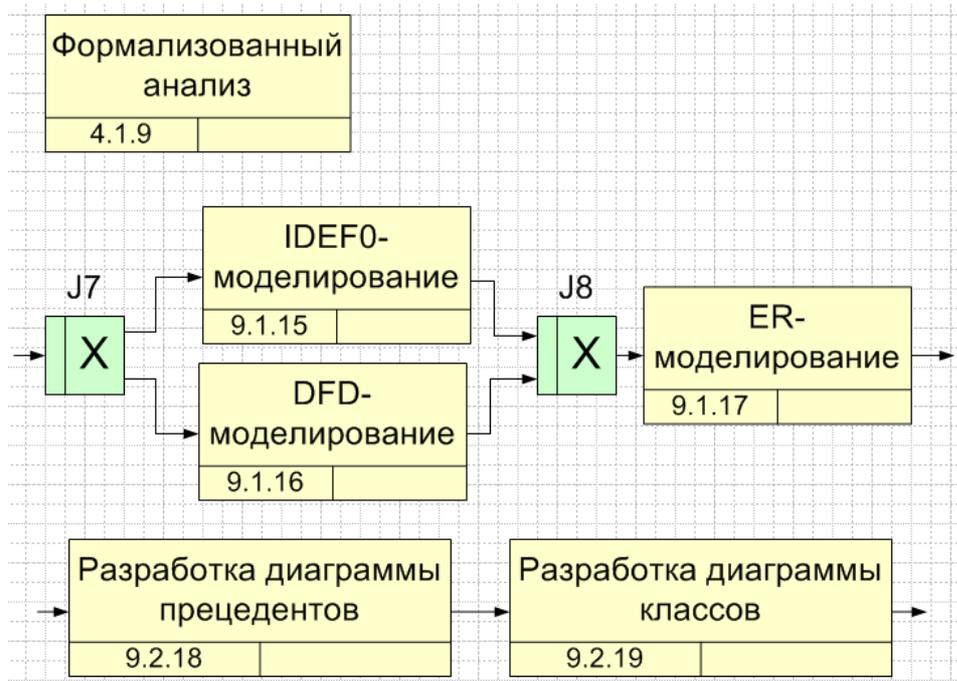


Рис. 3.12. Диаграмма блока 4.1.9

Лист 4. Альтернативная детализация блока 1.1.4.

На листах 2 и 3 была выполнена детализация работы Выполнение разделов курсовой работы по дисциплине «Проектирование информационных систем». Если же курсовая работа выполняется по предмету «Программирование», где предполагается в обязательном порядке реализация вычислительных алгоритмов, но исключается необходимость разработки базы данных, то детализация блока 1.1.4 будет совершенно другой (рис. 3.13).

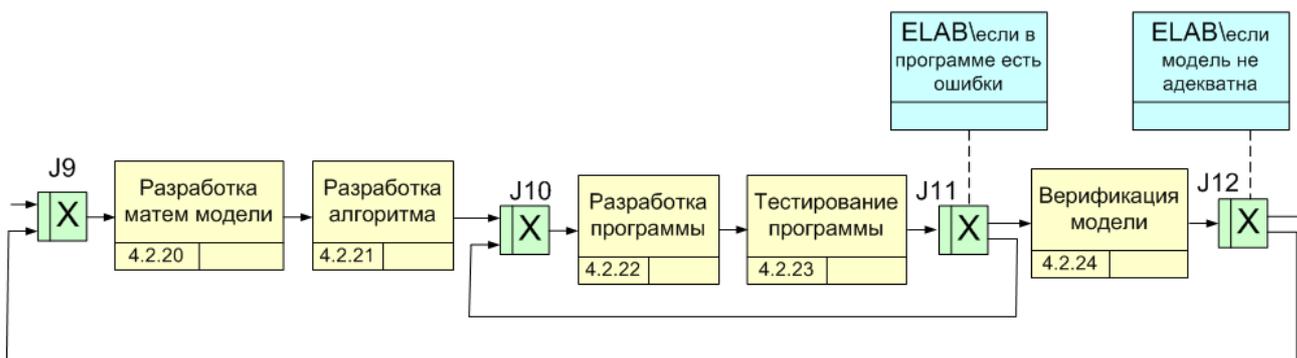


Рис. 3.13. Альтернативная детализация блока 1.1.4

На этой диаграмме показан итерационный процесс разработки программы, когда итерации задаются с помощью перекрестков и уточняющих объектов ссылок. Если есть необходимость задать циклический повтор, необходимо использовать объект ссылку типа GOTO.

4. Информационно-логический анализ на основе ER-диаграмм

4.1. Общие сведения

Понятие “данные”— это набор конкретных значений, параметров, характеризующих объект, условие, ситуацию или любые другие факторы. Модель данных — это некоторая абстракция, которая, будучи приложима к конкретным данным, позволяет пользователям и разработчикам трактовать их уже как информацию, то есть сведения, содержащие не только данные, но и взаимосвязь между ними.

Общая классификация моделей данных приводится на рис. 4.1.

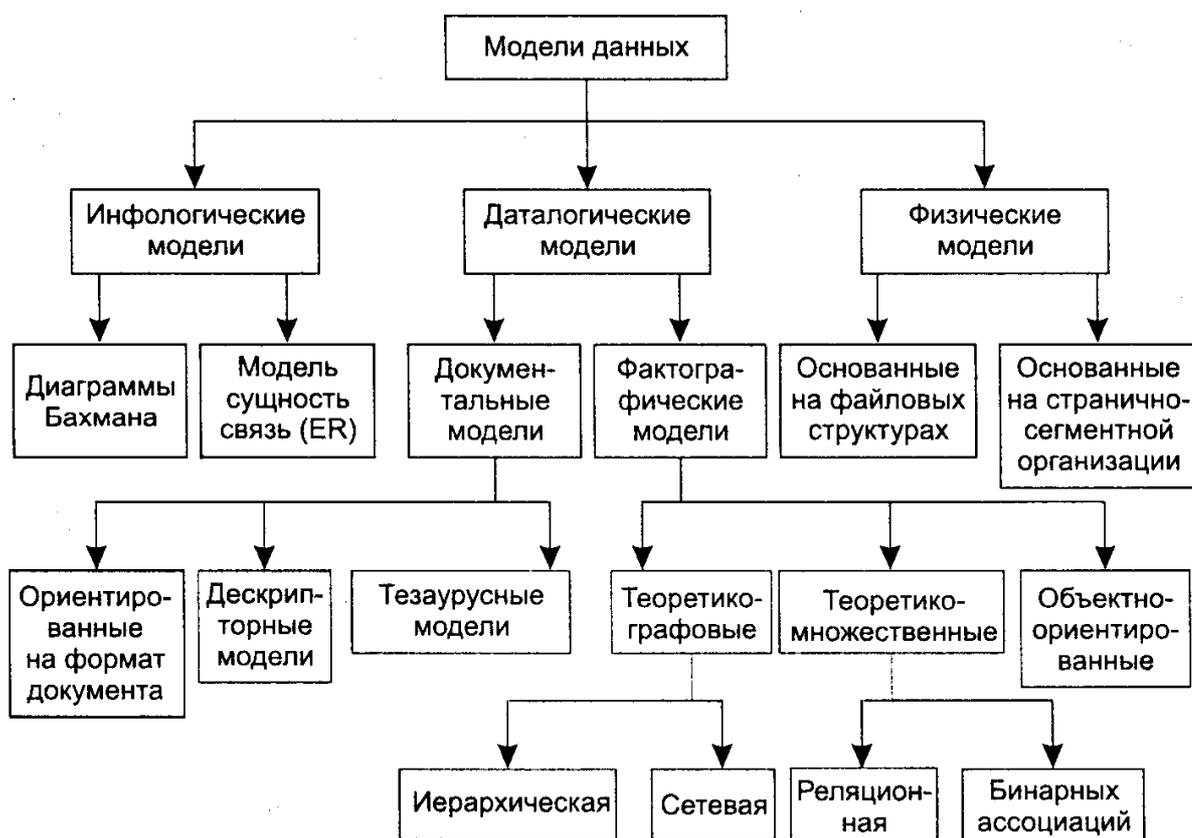


Рис. 4.1. Классификация моделей данных

В данной теме рассматривается только класс инфологических моделей и, частности, модели типа «сущность-связь» (ER-модели).

Основные элементы модели

Сущность (Entity), с помощью которой моделируется класс однотипных объектов.

Сущность имеет имя, уникальное в пределах моделируемой системы. Так как сущность соответствует некоторому классу однотипных объектов, то предполагается, что в системе существует множество экземпляров данной сущности. Объект, которому соответствует понятие сущности, имеет свой набор атрибутов — характеристик, определяющих свойства данного представителя класса. При этом набор атрибутов должен быть таким, чтобы можно было различать конкретные экземпляры сущности. Например, у сущности Клиент может быть следующий набор атрибутов: Номер_паспорта, Фамилия, Имя, Отчество, Адрес, Телефон.

Набор атрибутов, однозначно идентифицирующий конкретный экземпляр сущности, называют ключевым (первичный ключ – primary key - PK). Для сущности

Клиент ключевым будет атрибут Номер_паспорта, поскольку для всех граждан – клиентов фирмы номер и серия паспорта будут различны.

На практике применяются разные способы (нотации) изображения **ER-моделей**. В пособии используется нотация MS Visio (рис. 4.2).



Рис. 4.2. Изображение сущности в нотации MS Visio

Инфологическая модель отображает предметную область с точки зрения не только наличия в ней информационных объектов, но наличия и логических связей между ними (Relationship). Отсюда название диаграммы – ER.

Связи — это бинарные ассоциации, показывающие, как связаны экземпляры отношений между собой. Например, существует связь между сущностями “Клиент” и “Заказы_товаров”, - каждый клиент некоторой торговой компании, может быть участником нескольких заказов, выполненных в разное или одно и то же время, но один и тот же заказ имеет отношение только к одному клиенту. Поэтому рассматриваемая связь по мощности является связью “**один-ко-многим**” (1:M).

Связь устанавливается между первичным ключом одной сущности и внешним ключом другой сущности. Внешний ключ (Foreign Key - FK) – это атрибут в связанной сущности, который принимает значения, совпадающие со значениями РК главной сущности.

Важно, что все сущности должны быть обязательно связаны между собой.



Рис. 4.3. Изображение связи между сущностями в нотации Баркера (сверху) и MS Visio (снизу)

4.2. Пример разработки ER-модели

Требуется разработать информационно-логическую модель предметной области – «Распределение единиц оборудования вычислительной техники».

Работа с шаблоном и файлом модели

Для эффективной разработки ER-моделей в MS Visio можно воспользоваться готовым шаблоном Entity Relationship (рис. 4.4).

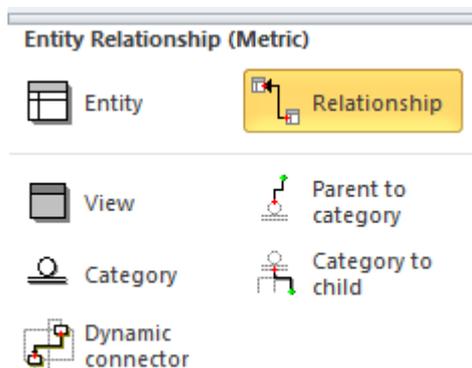
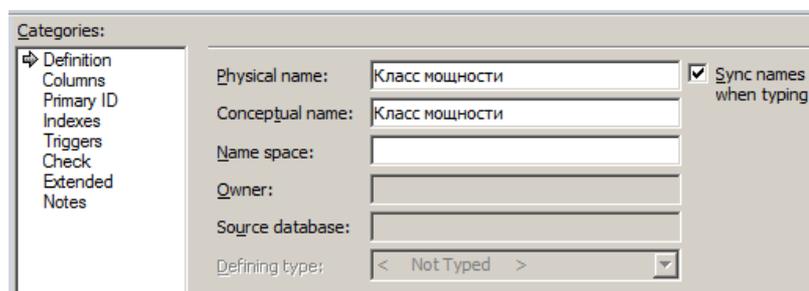


Рис. 4.4. Шаблон MS Visio Entity Relationship

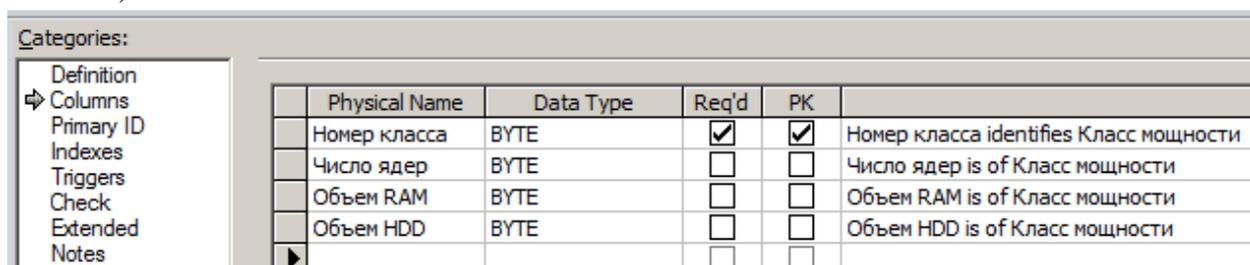
После выбора графического объекта типа Entity (Сущность), появляется диалог для ввода ее имени и далее таблица для ввода свойств (рис. 4.5).

В категории Колонки следует задать атрибуты сущности, тип данных для каждого атрибута, обязательность/необязательность значения, указать первичный ключ (рис. 4.5).

а)



б)



	Physical Name	Data Type	Req'd	PK	
	Номер класса	BYTE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Номер класса identifies Класс мощности
	Число ядер	BYTE	<input type="checkbox"/>	<input type="checkbox"/>	Число ядер is of Класс мощности
	Объем RAM	BYTE	<input type="checkbox"/>	<input type="checkbox"/>	Объем RAM is of Класс мощности
	Объем HDD	BYTE	<input type="checkbox"/>	<input type="checkbox"/>	Объем HDD is of Класс мощности
			<input type="checkbox"/>	<input type="checkbox"/>	

Рис. 4.5. Ввод имени сущности «Класс мощности» (а) и ее свойств (б)

Не забывайте периодически выполнять сохранение диаграммы в файле (по умолчанию vsd-формат или vsdx-формат).

Аналогично создаем сущности:

- Кабинеты;
- Единицы оборудования;

- Подразделения;
- Сотрудники.

	Physical Name	Data Type	Req'd	PK
	Номер	CHAR(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Назначение	CHAR(10)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	Число АРМ	BYTE	<input type="checkbox"/>	<input type="checkbox"/>
	▶ телефон	CHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>

	Physical Name	Data Type	Req'd	PK
	Инв. номер	CHAR(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Название	CHAR(10)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	Назначение	CHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>

	Physical Name	Data Type	Req'd	PK
	ID_подразделения	CHAR(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Название	CHAR(10)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	Число сотрудников	BYTE	<input type="checkbox"/>	<input type="checkbox"/>
	Руководитель	TEXT(20)	<input type="checkbox"/>	<input type="checkbox"/>

	Physical Name	Data Type	Req'd	PK
	Таб. номер	CHAR(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	ФИО	TEXT(20)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	Должность	CHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>

Рис. 4.6. Структура сущностей «Кабинеты», «Единицы оборудования», «Подразделения» и «Сотрудники»

Размещение сущностей на диаграмме

В соответствии с содержательным описанием на листе MS Visio размещаем все пять сущностей.

После размещения сущностей с помощью объекта Relationship выполняется связывание. Важно - стрелка связи должна указывать на главную сущность. Графический объект типа Relationship представляет собой динамический соединитель (connector) и поэтому при установлении связи нужно, чтобы он зафиксировал связь с графическим объектом типа Сущность. Факт фиксирования легко определить по появлению красной рамки вокруг объекта типа сущность.

Разместите все пять таблиц на листе этажами так, чтобы подчиненность сущностей легко читалась (рис. 4.7).

Обратите внимание на то, что мы в подчиненные сущности специально не вводили атрибуты, которые являются внешними ключами. Это нельзя делать потому, что после выполнения соединения Visio сам автоматически добавляет эти поля и обозначает их как FK – внешний ключ. Например, у сущности Сотрудники появился атрибут ID_подразделения, а у сущности Единицы оборудования появилось сразу три внешних ключа.

На этом разработка ER-модели считается завершенной.

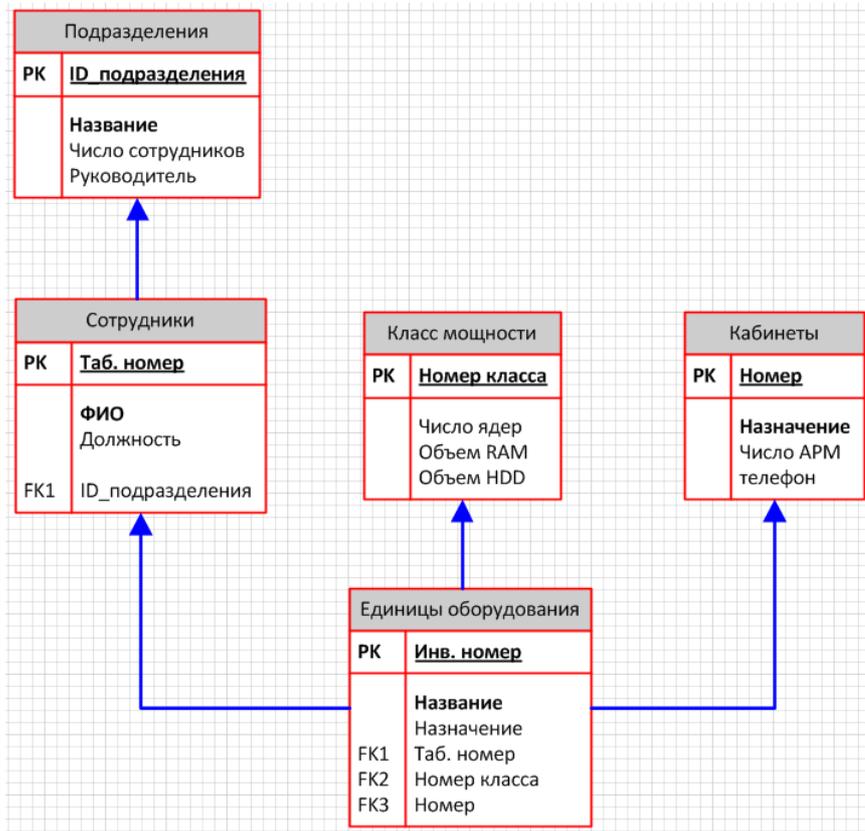


Рис. 4.7. ER-модель в окончательном виде