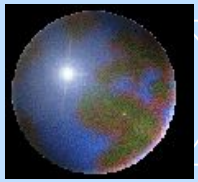
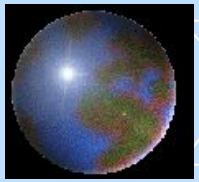


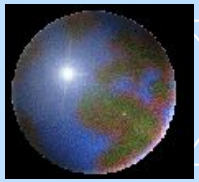
CASE-технологии

Тема 6



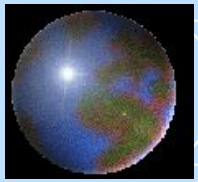
CASE (Computer Aided Software Engineering)

- CASE– технология представляет собой совокупность методов проектирования ПО, а также набор инструментальных средств, позволяющих в наглядной форме моделировать предметную область, анализировать эту модель на всех стадиях разработки и сопровождения ПО и разрабатывать приложения в соответствии с информационными потребностями пользователя



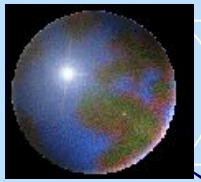
Бизнес-модель позволит решить задачи, стоящих перед предприятием

- задачи реорганизации бизнеса, обусловленной переходом от функциональной индустриальной модели к процессной;
- задачи применения информационных систем для управления бизнесом, обусловленной бурным ростом современных информационных технологий;
- сертификации бизнеса с применением комплекса стандартов серии ISO 9000, обусловленной повышением требований к качеству товаров и услуг.



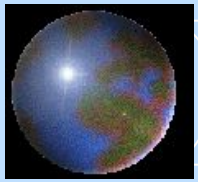
Детальная бизнес-модель позволит:

- описать, "увидеть" и скорректировать будущую систему до того, как она будет реализована физически;
- уменьшить затраты на создание системы;
- оценить работы по времени и результатам;
- достичь взаимопонимания между всеми участниками проекта;
- улучшить качество создаваемой системы.

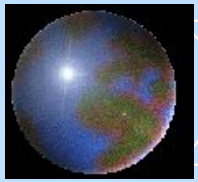


Значение моделей

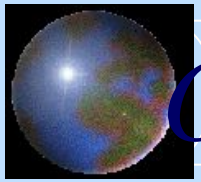
- модели позволяют осуществлять автоматизированное и быстрое обучение новых работников конкретному направлению деятельности предприятия;
- с помощью моделей можно осуществлять предварительное моделирование нового направления деятельности с целью выявления новых потоков данных, взаимодействующих подсистем и бизнес-процессов.



- Функционально-модульный (структурный)
- Объектно-ориентированный

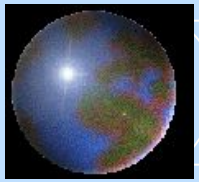


- **Функционально-модульный** или **структурный**. В его основу положен принцип функциональной декомпозиции, при которой структура системы описывается в терминах иерархии ее функций и передачи информации между отдельными функциональными элементами.



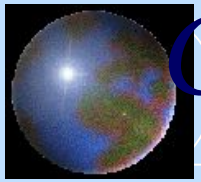
Общие принципы :

- принцип «разделяй и властвуй»;
- принцип иерархического упорядочения.



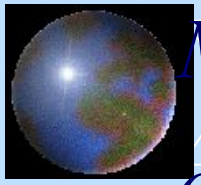
Для методов структурного анализа характерно

- разбиение на уровни абстракции с ограничением числа элементов на каждом из уровней (обычно от 3 до 6-7) ;
- ограниченный контекст, включающий лишь существенные на каждом уровне детали;
- использование строгих формальных правил записи;
- последовательное приближение к конечному результату.



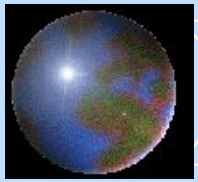
Структурные диаграммы иллюстрируют

- функции, которые система должна выполнять;
- отношения между данными;
- динамическое поведение системы (аспекты реального времени).

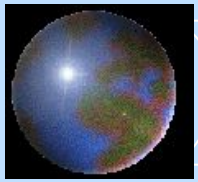


Модели, описывающие функциональную структуру системы :

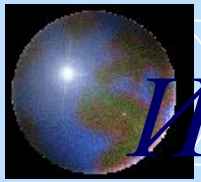
- DFD(Data Flow Diagrams) - диаграммы потоков данных;
- SADT(Structured Analysis and Design technique — метод структурного анализа и проектирования,) - модели и соответствующие функциональные диаграммы;
- ERD (Entity-Relationship Diagrams) диаграммы «сущность-связь»;
- STD (State Transition Diagrams) - диаграммы переходов состояний;
- структурные схемы (карты).



- **Объектно-ориентированный** подход использует объектную декомпозицию. При этом структура системы описывается в терминах объектов и связей между ними, а поведение системы описывается в терминах обмена сообщениями между объектами.

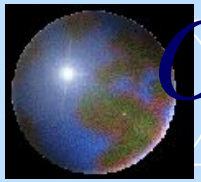


- Метод SADT представляет собой совокупность правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. Функциональная модель SADT отображает функциональную структуру объекта т.е. производимые им действия и связи между этими действиям.



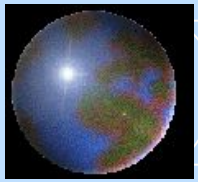
История создания

- Метод SADT разработан Дугласом Россом (SoftTech, Inc.) в 1973т.
- Метод SADT поддерживается Министерством обороны США, которое было инициатором разработки стандарта IDEFO (Icam DEFinition) - подмножества SADT, являющегося основной частью программы ICAM (Integrated Computer Aided Manufacturing - интегрированная компьютеризация производства), проводимой по инициативе ВВС США. IDEFO был утвержден в качестве федерального стандарта США.



Основные элементы этого метода основываются на следующих концепциях:

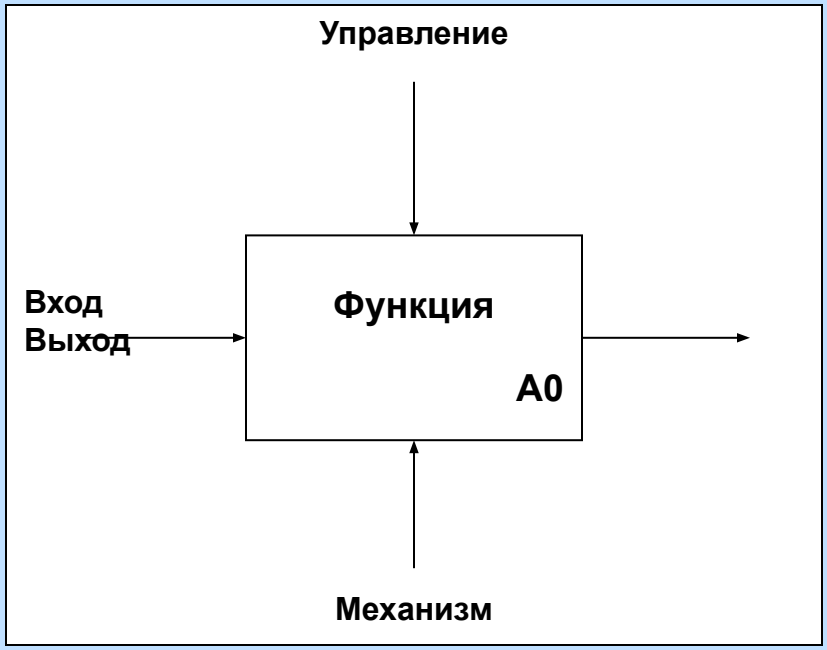
- графическое представление блочного моделирования.;
- строгость и точность;
- отделение организации от функции.



СОСТАВ ФУНКЦИОНАЛЬНОЙ МОДЕЛИ

- Результатом применения метода SADT является модель, которая состоит из диаграмм, фрагментов текстов и глоссария, имеющих ссылки друг на друга.
- Диаграммы — главные компоненты модели, все функции организации и интерфейсы на них представлены как блоки и дуги соответственно.
- Место соединения дуги с блоком определяет тип интерфейса.

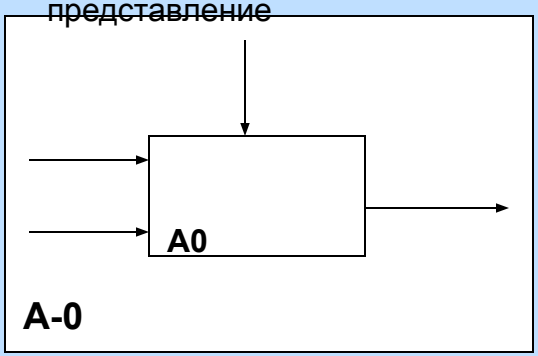
Нотация SADT

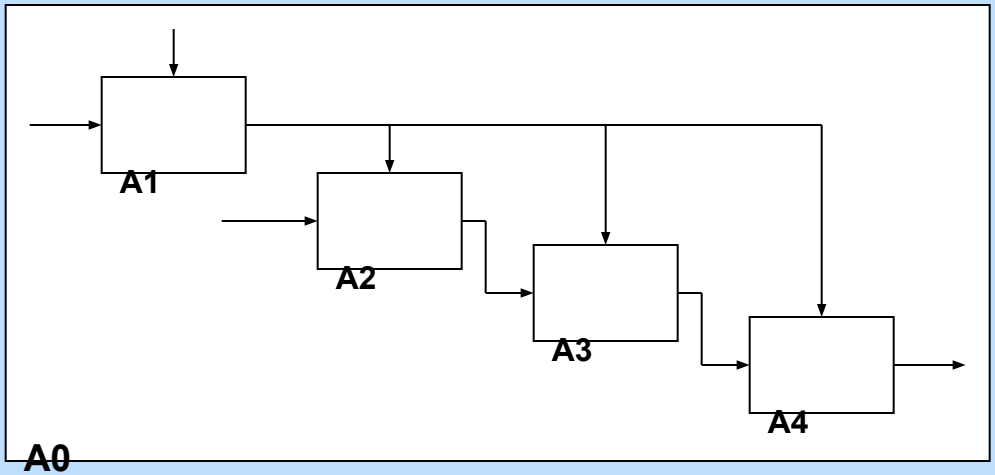


- Управляющая информация (управление)входит сверху
- Входная информация или объекты (вход), которые подвергаются обработке, показаны слева
- Результаты (выход) показаны справа
- Механизм (человек или автоматизированная система), который осуществляет операцию, входит снизу

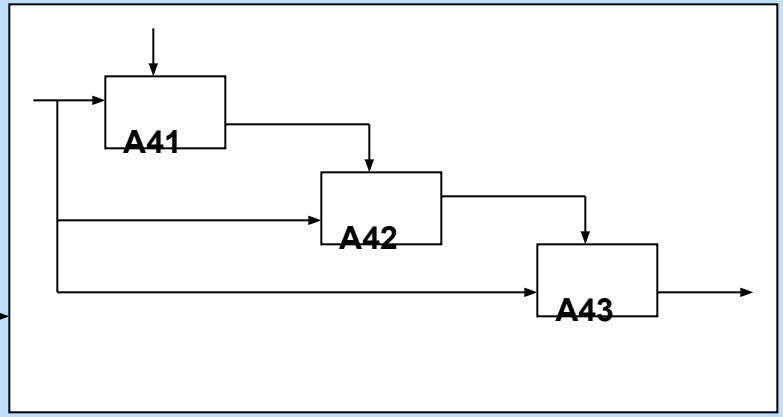
Более детальное представление

Общее представление



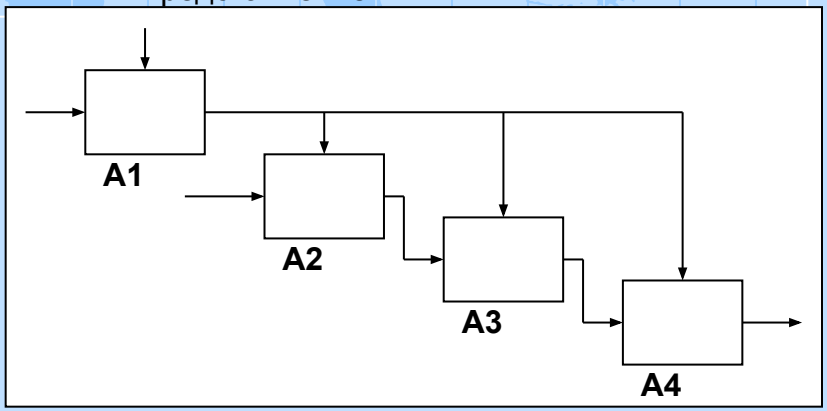


Верхняя диаграмма является родительской для нижней диаграммы



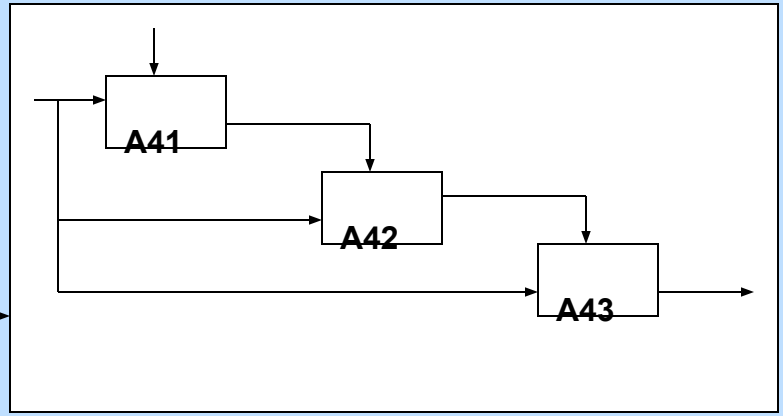
- Одной из наиболее важных особенностей метода SADT является постепенное введение все больших уровней детализации по мере создания диаграмм, отображающих модель.
- Каждый компонент модели может быть декомпозирован на другой диаграмме. Каждая диаграмма иллюстрирует «внутреннее строение" блока на родительской диаграмме.

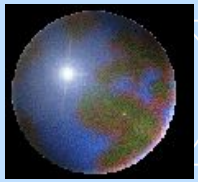
Более детальное представление



A0

Верхняя диаграмма является родительской для нижней диаграммы

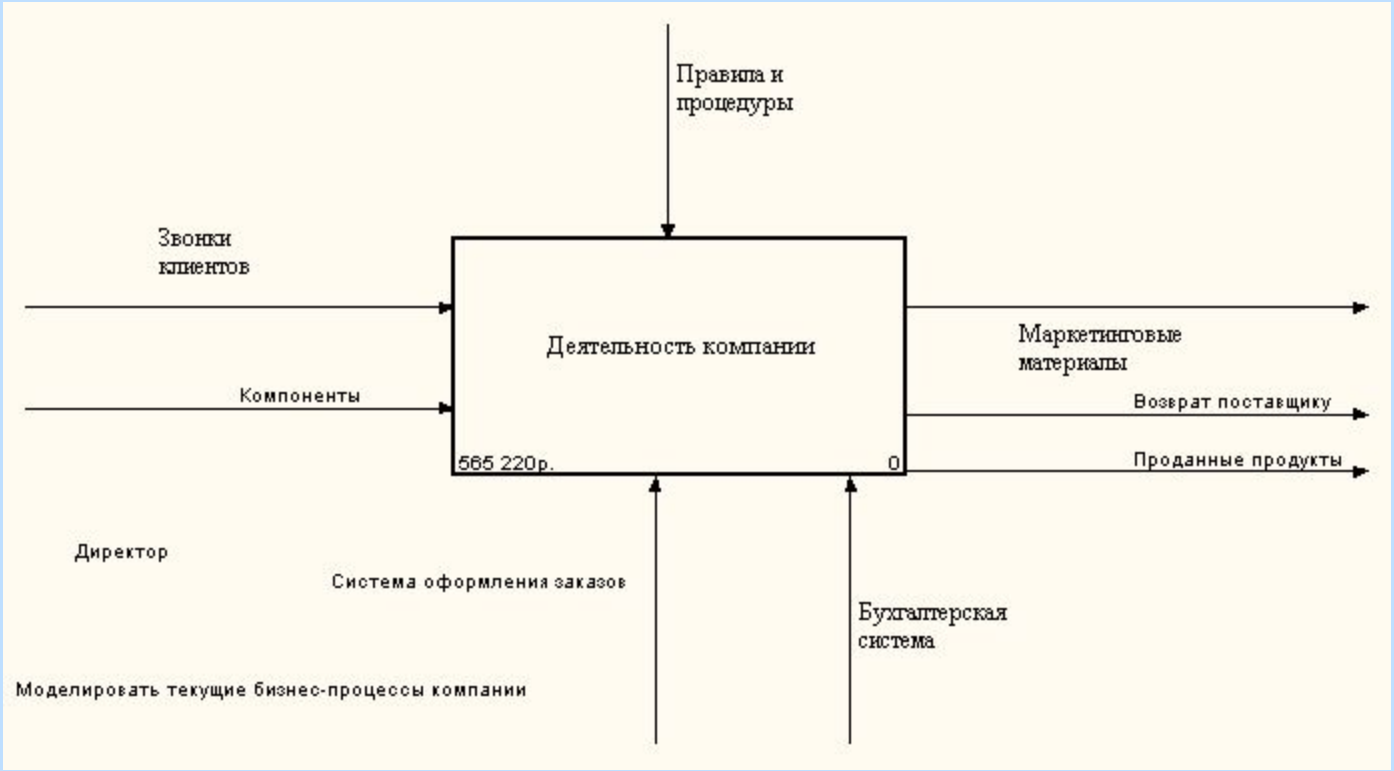


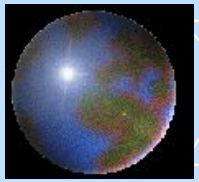


ПОСТРОЕНИЕ ИЕРАРХИИ ДИАГРАММ

1. Построение SADT-модели начинается с представления всей системы в виде простейшего компонента - одного блока и дуг, изображающих интерфейсы с функциями вне системы.

- Имя, указанной в блоке, является общим.
- Интерфейсные дуги соответствуют полному набору внешних интерфейсов системы в целом.

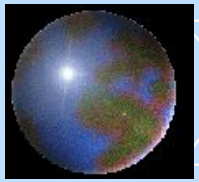

Правила и
процедуры
Звонки
клиентов
Деятельность компании
Маркетинговые
материалы
Компоненты
Возврат поставщику
565 220р.
0
Проданные продукты
Директор
Система оформления заказов
Бухгалтерская
система
Моделировать текущие бизнес-процессы компании



ПОСТРОЕНИЕ ИЕРАРХИИ ДИАГРАММ

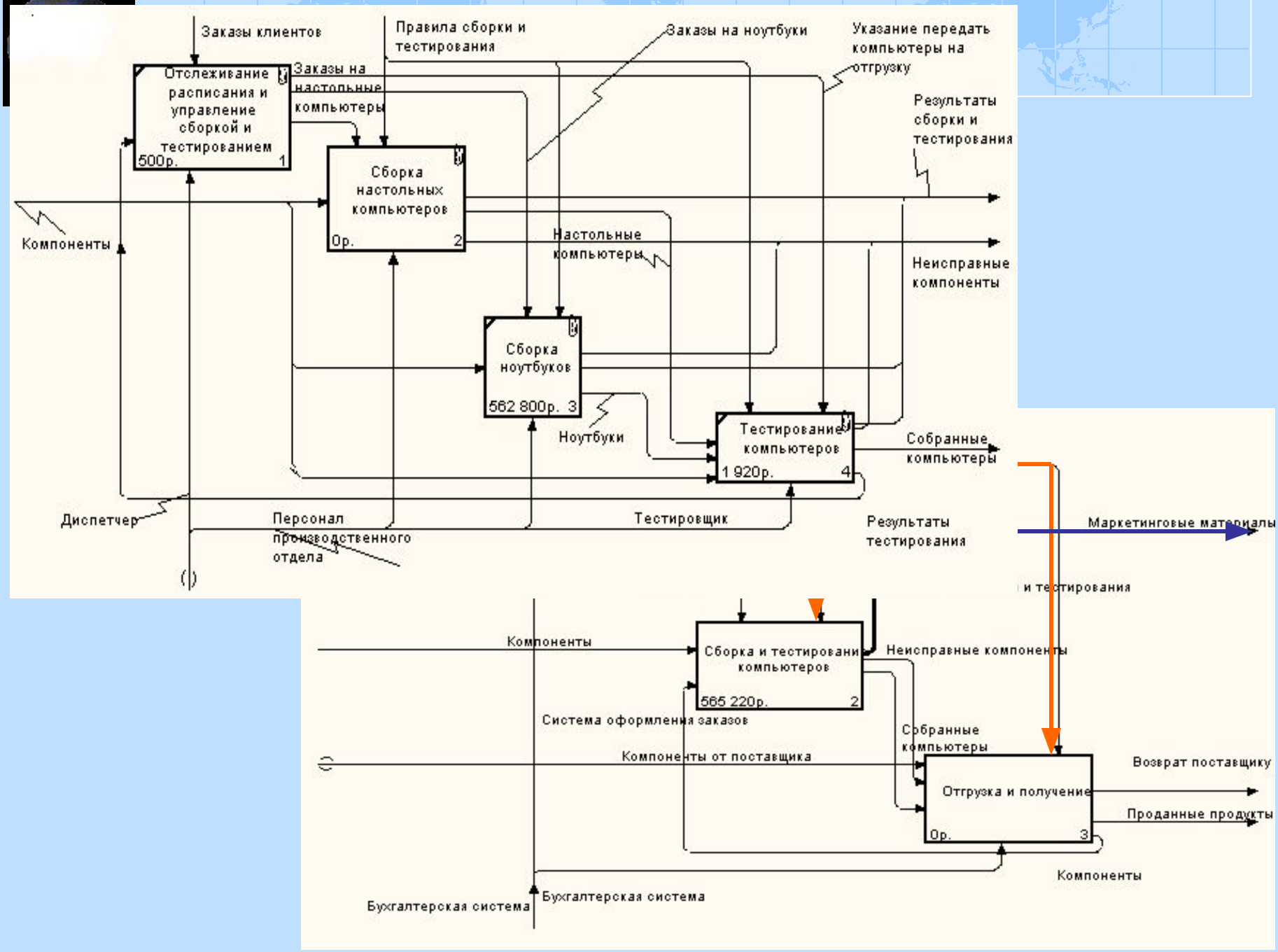
2. Блок, который представляет систему в качестве единого модуля, детализируется на другой диаграмме с помощью нескольких блоков, соединенных интерфейсными дугами.

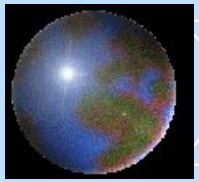
- Эти блоки определяют основные подфункции исходной функции.
- Каждая из этих подфункций может быть декомпозирована подобным образом в целях большей детализации.



ПОСТРОЕНИЕ ИЕРАРХИИ ДИАГРАММ

- Модель SADT представляет собой серию диаграмм с сопроводительной документацией, разбивающих сложный объект на составные части, которые изображены в виде блоков.
- На каждом шаге декомпозиции диаграмма предыдущего уровня называется родительской для более детальной диаграммы.

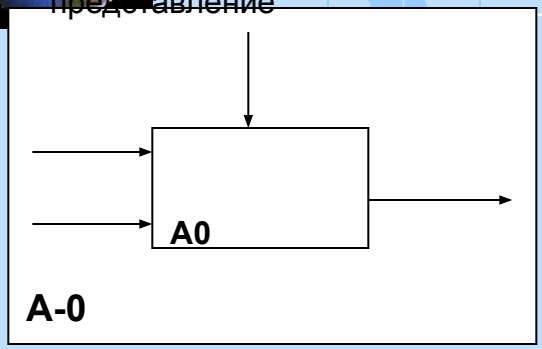

Заказы клиентов
Правила сборки и тестирования
Заказы на ноутбуки
Указание передать компьютеры на отгрузку
Отслеживание расписания и управление сборкой и тестированием
500р.
1
Заказы на настольные компьютеры
Сборка настольных компьютеров
0р.
2
Результаты сборки и тестирования
Компоненты
Настольные компьютеры
Неисправные компоненты
Сборка ноутбуков
562 800р.
3
Ноутбуки
Тестирование компьютеров
1 920р.
4
Собранные компьютеры
Диспетчер
Персонал производственного отдела
Тестировщик
Результаты тестирования
Маркетинговые материалы
и тестирования
Компоненты
Сборка и тестирование компьютеров
565 220р.
2
Неисправные компоненты
Система оформления заказов
Собранные компьютеры
Компоненты от поставщика
Возврат поставщику
Отгрузка и получение
0р.
3
Проданные продукты
Компоненты
Бухгалтерская система
Бухгалтерская система



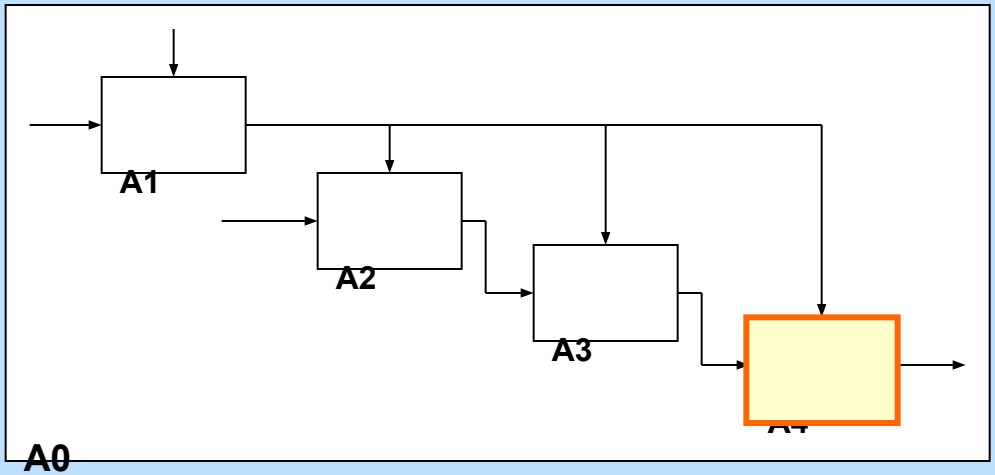
ПОСТРОЕНИЕ ИЕРАРХИИ ДИАГРАММ

- Дуги, входящие в блок и выходящие из него на диаграмме верхнего уровня, являются точно теми же самыми, что и дуги, входящие в диаграмму нижнего уровня и выходящие из нее, потому что блок и диаграммы изображают одну и ту же часть системы.

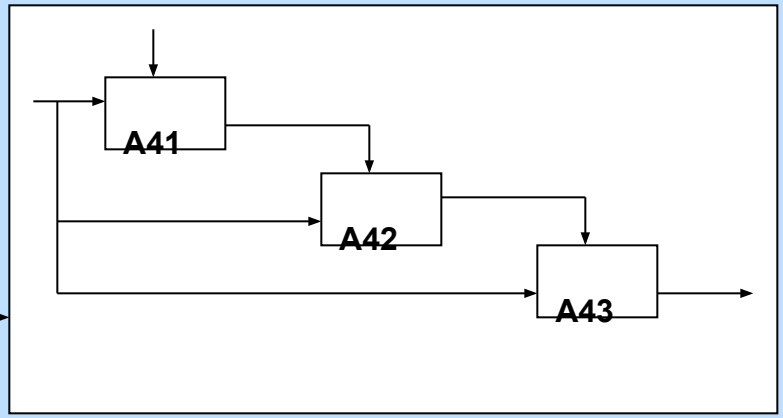
Общее представление



Более детальное представление

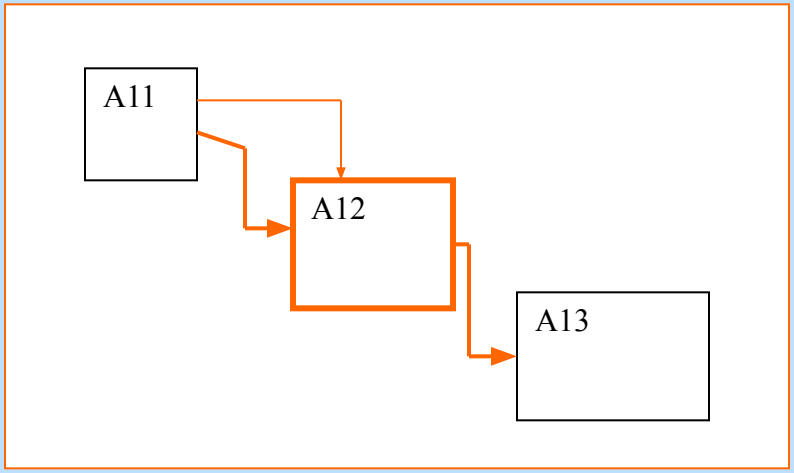


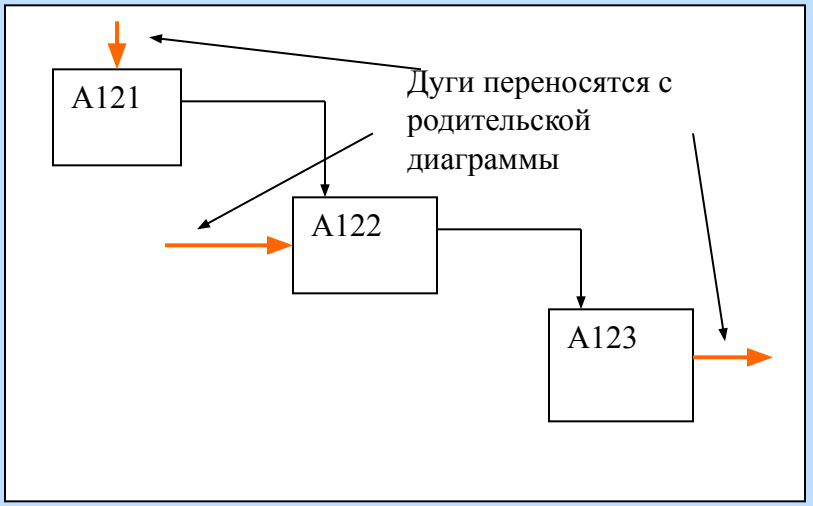
Верхняя диаграмма является родительской для нижней диаграммы



ПОСТРОЕНИЕ ИЕРАРХИИ ДИАГРАММ

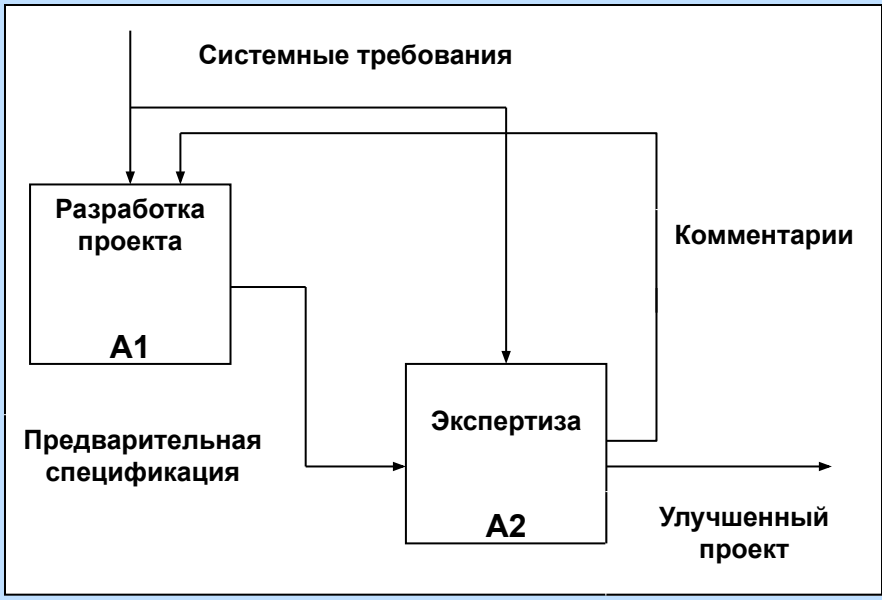
- Неприсоединенным дуги соответствуют входам, управлениям и выходам родительского блока.
- Источник или получатель пограничных дуг может быть обнаружен только на родительской диаграмме.
- Неприсоединенные концы должны соответствовать дугам на исходной диаграмме.
- Все граничные дуги должны продолжаться на родительской диаграмме, чтобы она была полной и непротиворечивой.



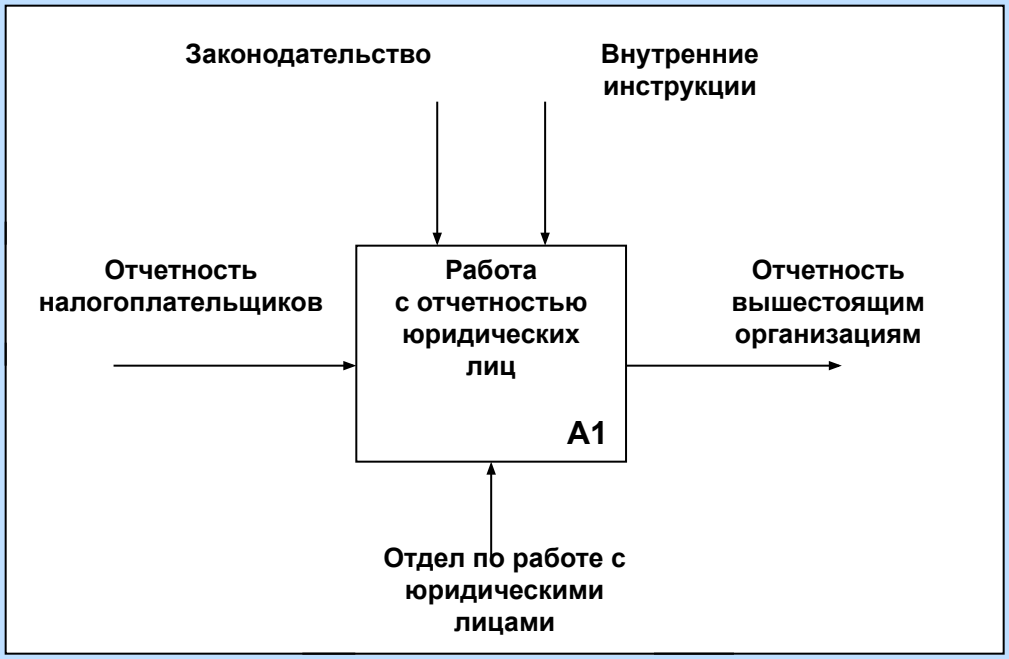


ПОСТРОЕНИЕ ИЕРАРХИИ ДИАГРАММ

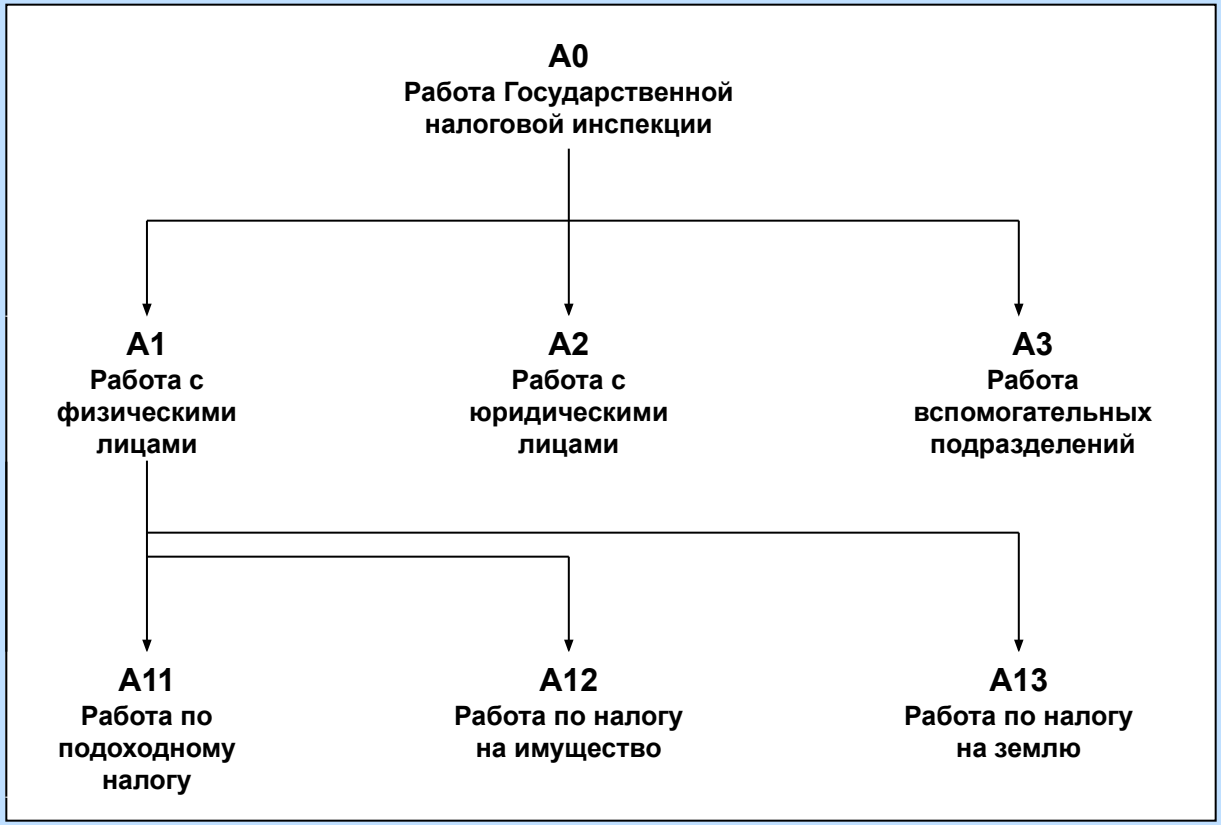
- На SADT-диаграммах не указаны явно ни последовательность, ни время.
- Обратные связи, итерации, продолжающиеся процессы и перекрывающиеся (по времени) функции могут быть изображены с помощью дуг.
- Обратные связи могуr выступать в вида комментариев, замечаний, исправлений и т. д

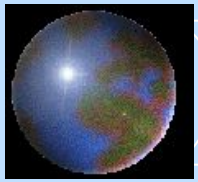


Пример бизнес-процесса



Пример дерева диаграмм



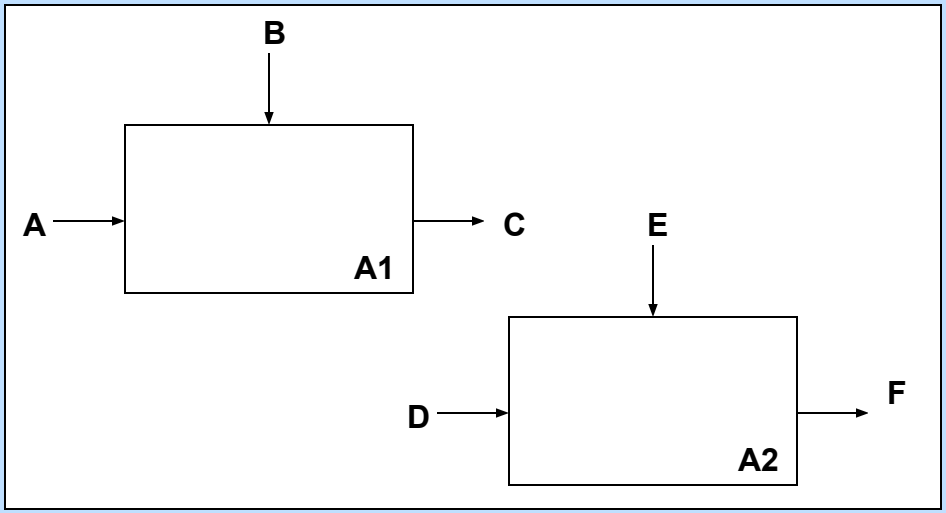


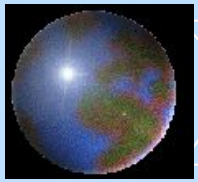
Типы связей между функциями:

- случайная;
- логическая;
- временная;
- процедурная;
- коммуникационная;
- последовательная;
- функциональная.

Типы связей

- Случайная связь — показывает, что конкретная связь между функциями незначительна или полностью отсутствует. Это относится к ситуации, когда имена данных на SADT-дугах в одной диаграмме имеют слабую связь друг с другом.



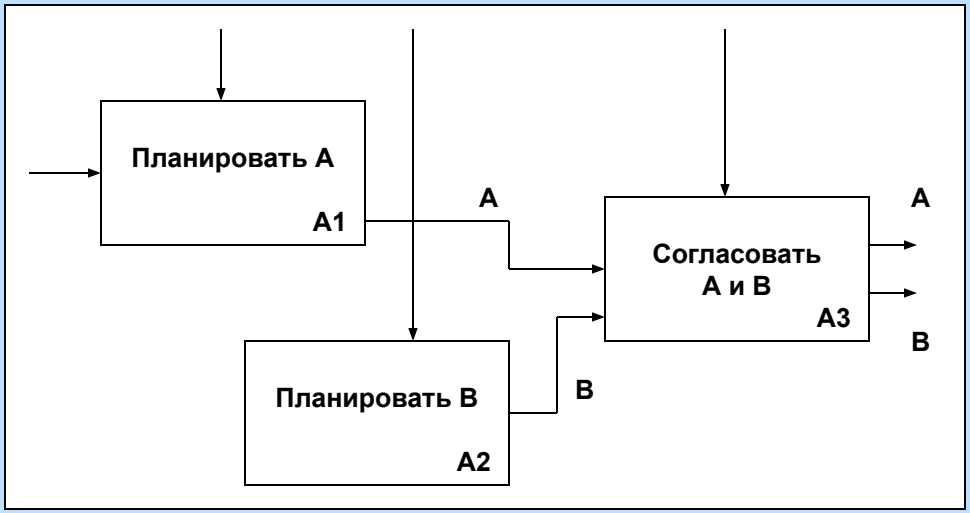


Типы связей

- Логическая связь – данные и функции собираются вместе благодаря тому, что они попадают в общий класс или набор элементов, но необходимых функциональных отношений между ними не обнаруживается.
- Временная связь – представляет функции, связанные во времени, когда данные используются одновременно или функции включаются параллельно, а не последовательно.

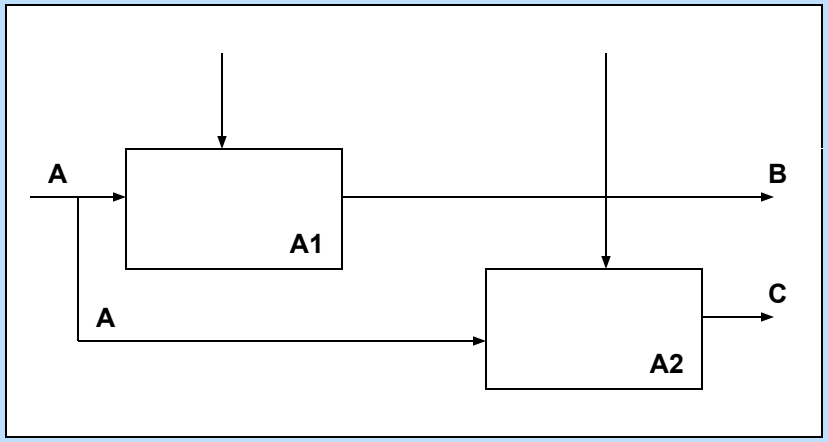
Типы связей

- Процедурная связь - функции сгруппированы вместе благодаря тому, что они выполняются в течение одной и той же части цикла или процесса.



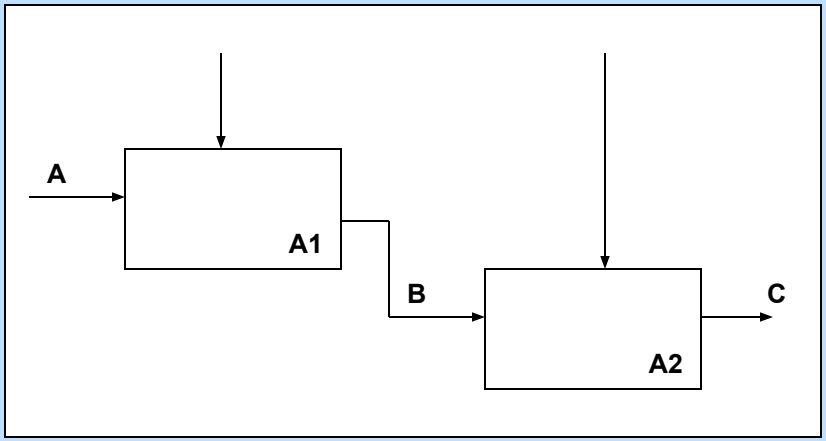
Типы связей

- Коммуникационная - функции группируются благодаря тому, что они используют одни и те же входные данные и/или производят одни и те же выходные данные



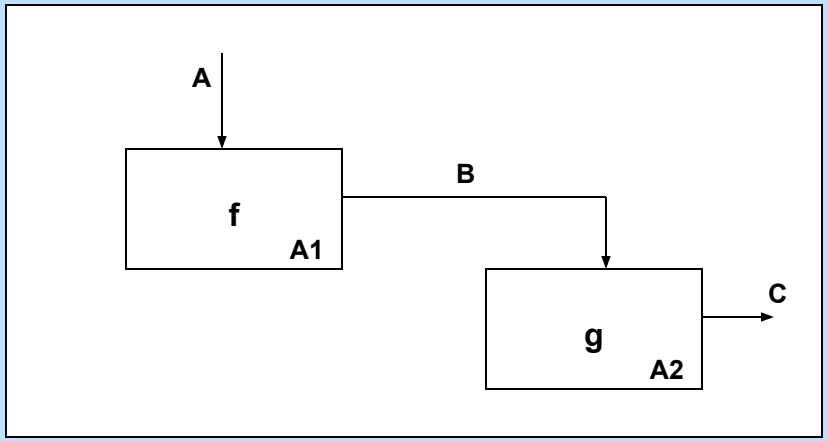
Типы связей

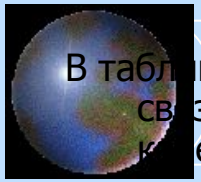
- Последовательная связь - выход одной функции служит входными данными для следующей функции. Связь между элементами на диаграмме является более тесной, чем в рассмотренных выше случаях, поскольку моделируются причинно-следственные зависимости



Типы связей

- Функциональная связь – все элементы функции влияют на выполнение одной и только одной функции. Одним из способов определения функционально-связанных диаграмм является рассмотрение двух блоков, связанных через управляющие дуги.
- В математических терминах необходимое условие для простейшего типа функциональной связи имеет следующий вид: C=g(B)=g(f(A))



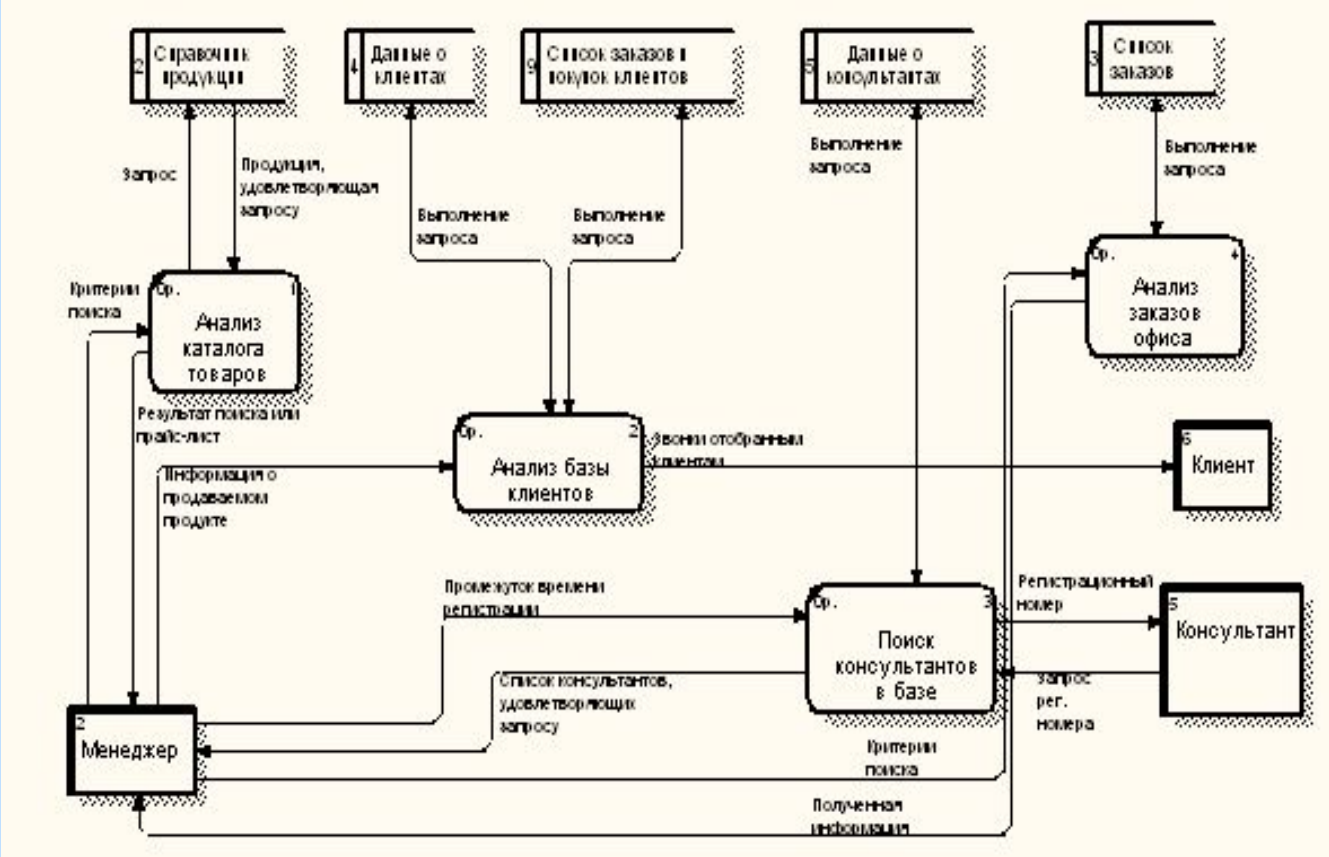


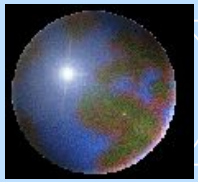
В таблице представлены все типы связей, рассмотренные выше. Уровни 4-6 устанавливают типы связей, которые разработчики считают важнейшими для получения диаграмм хорошего качества.

Уровень значимости	*Тип связи*	*Характеристика типа связи*	
		Для функций	**Для данных**
0	Случайная	Случайная	Случайная
1	Логическая	Функции одного и того же множества или типа (например, «редактировать все входы»)	Данные одного и того же множества или типа.
2	Временная	Функции одного и того же периода времени (например, "операции инициализации")	Данные, используемые в каком-либо временном интервале
3	Процедурная	Функции, работающие в одной и той же фазе или итераций (например, "первый проход компилятора")	Данные, используемые во время одной и той же фазы или итерации
4	Коммуникационная	Функции, использующие одни и те же данные	Данные, на которые воздействует одна и та же деятельность
5	Последовательная	Функции, выполняющие последовательные преобразования одних и тех же данных	Данные, преобразуемые последовательными функциями
6	Функциональная	Функции, объединяемые для выполнения одной функции	Данные, связанные с одной функцией

Диаграмма потоков данных

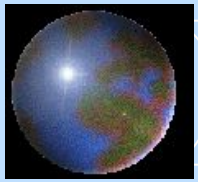
- DFD





Определение

- Диаграммы потоков данных (DFD)— демонстрирует, как каждый процесс преобразует свои входные данные в выходные и отношения между этими процессами.
- Модель системы определяется как иерархия диаграмм потоков данных, описывающих асинхронный процесс преобразования информации от ее входа в систему до выдачи пользователю.



Состав диаграмм потоков данных

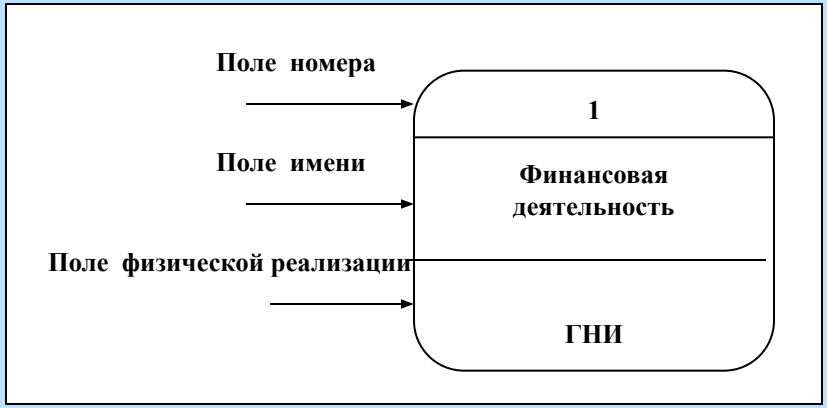
- внешние сущности;
- системы и подсистемы;
- процессы;
- накопители данных;
- потоки данных;
- информационный канал.

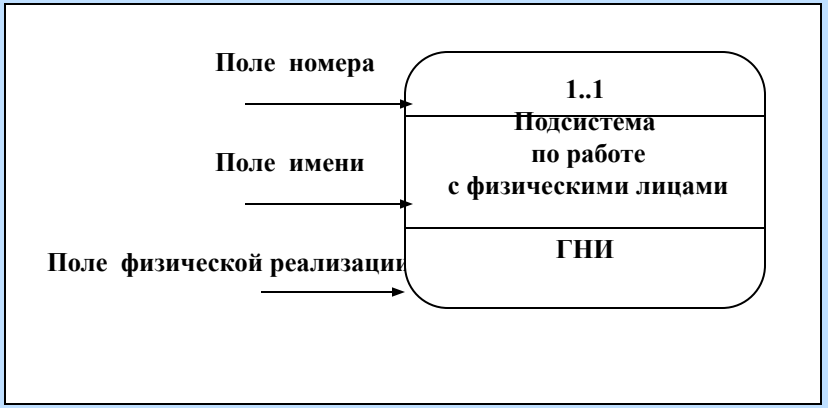
DFD описывает:

- представляет **внешние сущности**, представляющие собой источник или приемник информации, например заказчики, персонал, поставщики, клиенты, склад, которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы;

DFD описывает:

- **функции** обработки информации (**процессы**);





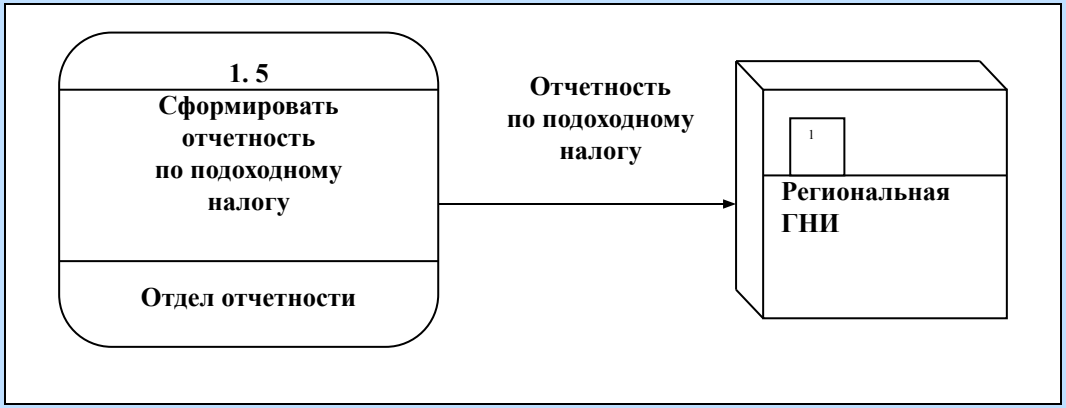
DFD описывает:

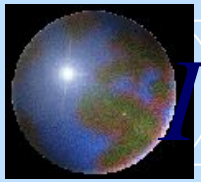
- **накопители данных**, это абстрактное устройство для хранения информации (справочники, документы, отчеты), которую можно в любой момент поместить в накопитель и, через некоторое время извлечь, причем способы помещения и извлечения могут быть любыми.

D1	**Реестр налогоплательщиков**

DFD описывает:

- определяет **потоки данных** (документы) являющиеся результатом работ или поступающие в систему извне;





Порядок построения

- Построение контекстной диаграммы
- Построение диаграмм первого уровня заключается в декомпозиции системы (подсистем), которые присутствует на контекстной диаграмме.
- Каждое, событие представляется в виде процесса с соответствующими входными и выходными потоками, накопителями данных, внешними сущностями и ссылками на другие процессы.

- Фактически спецификации представляют собой описания алгоритмов задач, выполняемых процессами. Спецификации содержат номер и/или имя процесса, списки входных и выходных данных и тело (описание) процесса, являющееся спецификацией алгоритма или операции, трансформирующей входные потоки данных в выходные.

Процесс: *Формирование сводного плана кафедры.*

Вход: Индивидуальные планы .

Выход: Сводный план.

Алгоритм:

1. Выбрать из накопителя Индивидуальные планы преподавателей
2. Сформировать «Сводный план кафедры» на основе данных по индивидуальным планам преподавателей» по следующим разделам:

Учебная работа;

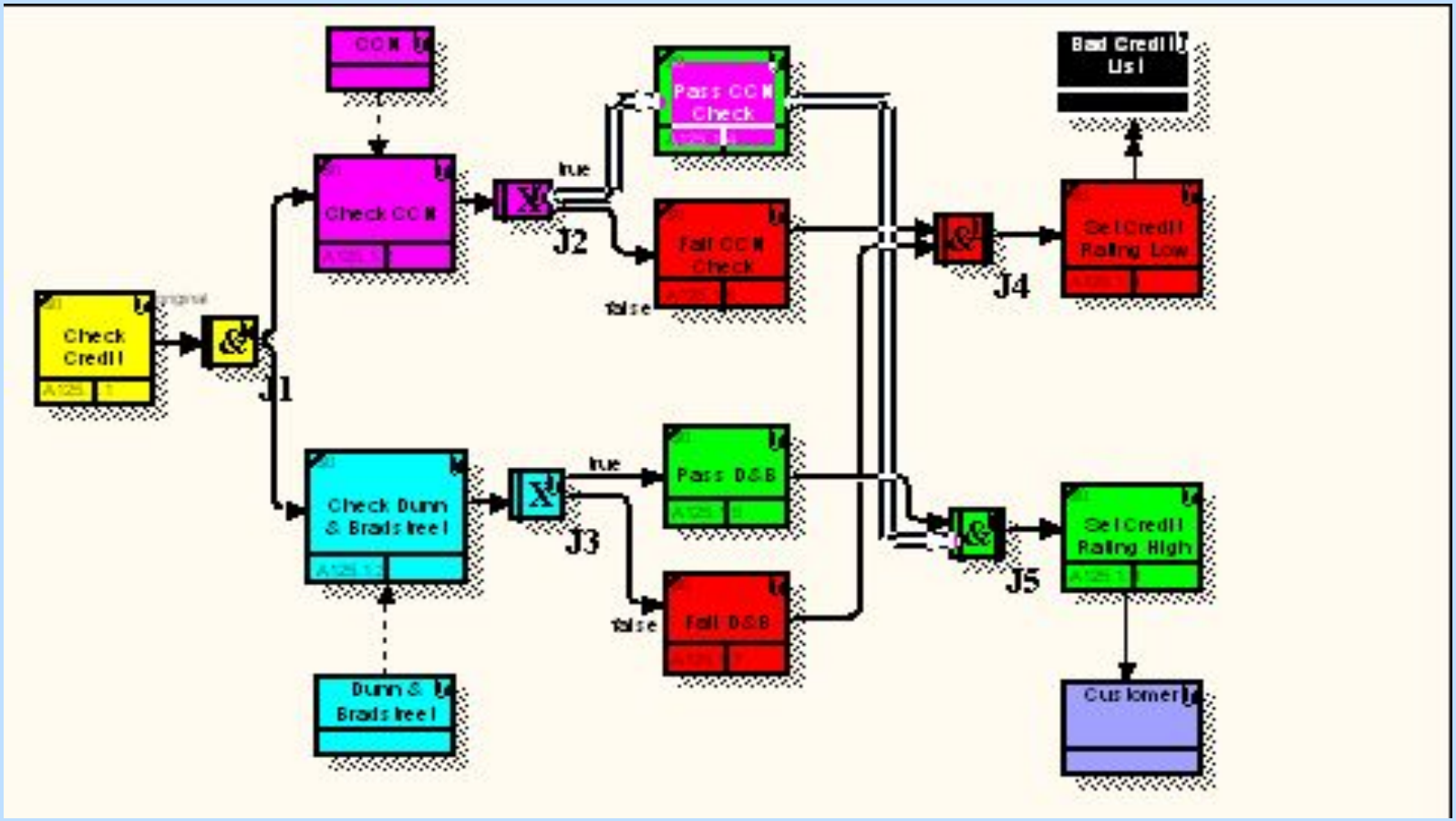
Методическая работа;

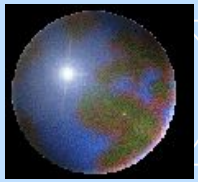
Научная работа.

3.Поместить документ в накопитель

Описание процессов

- Диаграмма IDEF3



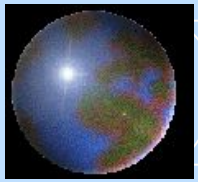


IDEF3 – это метод,

- Основная цель которого дать возможность аналитикам описать ситуацию, когда процессы выполняются в определенном последовательности, а также описать объекты, участвующие совместно в одном процессе.

Единицы работы (Unit of Work)

- Работа, является центральным компонентом модели

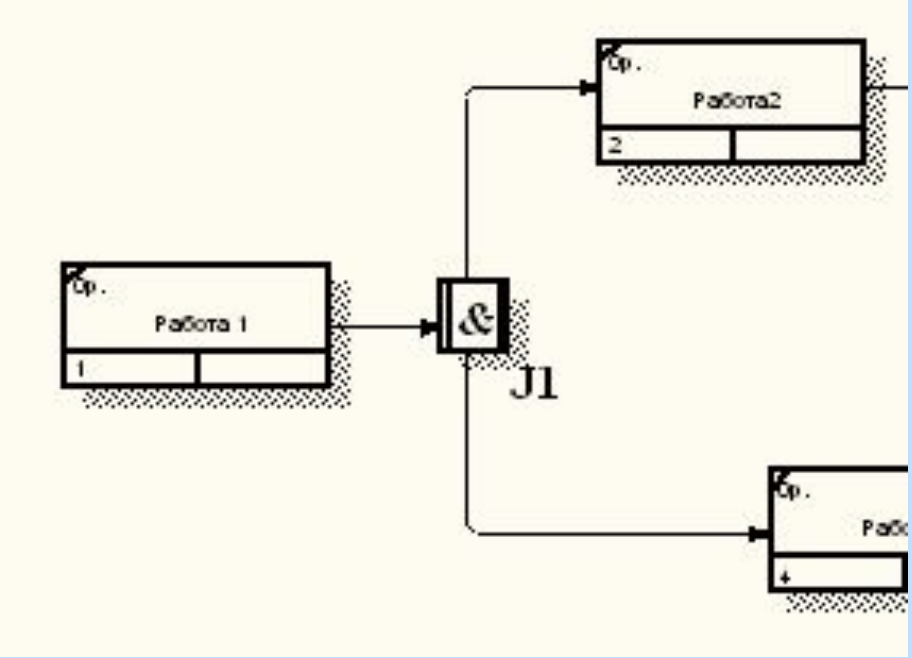


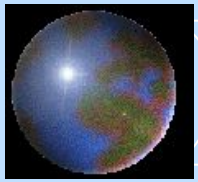
Работа имеет ассоциированный документ, который включает текстовое описание компонентов работы (Activity Properties\ UOW):

- Objects (объекты);
- Facts(факты);
- Constraints(ограничения);
- Description(дополнительное описание).

Связи

- Показывают взаимоотношение работ

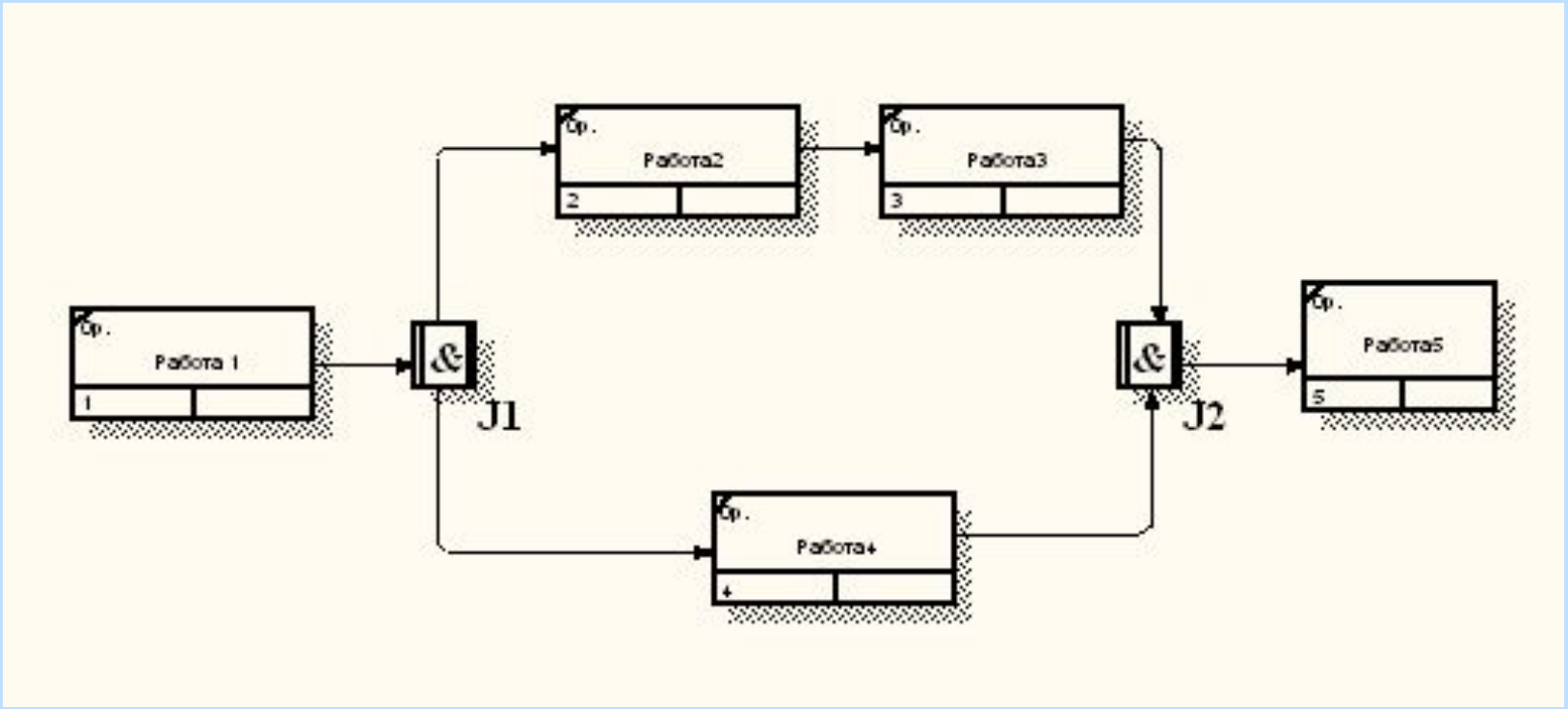




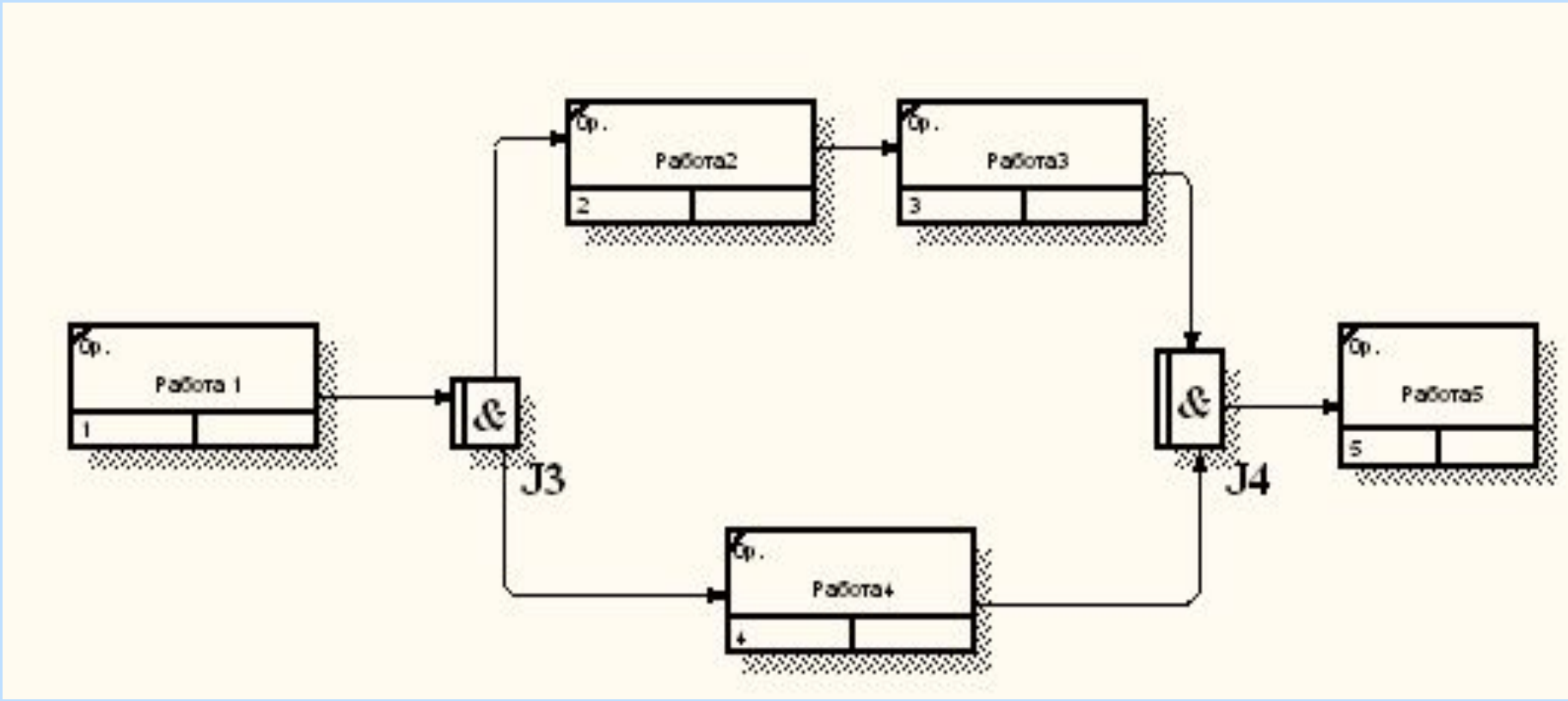
Перекрестки

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

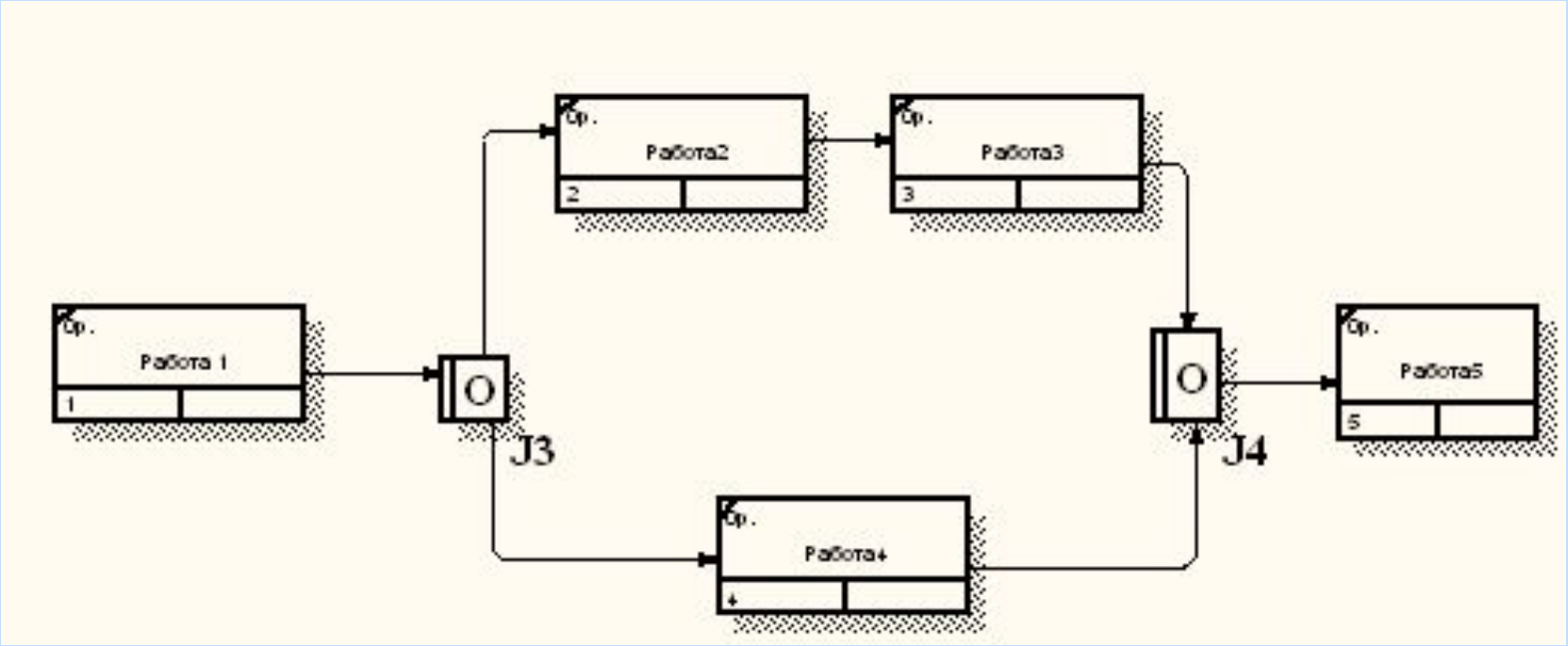
Синхронное «И»



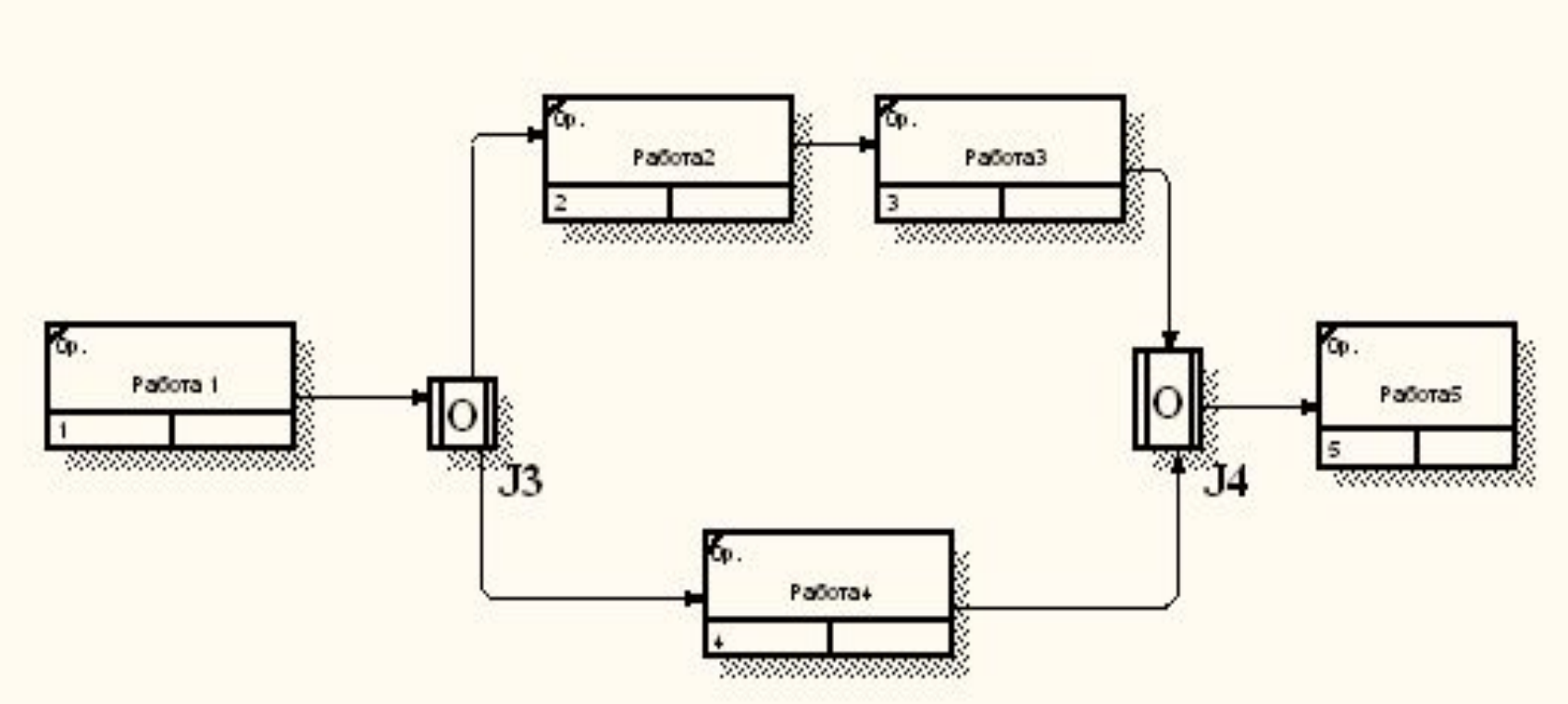
Асинхронное «И»



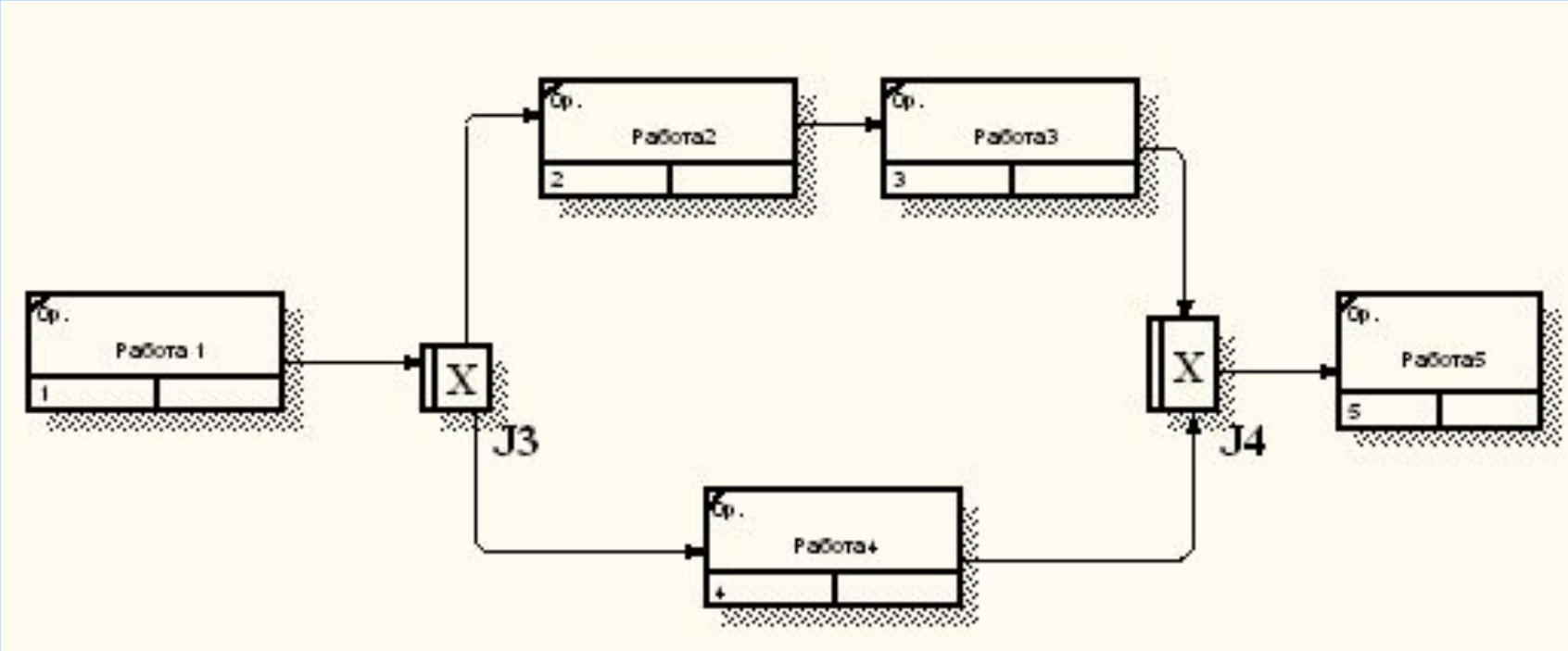
Асинхронное «ИЛИ»



Синхронное «ИЛИ»

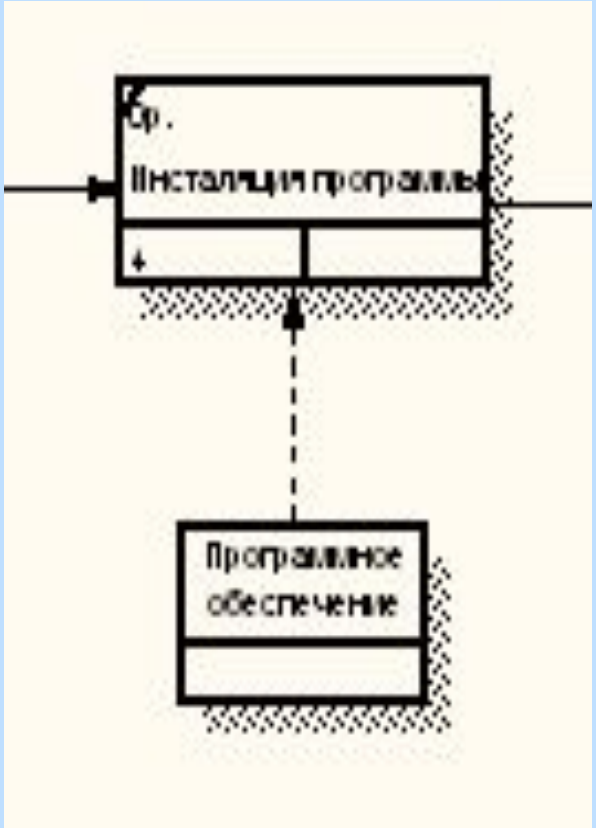


Исключающее «ИЛИ»

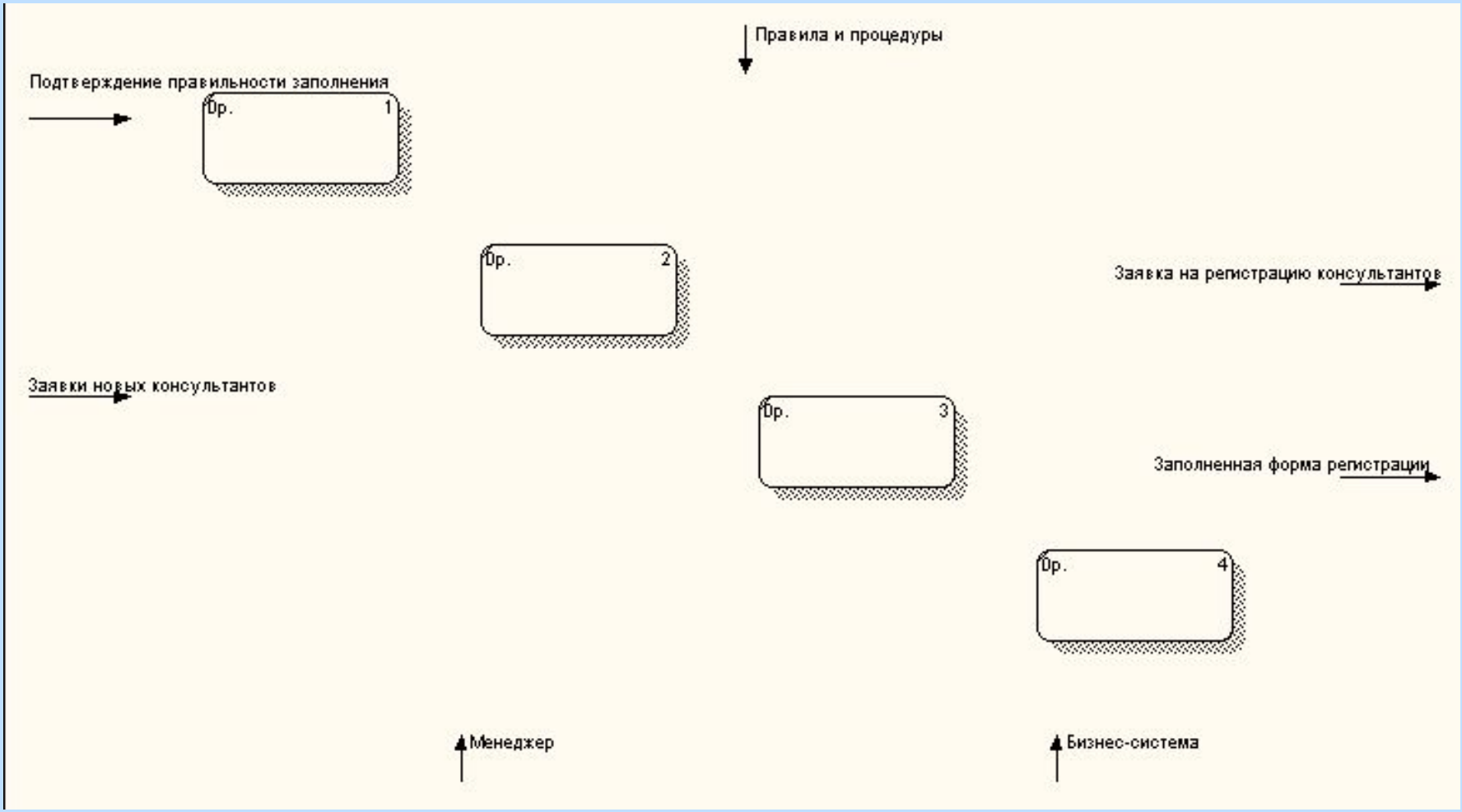


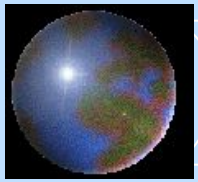
Объект ссылки

- Выражает некую идею, концепцию или данные, которые нельзя связать со стрелкой, перекрестком или работой.



Декомпозиция IDEF0





Декомпозиция IDEF0 в диаграмму DFD

- Удалить все граничные стрелки на DFD; создать соответствующие внешние сущности и хранилища данных;
- Создать внутренние стрелки, начинающиеся с внешних сущностей вместо граничных стрелок;
- Стрелки на диаграмме IDEF0 затоннелировать.