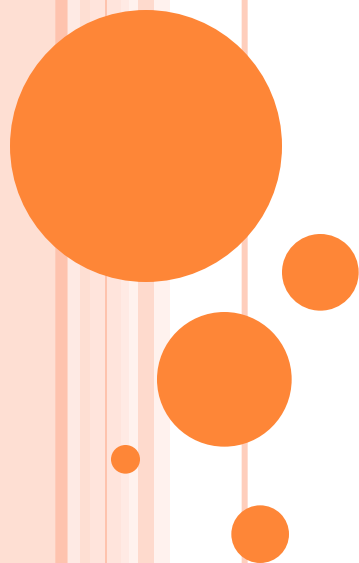
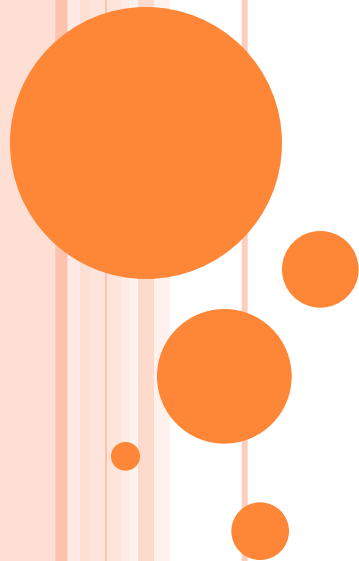


МЕТОДОЛОГИИ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ



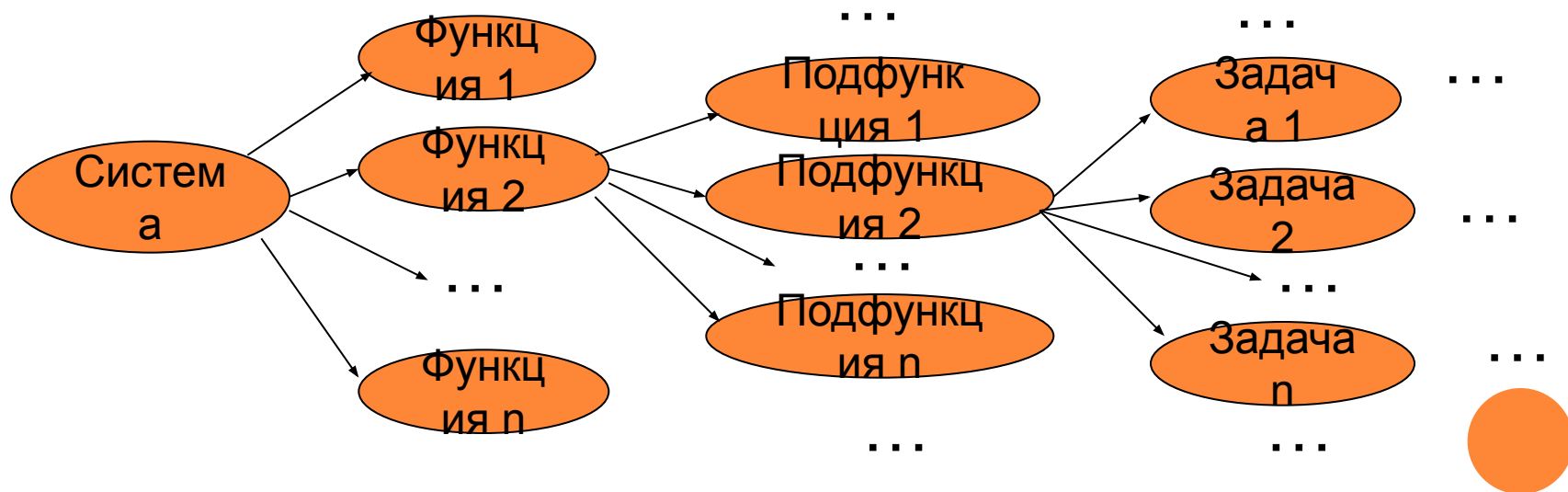
3.1 МЕТОДОЛОГИЯ СТРУКТУРНОГО МОДЕЛИРОВАНИЯ SADT

- **Сущность структурного подхода**
- **Основные принципы структурного подхода**



СУЩНОСТЬ СТРУКТУРНОГО ПОДХОДА К МОДЕЛИРОВАНИЮ СИСТЕМ

Система разбивается на функциональные подсистемы, которые, в свою очередь, делятся на подфункции, подфункции – на задачи и т.д. до конкретных процедур



КЛЮЧЕВЫЕ ПОНЯТИЯ СТРУКТУРНОГО АНАЛИЗА

Структурным анализом принято называть метод исследования системы, которое начинается с ее общего обзора, а затем детализируется, приобретая иерархическую структуру с все большим числом уровней.

- ▣ **Функция** – совокупность операций, сгруппированных по определенному признаку.
- ▣ **Бизнес-процесс** — связанная совокупность функций, в ходе выполнения которой потребляются определенные ресурсы и создается продукт (предмет, услуга, научное открытие, идея), представляющая ценность для потребителя.
- ▣ **Подпроцесс** – это бизнес-процесс, являющийся структурным элементом некоторого бизнес-процесса и представляющий ценность для потребителя.
- ▣ **Бизнес-модель** – структурированное графическое описание сети процессов и операций, связанных с данными, документами, организационными единицами и прочими объектами, отражающими существующую или предполагаемую деятельность предприятия.

БАЗОВЫЕ ПРИНЦИПЫ СТРУКТУРНОГО ПОДХОДА

- ▣ принцип "разделяй и властвуй" – принцип решения сложных проблем путем их разбиения на множество меньших независимых задач, легких для понимания и решения;
- ▣ принцип иерархического упорядочивания – принцип организации составных частей проблемы в иерархические древовидные структуры с добавлением новых деталей на каждом уровне.



ОСНОВНЫЕ ПРИНЦИПЫ СТРУКТУРНОГО ПОДХОДА

- ▣ **Принцип абстрагирования** – выделение существенных с некоторых позиций аспектов системы и отвлечении от несущественных с целью представления проблемы в простом общем виде.
- ▣ **Принцип формализации** – необходимость строгого методического подхода к решению проблемы.
- ▣ **Принцип упрятывания** – упрятывание несущественной на конкретном этапе информации: каждая часть "знает" только необходимую ей информацию.
- ▣ **Принцип концептуальной общности** – следование единой философии на всех этапах ЖЦ информационных систем (структурный анализ – структурное проектирование – структурное программирование – структурное тестирование).
- ▣ **Принцип полноты** – контроль на присутствие лишних элементов.
- ▣ **Принцип непротиворечивости** – обоснованность и согласованность элементов.
- ▣ **Принцип логической независимости** – заключается в концентрации внимания на логическом проектировании для обеспечения независимости от физического проектирования.
- ▣ **Принцип независимости данных** – модели данных должны быть проанализированы и спроектированы независимо от процессов их логической обработки, а также от их физической структуры и распределения.
- ▣ **Принцип структурирования данных** – данные должны быть структурированы и иерархически организованы.
- ▣ **Принцип доступа конечного пользователя** – пользователь должен иметь средства доступа к базе данных, которые он может использовать непосредственно (без программирования).

СТРУКТУРНЫЙ ПОДХОД К ПРОЕКТИРОВАНИЮ ИС

SADT (Structured Analysis and Design Technique - Технология структурного анализа и проектирования) - одна из самых известных и широко используемых систем проектирования. Создатель методологии SADT – Дуглас Росс.


На ее основе разработана известная методология **IDEF0** (Interchange DEFinition), которая является основной частью программы “Интеграция компьютерных и промышленных технологий”, проводимой по инициативе ВВС США.



ОСНОВНЫЕ ПОЛОЖЕНИЯ МЕТОДОЛОГИИ SADT

- графическое изображение блоков и дуг SADT-диаграммы отображает функцию в виде блока, а входы и выходы представляются дугами, соответственно входящими в блок и выходящими из него. Взаимодействие блоков друг с другом описываются с помощью дуг, выражающих "ограничения", которые определяют, когда и каким образом функции выполняются и управляются;
- выполнение правил SADT требует строгости и точности, не накладывая в то же время сильных ограничений на действия аналитика.

Правила SADT:

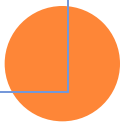
- ограничение количества блоков на каждом уровне декомпозиции (как правило 3-6 блоков);
 - связь диаграмм осуществляется при помощи нумерации блоков;
 - метки и наименования должны быть уникальными, т.е. не допускается повторение имен;
 - входы и управления должна разделяться.
- 

СТАНДАРТЫ IDEF (INTEGRATED COMPUTER AIDED MANUFACTURING DEFINITION)

- ▣ **IDEFO** - методология функционального моделирования. Система отображается в виде набора взаимосвязанных функциональных блоков.
- ▣ **IDEF1** – методология моделирования информационных потоков внутри системы, позволяющая отображать и анализировать их структуру и взаимосвязи;
- ▣ **IDEF1X** (IDEF1 eXtended) – методология построения реляционных структур. IDEF1X относится к типу методологий “Сущность-взаимосвязь” (ER – Entity-Relationship) и используется для моделирования реляционных баз данных в системе;
- ▣ **IDEF3** – методология документирования процессов. С помощью IDEF3 описываются сценарий и последовательность операций для каждого процесса.
- ▣ **IDEF4** – методология построения объектно-ориентированных систем.

СУЩНОСТЬ ФУНКЦИОНАЛЬНОГО МОДЕЛИРОВАНИЯ

В основе функционального моделирования лежит функциональное содержание системы, в качестве отношений между функциями рассматривается информация об объектах, связывающих эти функции.

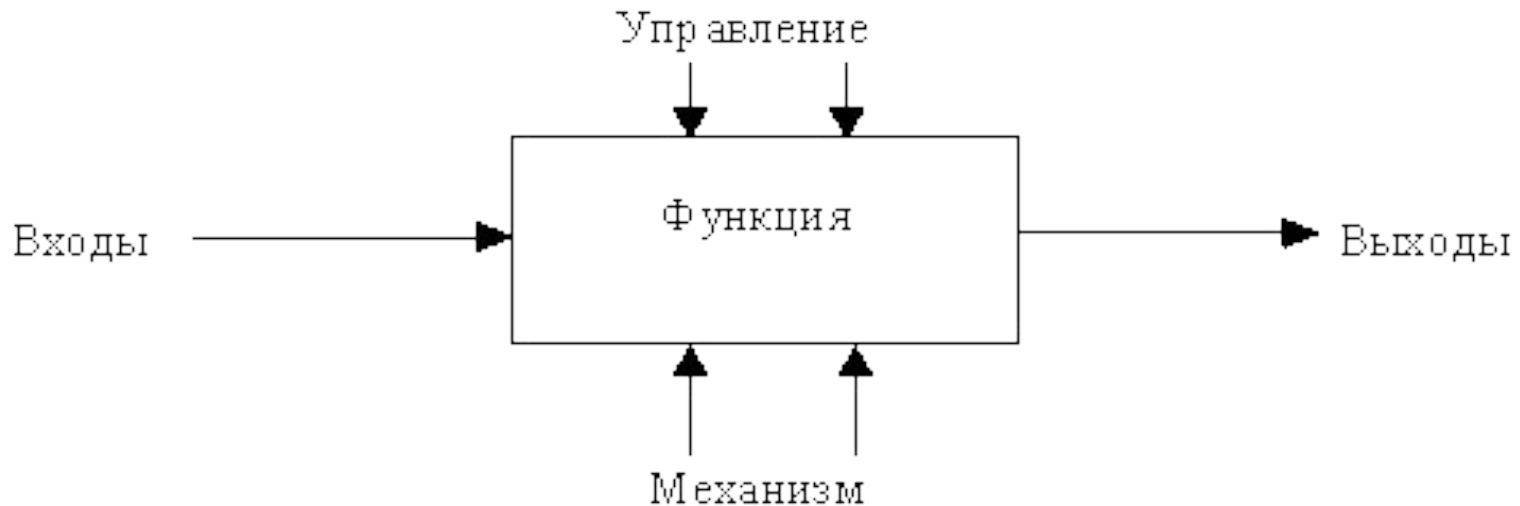


СОСТАВ ФУНКЦИОНАЛЬНОЙ МОДЕЛИ

SADT-модель - это описание системы, у которого есть единственный субъект, цель и одна точка зрения.

Цель - набор вопросов, на которые должна ответить модель.

Точка зрения - позиция, с которой описывается система.

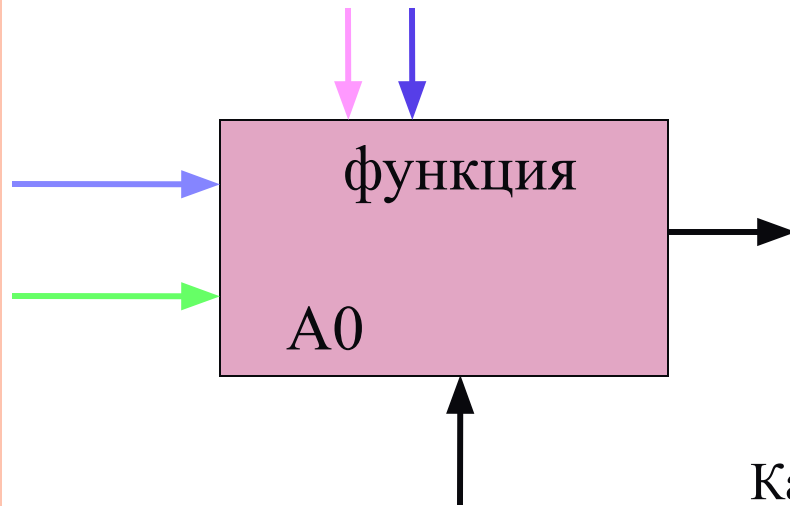


СИНТАКСИС SADT-ДИАГРАММ

- **Диаграммы содержат блоки и дуги;**
- **Блоки представляют функции;**
- **Блоки имеют доминирование (выражается в их ступенчатом расположении, причем доминирующий блок располагается в левом верхнем углу диаграммы);**
- **Дуги изображают наборы объектов, передаваемых между блоками;**
- **Дуги изображают различные типы взаимосвязей между блоками:**
 - **ВЫХОД – управление**
 - **ВЫХОД – ВХОД**
 - **обратная связь по управлению**
 - **обратная связь по входу**
 - **ВЫХОД – МЕХАНИЗМ.**



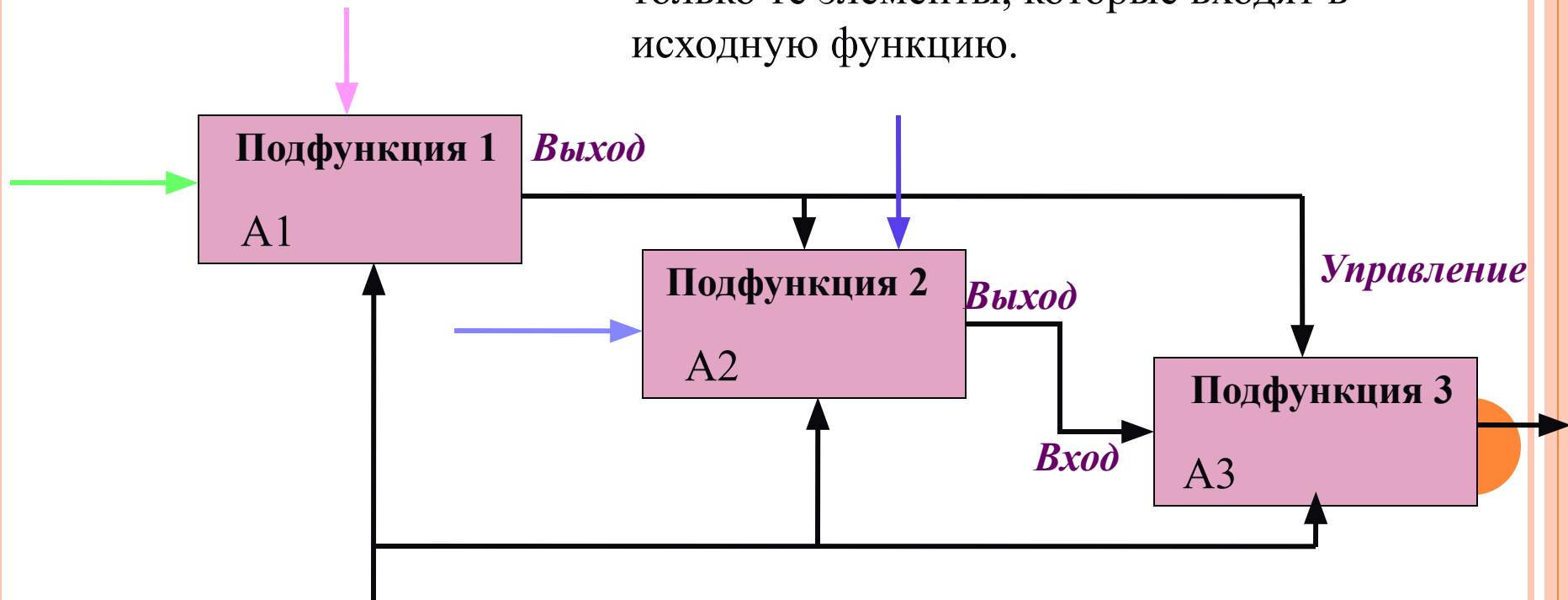
ДЕКОМПОЗИЦИЯ ФУНКЦИОНАЛЬНЫХ ДИАГРАММ



Контекстная диаграмма определяет все функции, входы и выходы, которые могут появиться на диаграммах нижних уровней

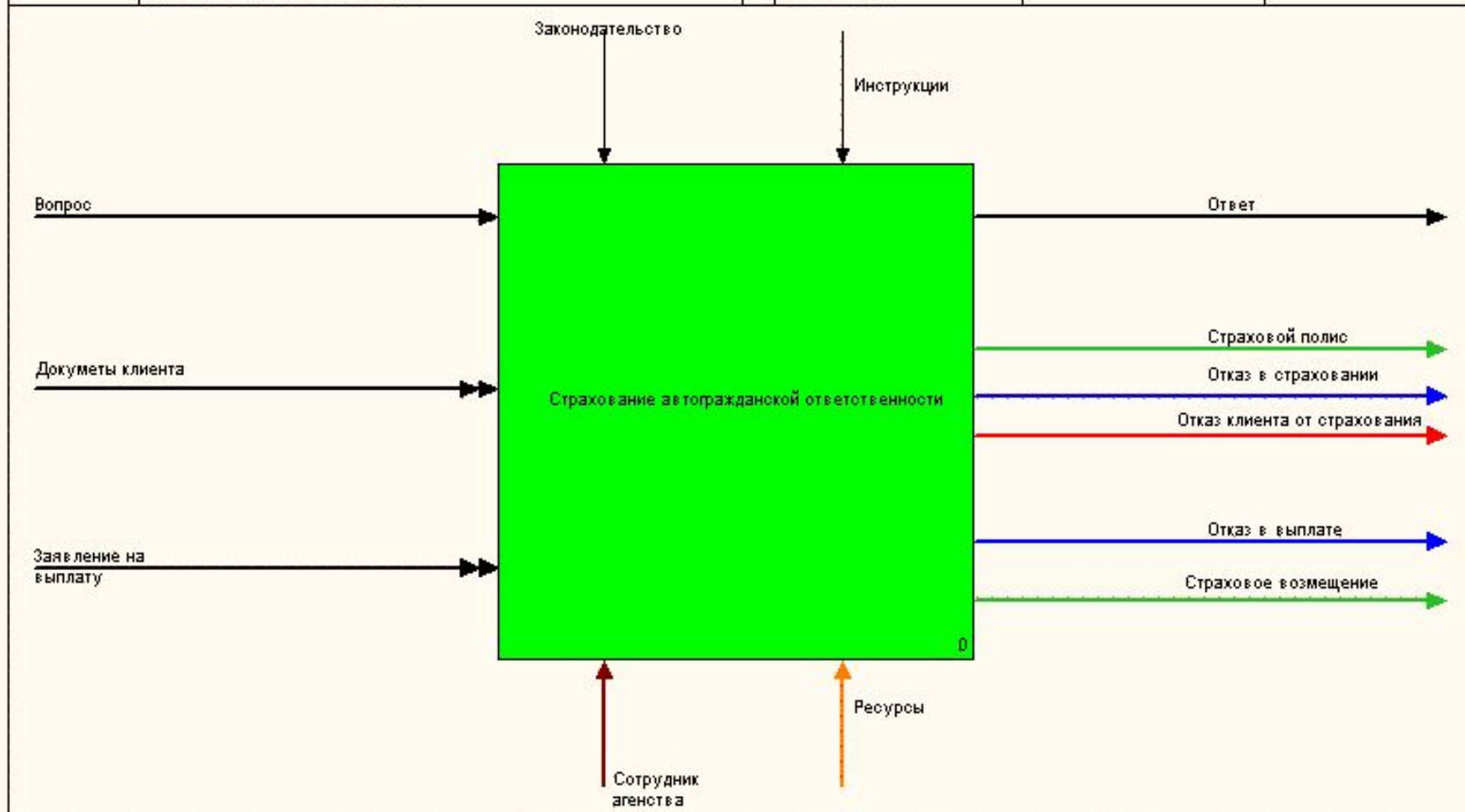
IDEFO

Каждая подфункция может содержать только те элементы, которые входят в исходную функцию.




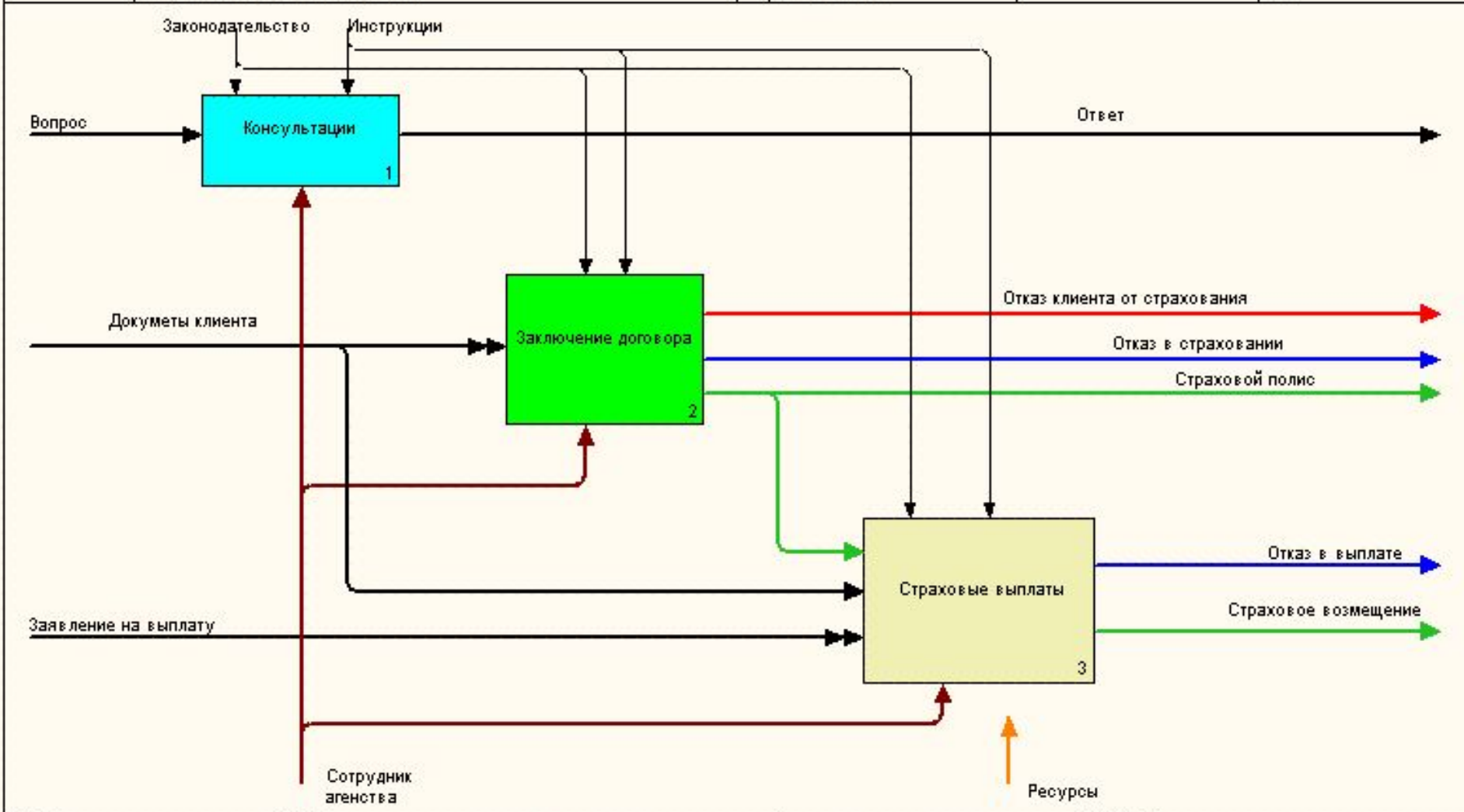
ПРИМЕР SADT-ДИАГРАММ(КОНТЕКСТНАЯ ДИАГРАММА)

USED AT:	AUTHOR: Свечников	DATE: 07.12.2003	WORKING	READER	DATE	CONTEXT: TOP
	PROJECT: Страхование АГО	REV: 11.02.2004	DRAFT			
			RECOMMENDED			
			PUBLICATION			
	NOTES: 1 2 3 4 5 6 7 8 9 10					



NODE: C-0	TITLE: Страхование автогражданской ответственности	NUMBER:
--------------	---	---------

USED AT:	AUTHOR: Свечников	DATE: 07.12.2003	WORKING	READER	DATE	CONTEXT:  C-0
	PROJECT: Страхование АГО	REV: 11.02.2004	DRAFT			
	NOTES: 1 2 3 4 5 6 7 8 9 10		RECOMMENDED			
			PUBLICATION			




NODE: CO	TITLE: Страхование автогражданской ответственности	NUMBER:
-------------	---	---------

рис.2

3.1.1 МЕТОДОЛОГИЯ IDEF0

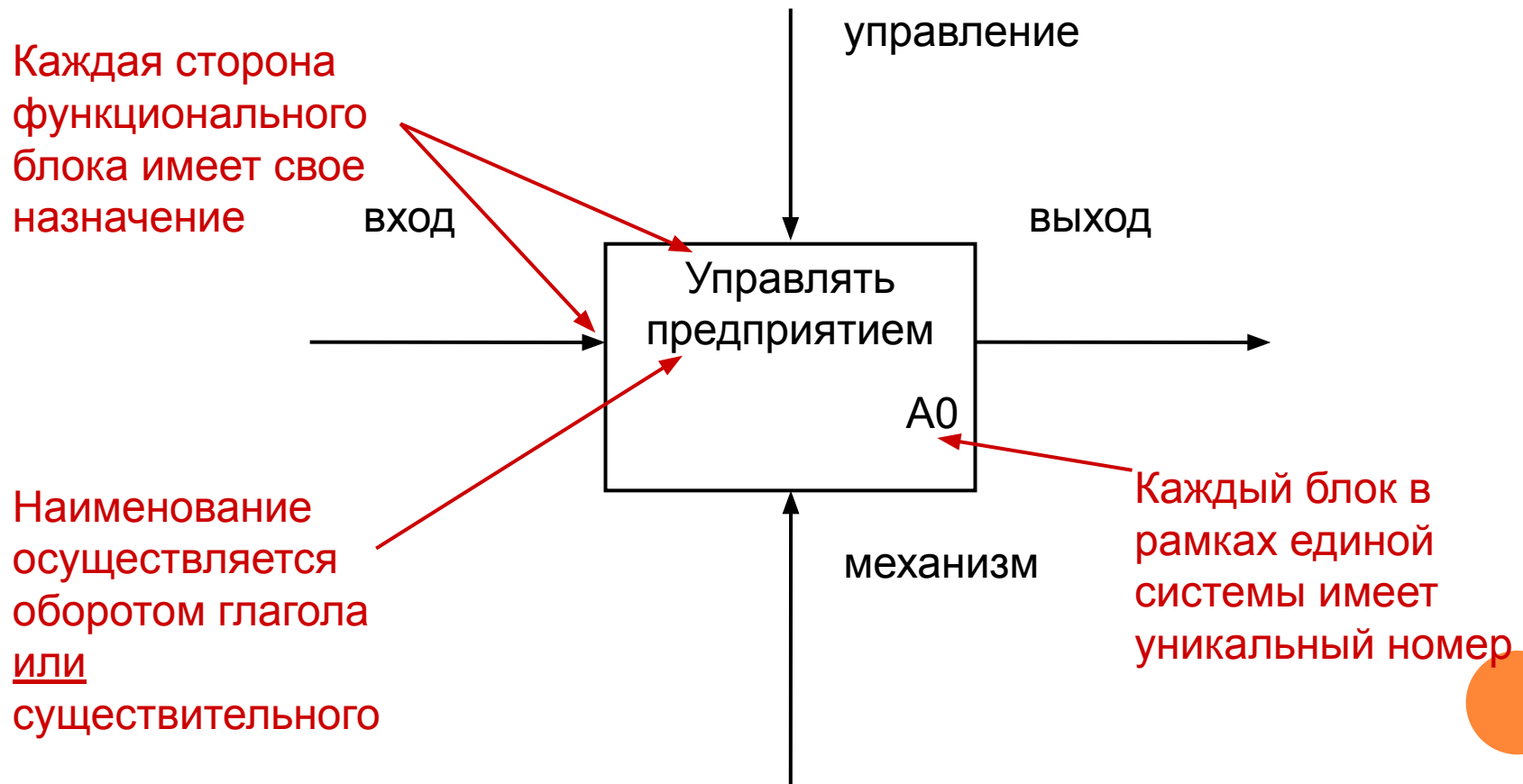
- Сущность методологии функционального моделирования IDEF0
- Основные понятия методологии IDEF0
- Правила построения моделей IDEF0
- Пример функциональной модели в нотации IDEF0

ОСНОВНЫЕ ПОНЯТИЯ МЕТОДОЛОГИИ IDEF0

- ▣ **Модель** – искусственный объект, представляющий собой отображение (образ) системы и ее компонентов.
 - ▣ **Система** представляет собой совокупность взаимосвязанных и взаимодействующих частей, выполняющих некоторую полезную работу.
 - ▣ **Лаконичность и точность.** Документация, описывающая систему, должна быть точной и лаконичной.
 - ▣ **Передача информации.** Средства IDEF0 облегчают передачу информации от одного участника разработки модели (отдельного разработчика или рабочей группы) к другому.
 - ▣ **Строгость и формализм.** Разработка моделей IDEF0 требует соблюдения ряда строгих формальных правил, обеспечивающих преимущества методологии в отношении однозначности, точности и целостности сложных многоуровневых моделей.
 - ▣ **Итеративное моделирование.** Разработка модели в IDEF0 представляет собой пошаговую, итеративную процедуру.
 - ▣ **Отделение «организации» от «функций».** При разработке моделей следует избегать изначальной «привязки» функций исследуемой системы к существующей организационной структуре моделируемого объекта (предприятия, фирмы).
- 

ФУНКЦИОНАЛЬНЫЙ БЛОК

- Олицетворяет некоторую конкретную функцию или работу в рамках рассматриваемой системы
- РД IDEF0 – 2000*: прямоугольник, содержащий имя и номер и используемый для описания функции

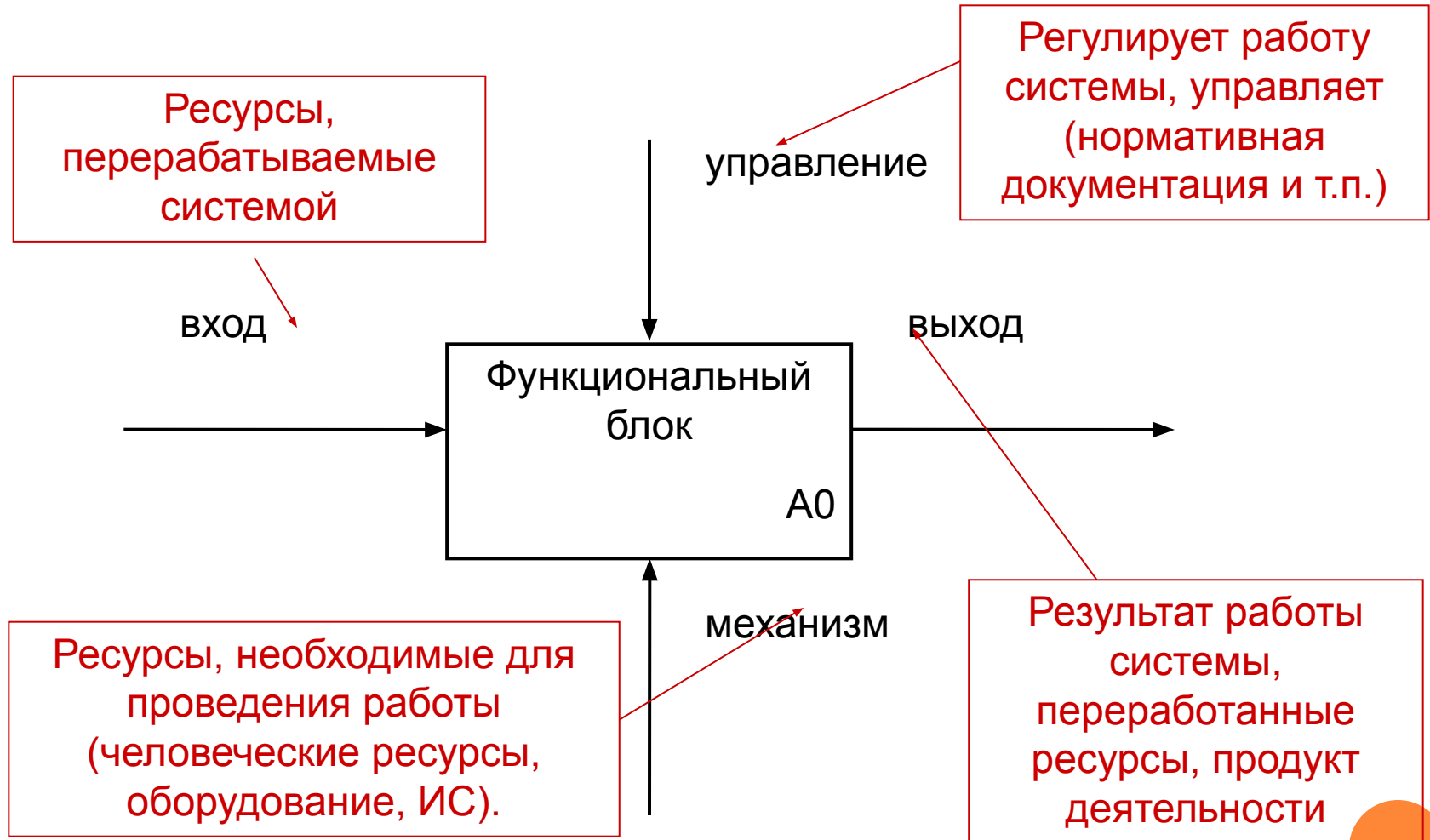


ИНТЕРФЕЙСНАЯ ДУГА

- Интерфейсная дуга отображает элемент системы, который обрабатывается функциональным блоком или оказывает иное влияние на функцию, отображаемую функциональным блоком.
- *Графически* изображается в виде однонаправленной стрелки.
- Каждая дуга должна иметь свое уникальное *название*, сформулированное оборотом существительного (должно отвечать на вопросы кто?, что?). Примеры: информация, разработчик, документ, обработанная заявка.
- В зависимости от того, к какой стороне блока она подходит, интерфейсная дуга будет являться *входящей, выходящей, управления, механизма*.



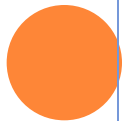
ИНТЕРФЕЙСНАЯ ДУГА



Стрелки входа может не быть. Остальные интерфейсные дуги обязательны.

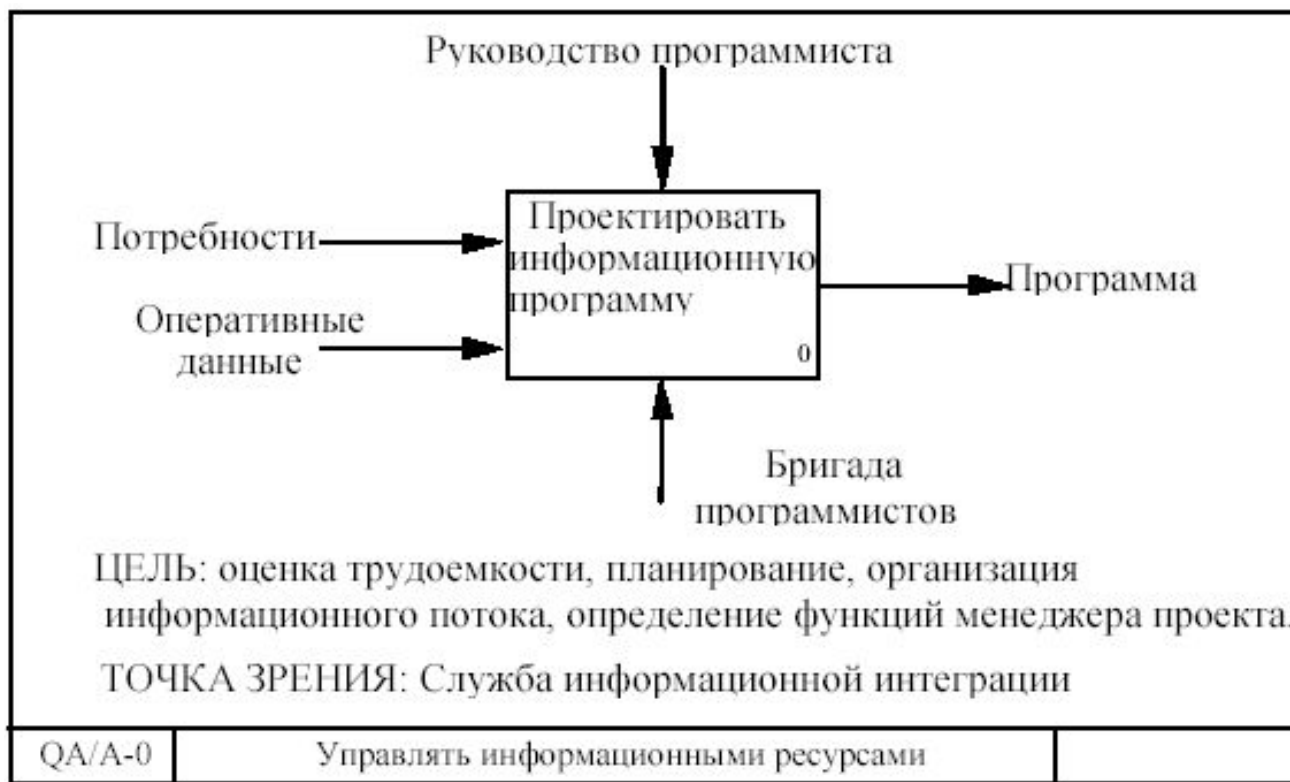
ДЕКОМПОЗИЦИЯ

- Принцип декомпозиции применяется при разбиении сложных процессов на составляющие его функции. При этом уровень детализации определяется непосредственно разработчиком модели.
- Модель IDEF0 всегда начинается с рассмотрения системы как единого целого, т.е. одного функционального блока с интерфейсными дугами, простирающимися за пределы рассматриваемой области. Такая диаграмма называется *контекстной*, она обозначается идентификатором А-0.
- Для определения границ системы на контекстной диаграмме обязательно должны быть цель и точка зрения.



КОНТЕКСТНАЯ ДИАГРАММА ВЕРХНЕГО УРОВНЯ

Эта диаграмма называется А-0 (А ноль). Стрелки на этой диаграмме отображают связи объекта моделирования с окружающей средой.

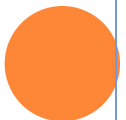


ЦЕЛЬ МОДЕЛИРОВАНИЯ

Цель моделирования должна отвечать на следующие вопросы:

- Почему процесс должен быть смоделирован?
- Что должна показывать модель?
- Что может получить читатель?

Примеры целей: «Идентифицировать слабые стороны процесса сбора данных», «Определить ответственность сотрудников для написания должностных инструкций» и т.п.

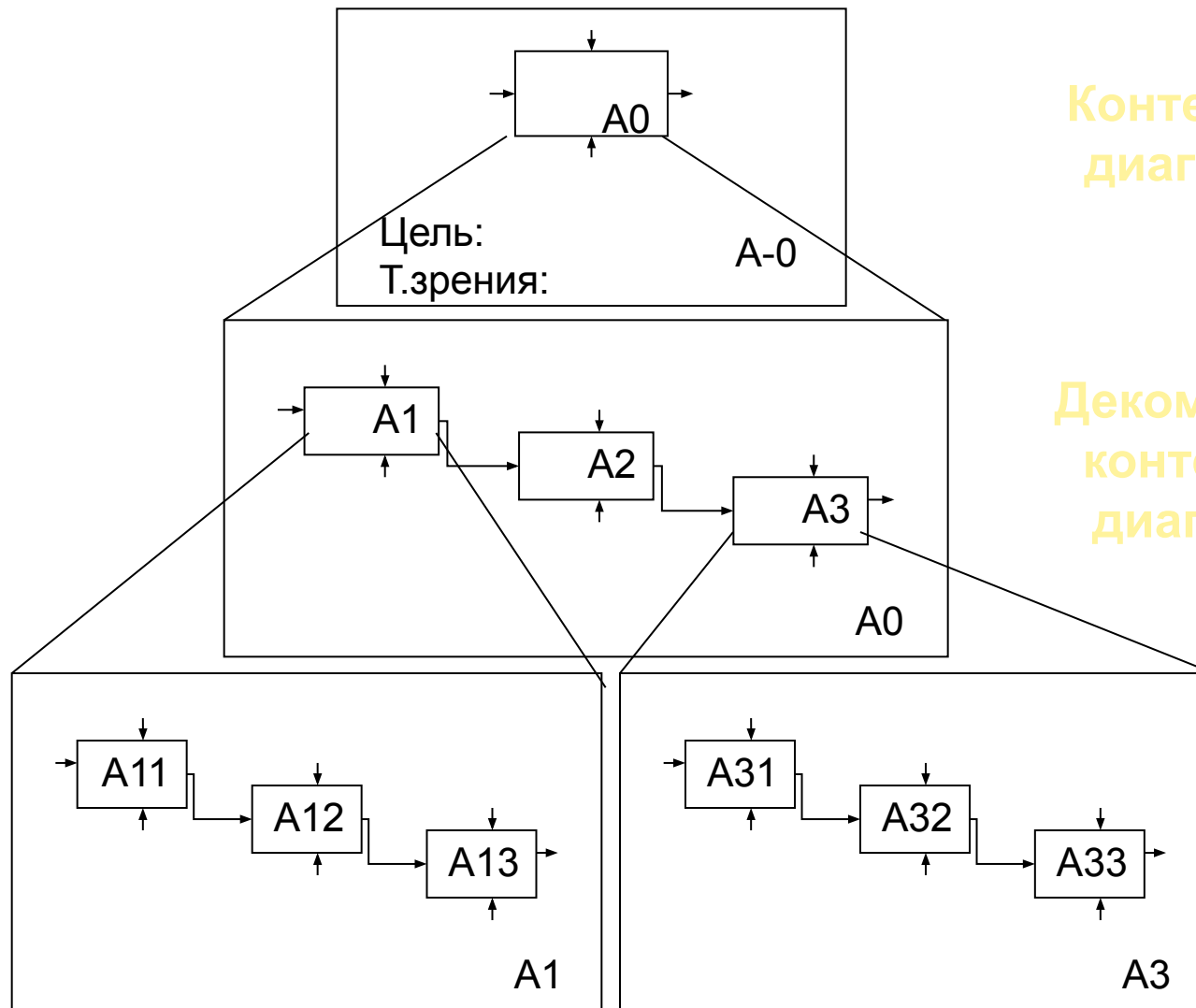


ТОЧКА ЗРЕНИЯ

- Точка зрения – позиция, с которой будет строиться модель. В качестве точки зрения берется взгляд человека, который видит систему в нужном для моделирования аспекте.
- Как правило, выбирается точка зрения человека, **ответственного** за выполнение моделируемой работы.
- Между целью и точкой зрения должно быть жесткое соответствие.



ДЕКОМПОЗИЦИЯ



Контекстная
диаграмма

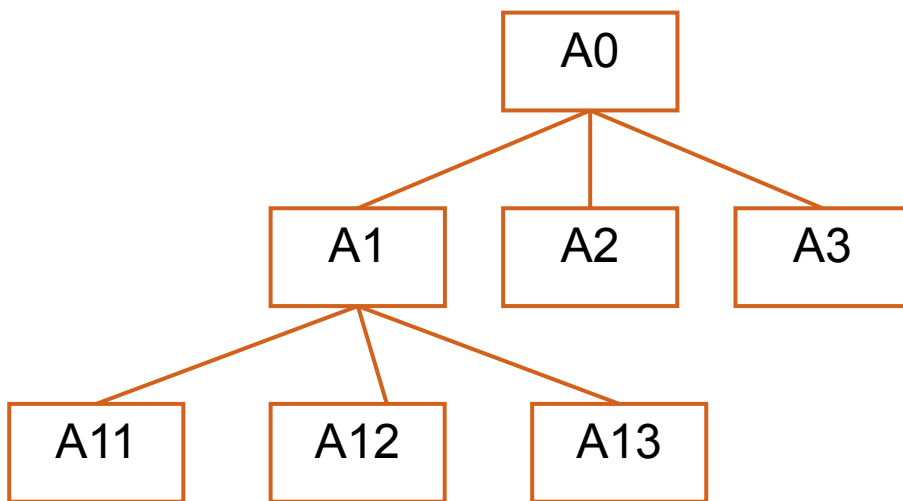
Декомпозиция
контекстной
диаграммы

Декомпозиция блока A1

Декомпозиция блока A3



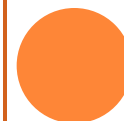
ДЕКОМПОЗИЦИЯ



Дерево узлов

A0 _____
A1 _____
 A11 _____
 A12 _____
 A13 _____
A2 _____
A3 _____

Индекс узлов



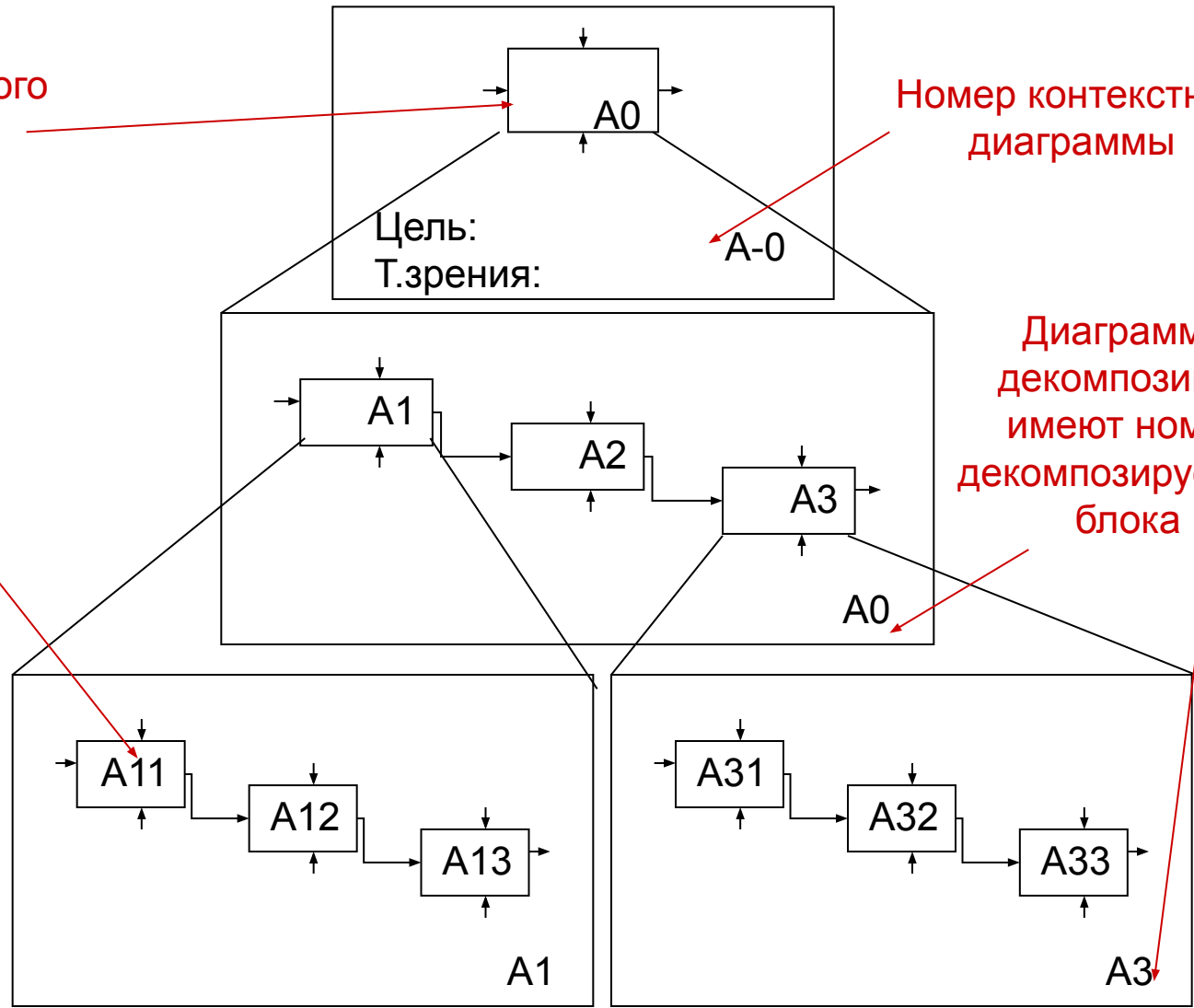
НУМЕРАЦИЯ РАБОТ И ДИАГРАММ

Номер функционального блока на контекстной диаграмме

Номер контекстной диаграммы

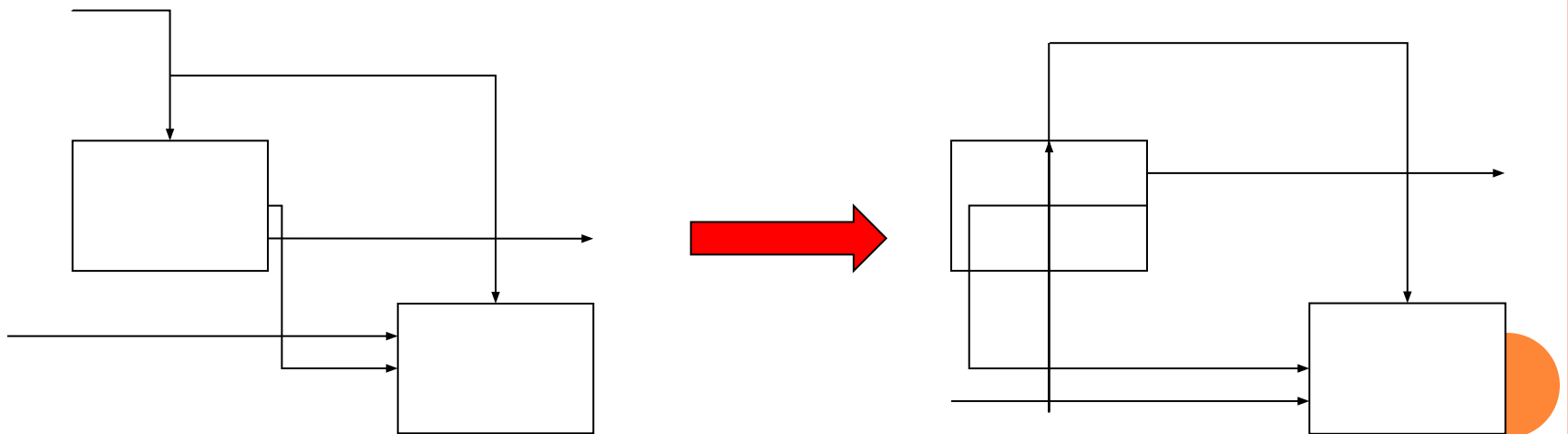
- Формат номера блока:
1. Префикс
 2. Номер родительской работы
 3. Собственный порядковый номер

Диаграммы декомпозиции имеют номер декомпозируемого блока



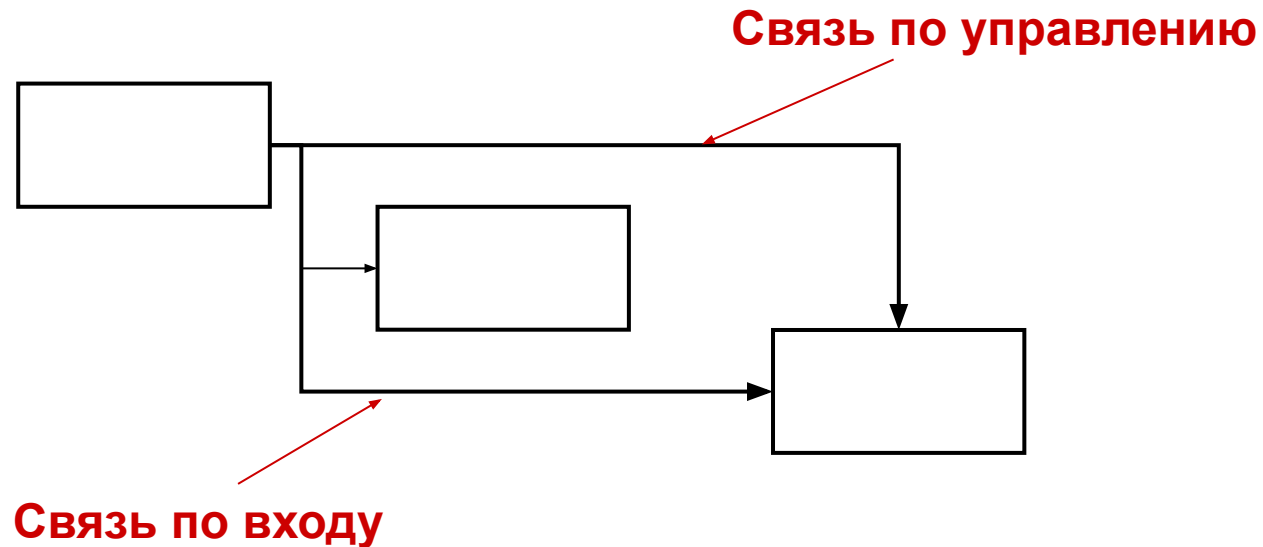
ОСНОВНЫЕ ПРАВИЛА ПОСТРОЕНИЯ ДИАГРАММ

1. На одной диаграмме рекомендуется рисовать от 3 до 6 блоков. Иначе диаграмма будет плохо читаемой.
2. Функциональные блоки должны располагаться слева направо сверху вниз в порядке доминирования.
3. Следует избегать излишнего пересечения стрелок.

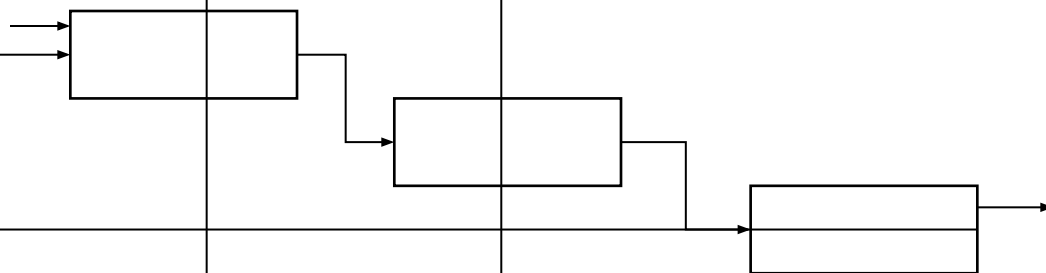


ОСНОВНЫЕ ПРАВИЛА ПОСТРОЕНИЯ ДИАГРАММ

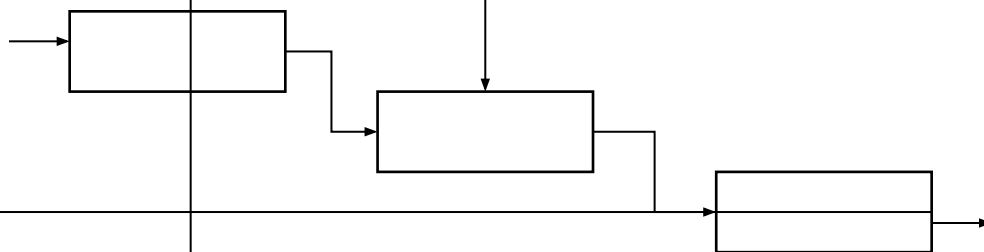
4. Выход одного блока может являться входом (управлением) для другого. Могут быть и обратные связи по входу и управлению.



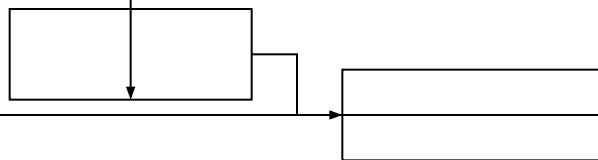
ОСНОВНЫЕ ПРАВИЛА ПОСТРОЕНИЯ ДИАГРАММ



а) обратная связь по входу



б) обратная связь по управлению



в) обратная связь по механизму

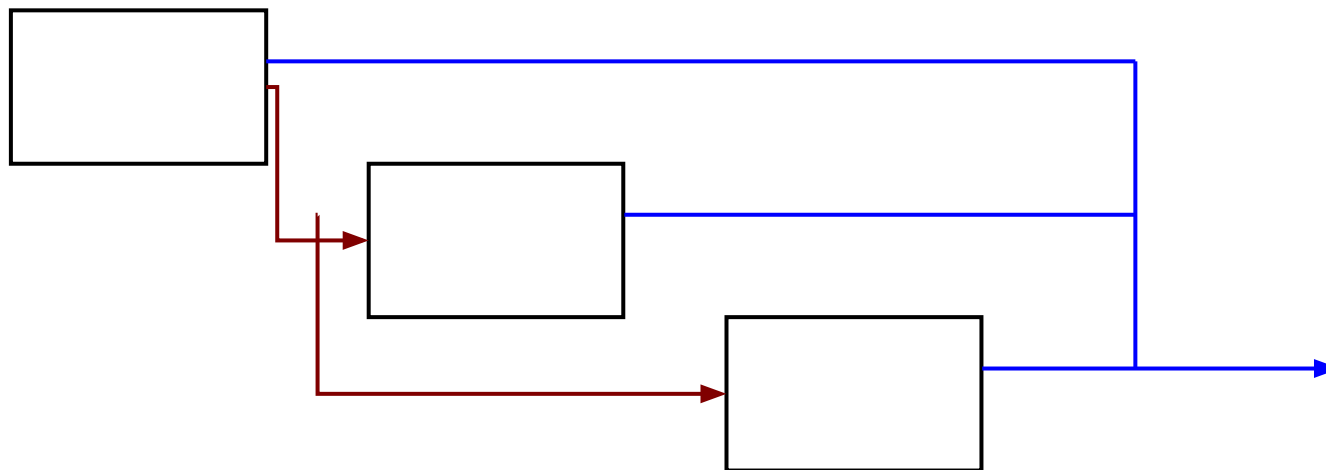
Обратная связь по входу, как правило, используется для описания циклов.



Обратная связь по управлению – выход нижестоящей работы передается на управление вышестоящей

Обратная связь по механизму – выход нижестоящей работы создает ресурсы, выполняющие вышестоящую работу

ОСНОВНЫЕ ПРАВИЛА ПОСТРОЕНИЯ ДИАГРАММ

5. Стрелки могут быть сливающимися и разветвляющимися

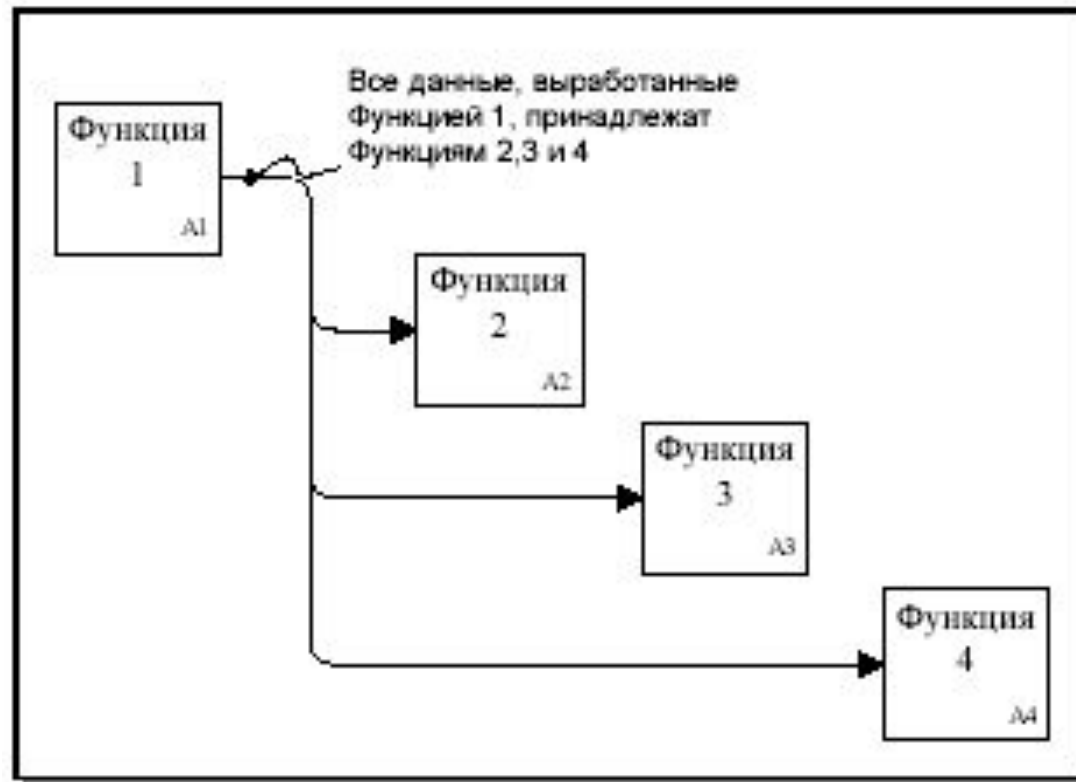


-  Слияние стрелок
-  Разветвление стрелок



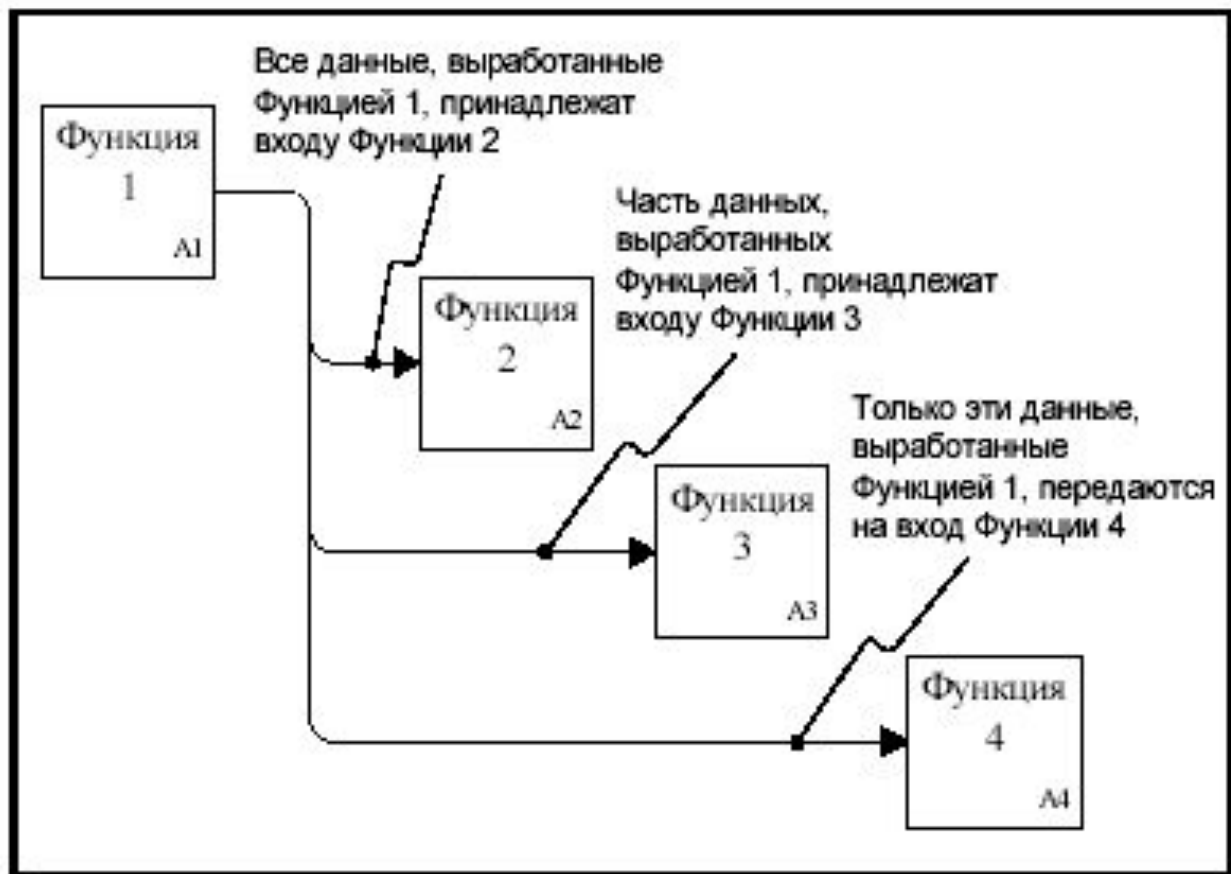
ВЕТВЛЕНИЕ И СЛИЯНИЕ СЕГМЕНТОВ СТРЕЛОК

непомеченные сегменты содержат все объекты, указанные в метке стрелки перед ветвлением (т.е. все объекты принадлежат каждому из сегментов)



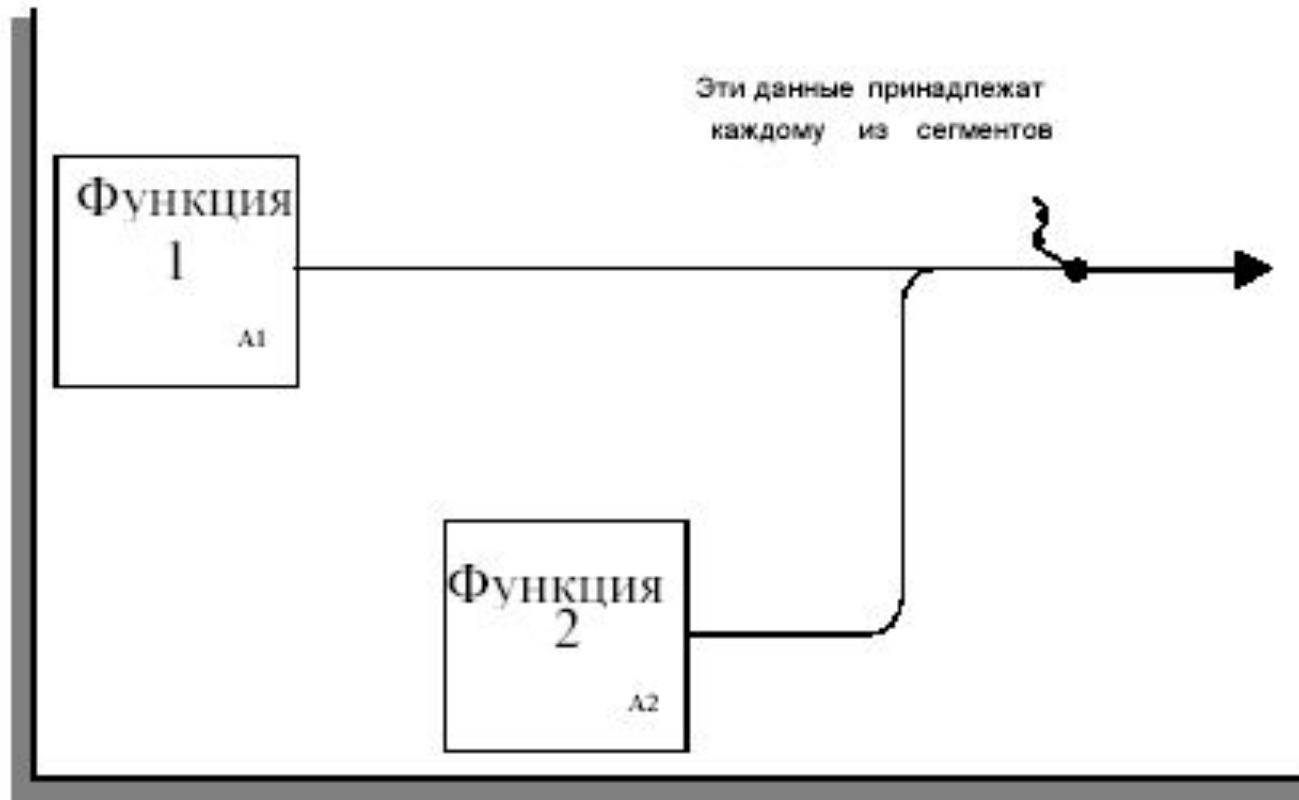
ВЕТВЛЕНИЕ И СЛИЯНИЕ СЕГМЕНТОВ СТРЕЛОК

сегменты, помеченные после точки ветвления, содержат все объекты, указанные в метке стрелки перед ветвлением, или их часть, описываемую меткой каждого конкретного сегмента;



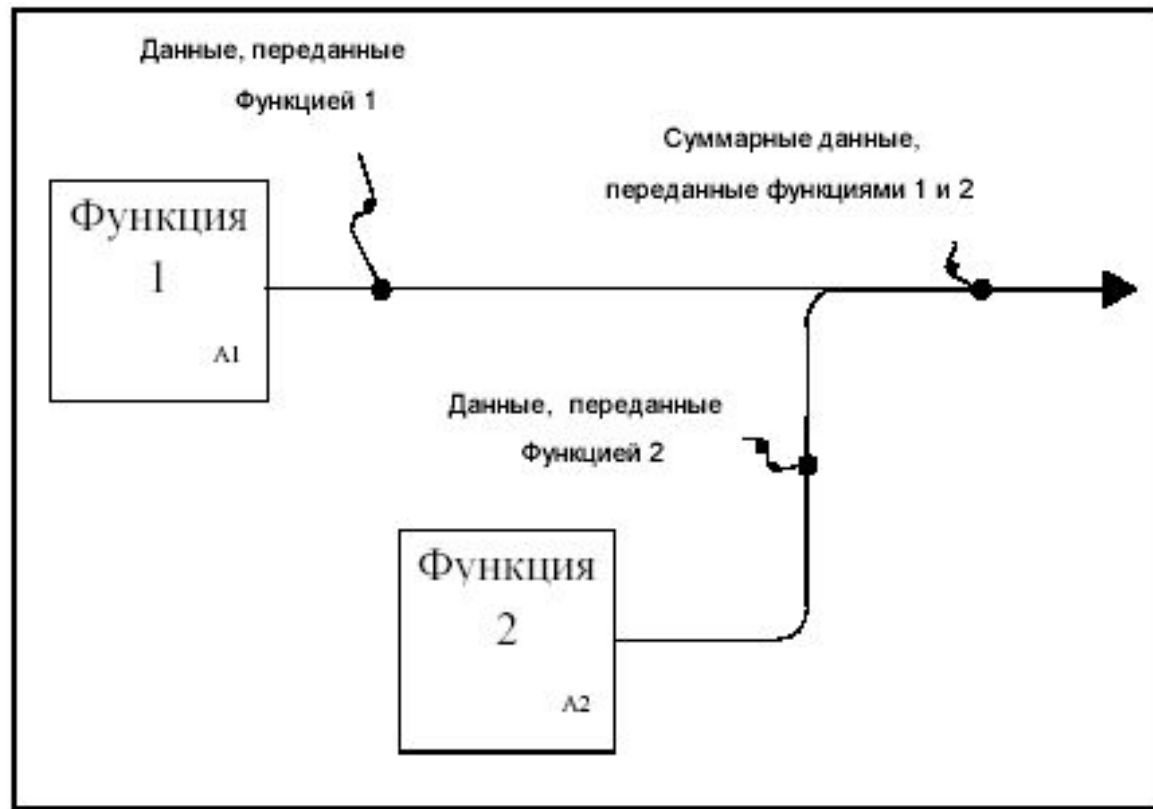
ВЕТВЛЕНИЕ И СЛИЯНИЕ СЕГМЕНТОВ СТРЕЛОК

при слиянии непомеченных сегментов объединенный сегмент стрелки содержит все объекты, принадлежащие сливаемым сегментам и указанные в общей метке стрелки после слияния



ВЕТВЛЕНИЕ И СЛИЯНИЕ СЕГМЕНТОВ СТРЕЛОК

при слиянии помеченных сегментов объединенный сегмент содержит все или некоторые объекты, принадлежащие сливаемым сегментам и перечисленные в общей метке после слияния; если общая метка после слияния отсутствует, это означает, что общий сегмент передает все объекты, принадлежащие сливаемым сегментам;



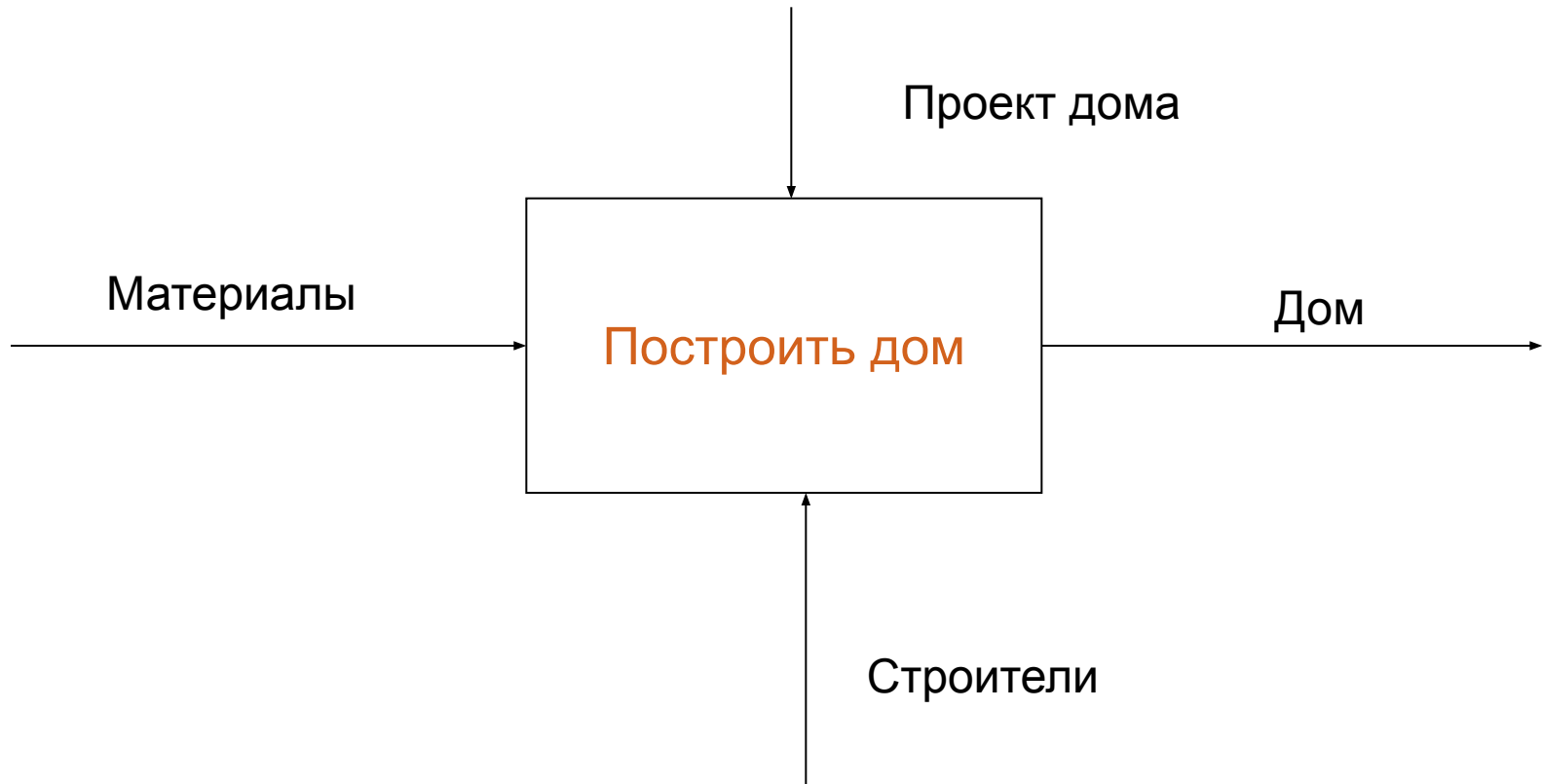
ТУННЕЛЬНЫЕ СТРЕЛКИ

Иногда необходимо отобразить граничные стрелки, которые значимы на данном уровне и не значимы на родительской диаграмме. Например, некоторые данные используются только на данном уровне и не используются на других. Без использования механизма туннелирования малозначимая стрелка появится на всех уровнях модели, что затруднит чтение диаграмм.



ПРИМЕР МОДЕЛИ ПРОЦЕССА ПОСТРОЙКИ САДОВОГО ДОМИКА

1. Строим контекстную диаграмму.

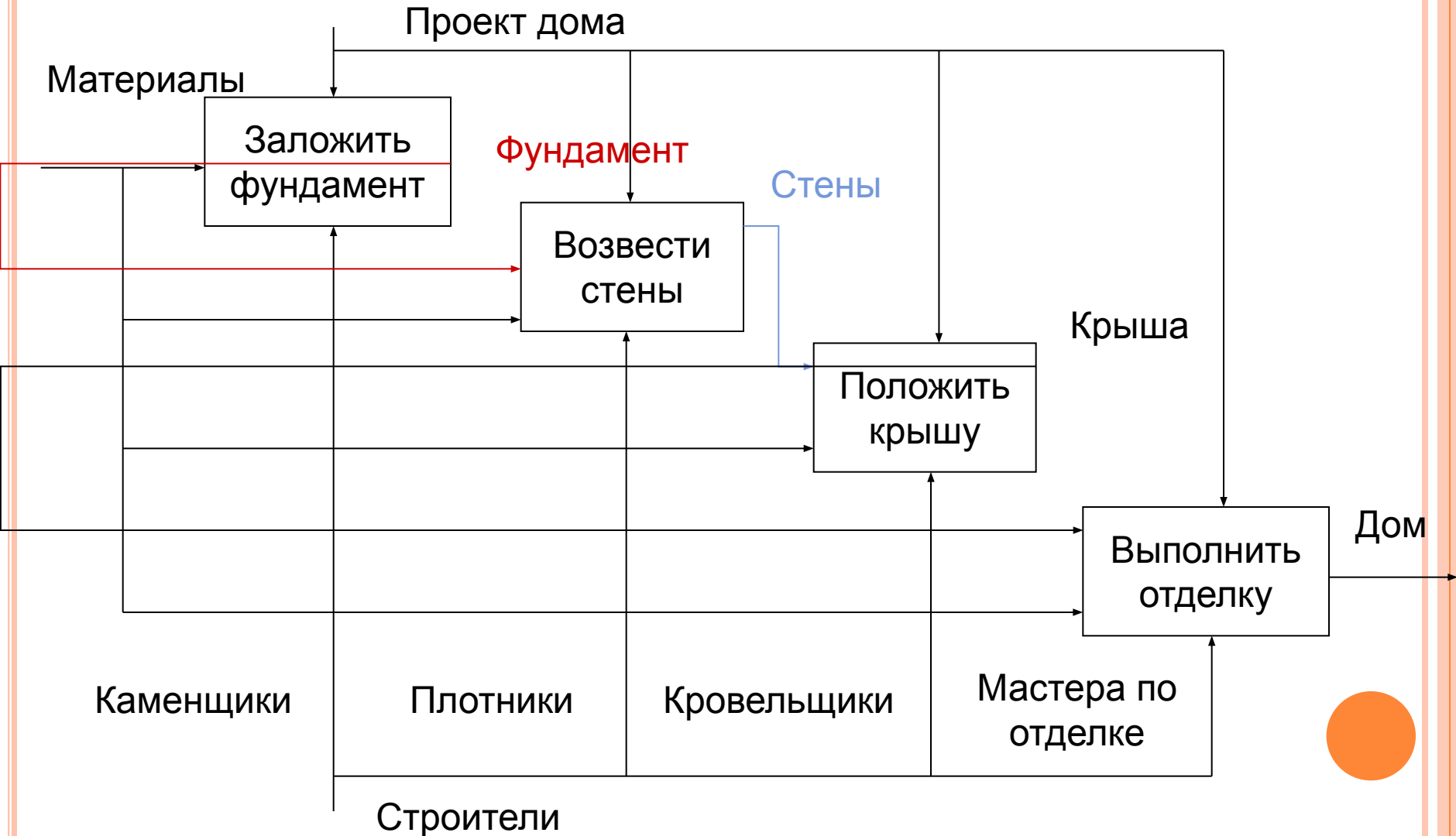


Цель: Определить действия, необходимые для постройки дачного домика

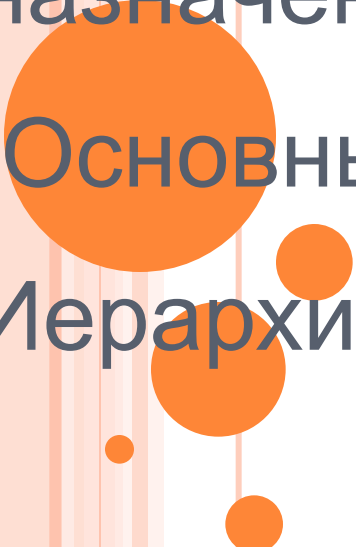
Точка зрения: владельца дачного участка

ПРИМЕР МОДЕЛИ ПРОЦЕССА ПОСТРОЙКИ САДОВОГО ДОМИКА

2. Декомпозируем контекстную диаграмму



3.1.2 ДИАГРАММЫ ПОТОКОВ ДАННЫХ (DFD)

- Определение и функциональное назначение DFD-моделей
 - Основные компоненты DFD-моделей
 - Иерархия DFD
- 

ЧТО ТАКОЕ DFD-МОДЕЛЬ

- **DFD – Data Flow Diagrams** – диаграммы потоков данных
- **Модель системы** определяется как иерархия диаграмм потоков данных, описывающих асинхронный процесс преобразования информации от ее входа в систему до выдачи пользователю.



ЧТО ТАКОЕ DFD-МОДЕЛЬ?

- Главная цель такого представления — продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.
- **Примечание.** *DFD-модели могут быть использованы в дополнение к модели IDEF0 для более наглядного отображения текущих операций документооборота в корпоративных системах обработки информации.*



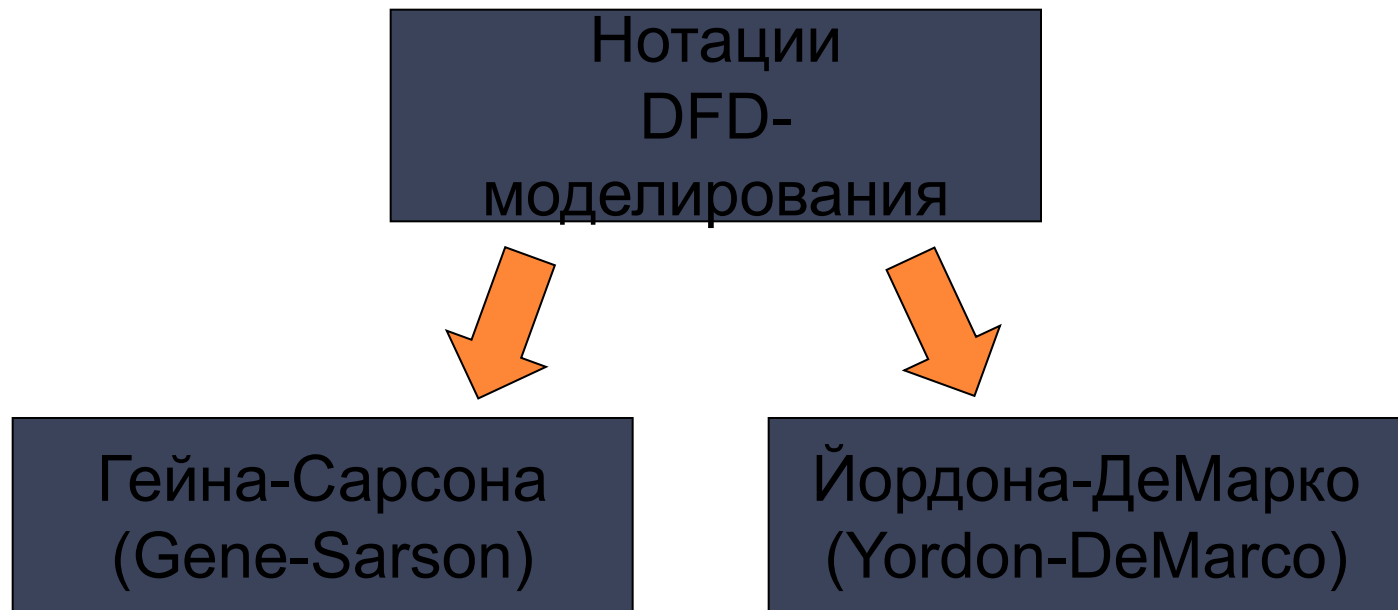
ОСНОВНЫЕ КОМПОНЕНТЫ ДИАГРАММ ПОТОКОВ ДАННЫХ

Основными компонентами диаграмм потоков данных являются:

- внешние сущности
- системы и подсистемы
- процессы
- накопители данных
- потоки данных.



НОТАЦИИ, ИСПОЛЬЗУЕМЫЕ В DFD-МОДЕЛИРОВАНИИ



Примечание. В зависимости от используемой нотации графическое представление элементов диаграмм будет различным

ВНЕШНЯЯ СУЩНОСТЬ

- Представляет собой *материальный объект* или *физическое лицо*, являющееся *источником* или *приемником информации* (например, заказчики, клиенты, поставщики, склад, персонал, банк).
- Внешняя сущность находится за пределами границ анализируемой системы.
- Одна и та же внешняя сущность может быть использована многократно на одной или нескольких диаграммах.



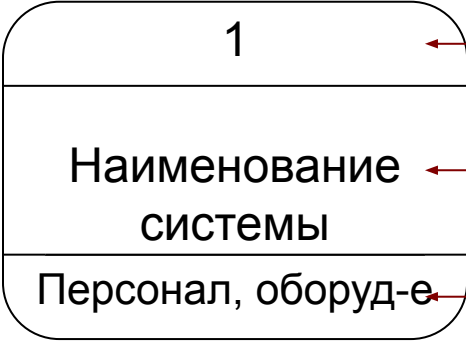
Внешняя сущность в
нотации Йордона-ДеМарко

Внешняя сущность в
нотации Гейна-Сарсона

СИСТЕМА И ПОДСИСТЕМА

- При построении модели сложной системы она может быть представлена в самом общем виде на так называемой **контекстной диаграмме** в виде одной **системы**, либо в виде ряда **подсистем**.
- **Наименование** системы/подсистемы представляется в виде словосочетания с отглагольным существительным (рассмотрение повестки дня, решение задачи, получение денег и т.п.).

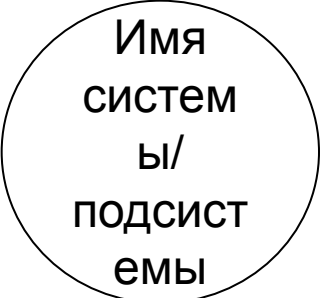
**Система/подсистема
в нотации Гейна-
Сарсона**



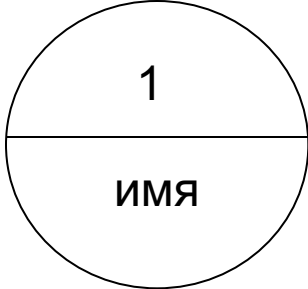
← Поле идентификации

← Поле имени

← Поле физической реализации



или



**Система/подсистема в
нотации Йордона-
ДеМарко**

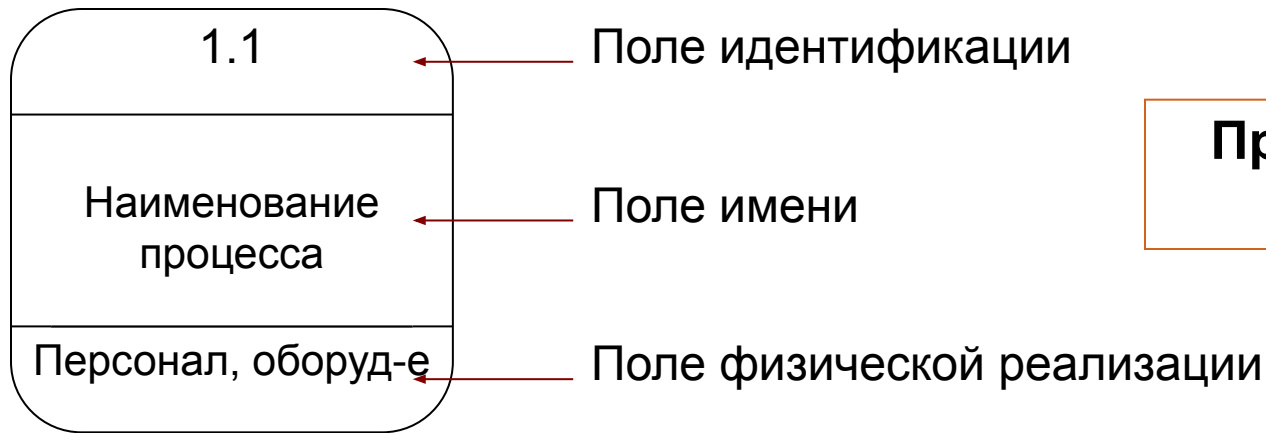


ПРОЦЕСС

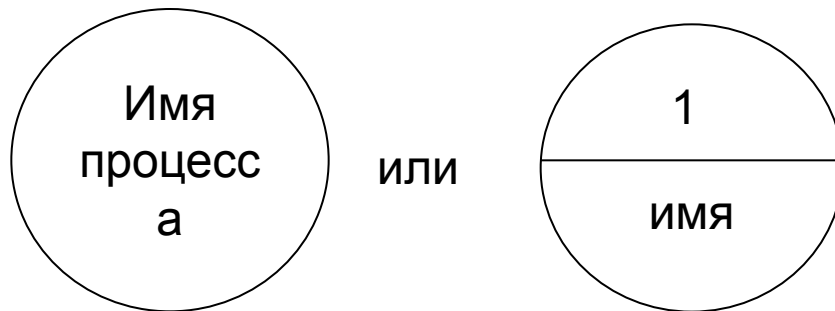
- Представляет собой *преобразование* ВХОДНЫХ ПОТОКОВ в ВЫХОДНЫЕ в соответствии с определенным алгоритмом.
- Примеры: обработка входных документов и выпуск отчетности определенным подразделением, процессы физически реализованного устройства.
- Процесс *именуется* в виде словосочетания с активным глаголом в неопределенной форме, за которым следует существительное в винительном падеже.



ПРОЦЕСС



**Процесс в нотации
Гейна-Сарсона**



**Процесс в нотации
Йордона-ДеМарко**

!!!! Процесс отличается от системы/подсистемы по полю наименования!!!!

НАКОПИТЕЛЬ ДАННЫХ

Это абстрактное устройство для хранения информации, которую можно в любой момент поместить в накопитель и через некоторое время извлечь.



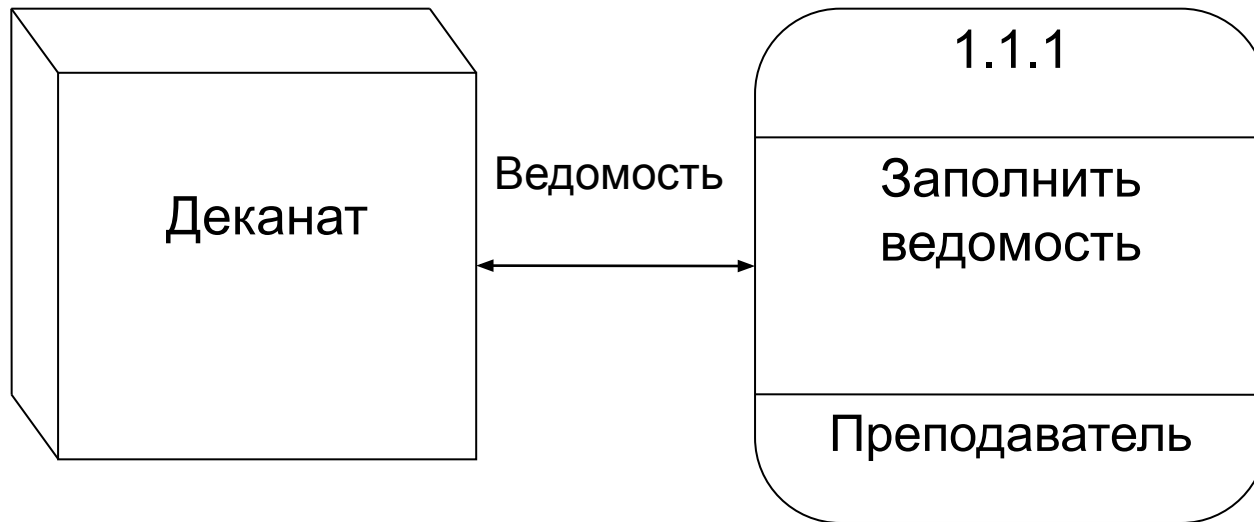
Примеры: ящик в картотеке, таблицы в ОЗУ, файл на электронном носителе

Примечание: В нотациях Гейна-Сарсона и Йордона-ДеМарко графическое представление данного элемента аналогичное.



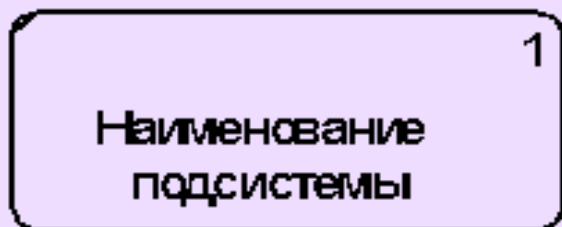
ПОТОК ДАННЫХ

- Определяет информацию, передаваемую через некоторые соединения от источника к приемнику. Реальный поток данных может быть информацией, передаваемой по кабелю между двумя устройствами, пересылаемыми по почте письмами и т.п.



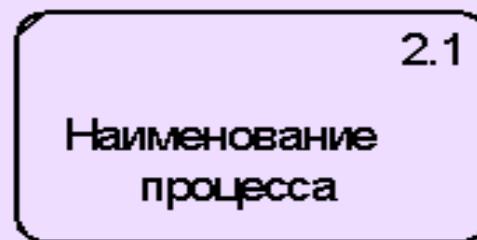
НУМЕРАЦИЯ ОБЪЕКТОВ

Системы, подсистемы



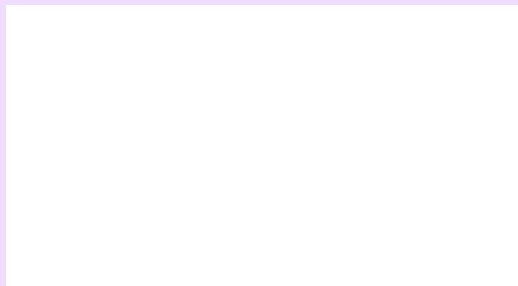
[Префикс] + собственный номер

Процессы



[Префикс]+номер родительской подсистемы+собственный номер

Внешние сущности



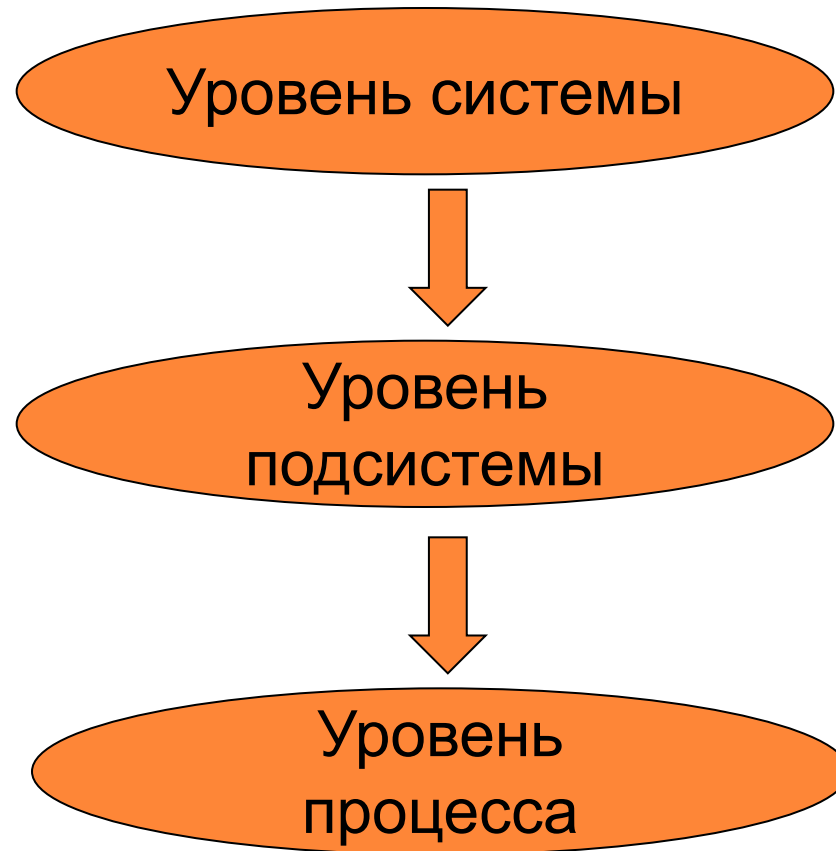
[Префикс]+номер

Хранилища данных



[Префикс]+номер

УРОВНИ DFD-МОДЕЛИ



ПОСТРОЕНИЕ ИЕРАРХИИ DFD

1. Построение диаграмм уровня системы и подсистемы



ПОСТРОЕНИЕ ИЕРАРХИИ DFD

2. Построение диаграмм уровня процесса

ПРИМЕР DFD-МОДЕЛИ ПОСТРОЙКИ ДАЧНОГО ДОМИКА

1. Контекстная диаграмма уровня системы

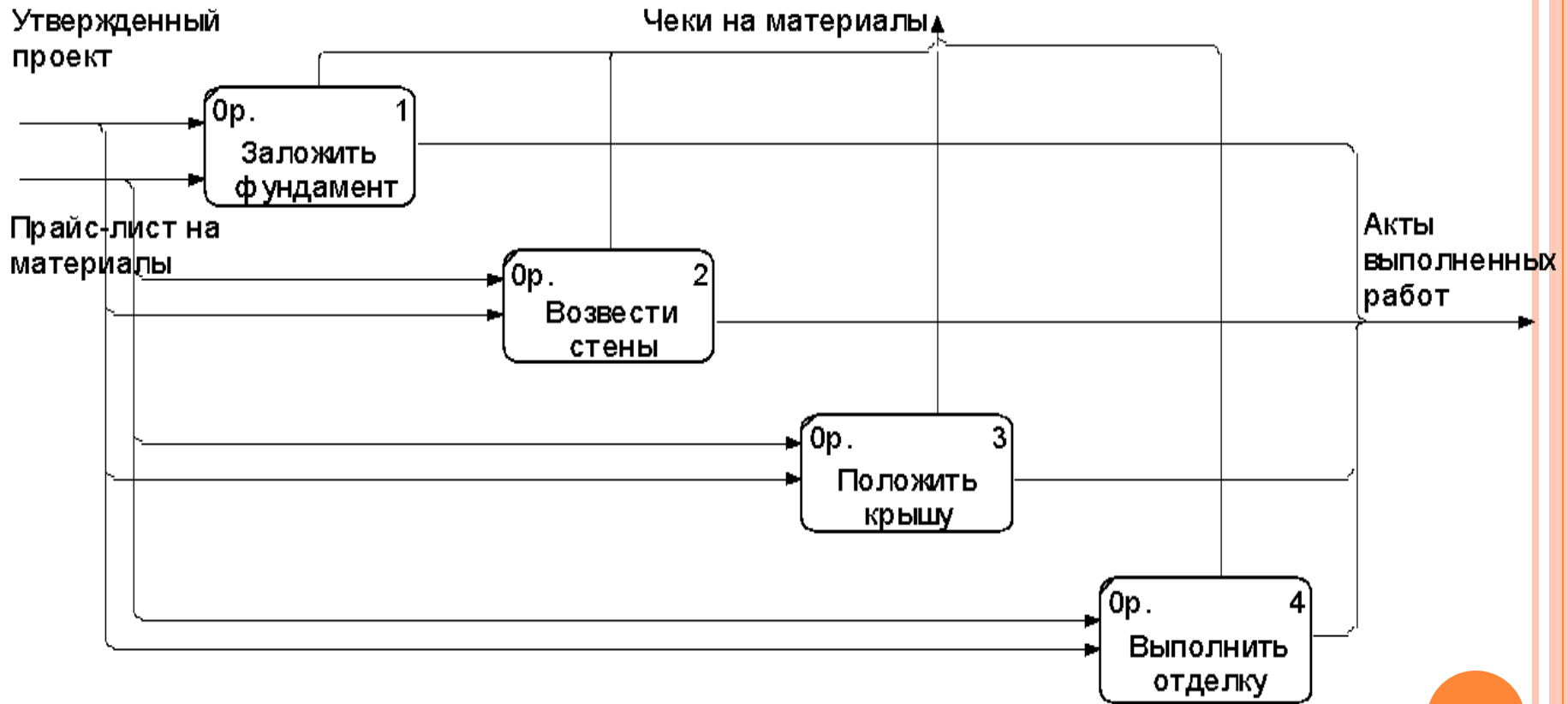


ПРИМЕР DFD-МОДЕЛИ ПОСТРОЙКИ ДАЧНОГО ДОМИКА

2. Диаграмма уровня подсистемы

ПРИМЕР DFD-МОДЕЛИ ПОСТРОЙКИ ДАЧНОГО ДОМИКА

3. Диаграмма уровня процесса



3.1.3 МЕТОДОЛОГИЯ IDEF3

- Понятие динамического моделирования
- Методология IDEF3
- Основные элементы динамической модели
- Правила и особенности построения IDEF3-модели
- Декомпозиция в IDEF3

ЧТО ОТРАЖАЕТ МОДЕЛЬ IDEF3?

В общем случае, процесс – это упорядоченная последовательность действий.

Следовательно, процессная модель IDEF3 *позволяет:*

- Отобразить последовательность процессов
- Показать логику взаимодействия элементов системы.

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



ОСНОВНЫЕ ЭЛЕМЕНТЫ ДИАГРАММ IDEF3

Точка зрения на модель - это точка зрения человека, ответственного за работу в целом.

Цель модели — те вопросы, на которые призвана ответить модель.

Единицы работы - Unit of Work (UOW), также называемые работами (activity), являются центральными компонентами модели.

Связи. Связи показывают взаимоотношения 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.


Объект ссылки - в IDEF3 выражает некую идею, концепцию или данные, которые нельзя связать со стрелкой, перекрестком или работой.





ЕДИНИЦЫ РАБОТ

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

Номер работы является уникальным, присваивается при ее создании и не меняется никогда



Словосочетание с отглагольным существительным, изображающим действие (выполнение, изготовление,...)
Или
Инфинитив глагола (изготовить продукцию)



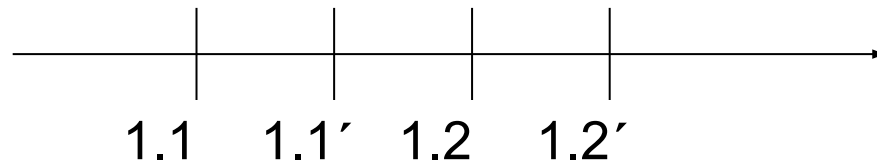
СВЯЗИ

- СВЯЗИ показывают *взаимоотношения* работ.
- Связи *однонаправлены* и могут быть направлены куда угодно
- Обычно диаграммы рисуют таким образом, чтобы связи были направлены слева направо
- Различают *3 типа* связей:
 - Старшая стрелка
 - Стрелка отношений
 - Поток объектов.



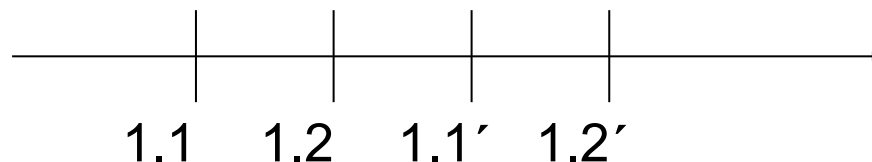
СВЯЗЬ «СТАРШАЯ СТРЕЛКА»

- Связь типа «*временное предшествование*» - *Precedence*
- Соединяет единицы работ
- Показывает, что работа-источник должна быть закончена **прежде**, чем начнется работа-цель



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




- Стрелка, изображающая **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 (СОЕДИНЕНИЯ)

- 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.
- 

ТИПЫ ПЕРЕКРЕСТКОВ

Обозначение	Наименование	Смысл в случае слияния стрелок	Смысл в случае разветвления стрелок
	<i>Асинхронное «И»</i>	Все предшествующие процессы должны быть завершены	Все последующие процессы должны быть запущены
	<i>Синхронное «И»</i>	Все предшествующие процессы должны быть завершены одновременно	Все последующие процессы запускаются одновременно
	<i>Асинхронное «ИЛИ»</i>	Один или несколько предшествующих процессов должны быть завершены	Один или несколько следующих процессов должны быть запущены

ТИПЫ ПЕРЕКРЕСТКОВ

Обозначение	Наименование	Смысл в случае слияния стрелок	Смысл в случае разветвления стрелок
	<i>Синхронное «ИЛИ»</i>	Один или несколько предшествующих процессов должны быть завершены одновременно	Один или несколько следующих процессов должны быть запущены одновременно
	<i>Эксклюзивное (исключающее) «ИЛИ»</i>	Только один предшествующий процесс должен быть завершен	Только один следующий процесс запускается

ПРАВИЛА СОЗДАНИЯ 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»**

ПРАВИЛА СОЗДАНИЯ ПЕРЕКРЕСТКОВ

3. Перекресток для слияния «И» не может следовать за перекрестком типа исключительного «ИЛИ»

ПРАВИЛА СОЗДАНИЯ ПЕРЕКРЕСТКОВ

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.



ПРИМЕРЫ



ПРИМЕРЫ



ПРИМЕРЫ

КОМБИНАЦИИ ПЕРЕКРЕСТКОВ

- Перекрестки могут комбинироваться для создания сложных соединений

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



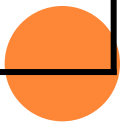
ОБЪЕКТ ССЫЛОК

- *Официальная спецификация IDEF3 различает 3 стиля объектов ссылок – безусловные (unconditional), синхронные (synchronous), асинхронные (asynchronous).*
- ВРWin поддерживает только *безусловные* объекты ссылок.




ТИПЫ ОБЪЕКТОВ ССЫЛОК

Тип объекта ссылок	Назначение
1. Object	Используется для описания того, что в действии принимает участие какой-либо заслуживающий отдельного внимания объект
2. Ссылка GOTO	Используется для реализации цикличности выполнения действий. Этот объект также может относиться к перекрестку
3. Единица действий UOB (Unit of Behavior)	Используется для многократного отображения на диаграмме одного и того же действия, но без цикла



ТИПЫ ОБЪЕКТОВ ССЫЛОК

Тип объекта ССЫЛОК	Назначение
4. <i>Заметка</i> (Note)	Используется для документирования какой-либо важной информации общего характера, относящейся к изображаемому на диаграммах. Служит альтернативой методу помещения текстовых заметок непосредственно на диаграммах
5. <i>Уточнение</i> <i>Elaboration</i> (ELAB)	Для уточнения или более подробного описания изображаемого на диаграмме. Обычно используется для детального описания разветвления или слияния стрелок на перекрестках



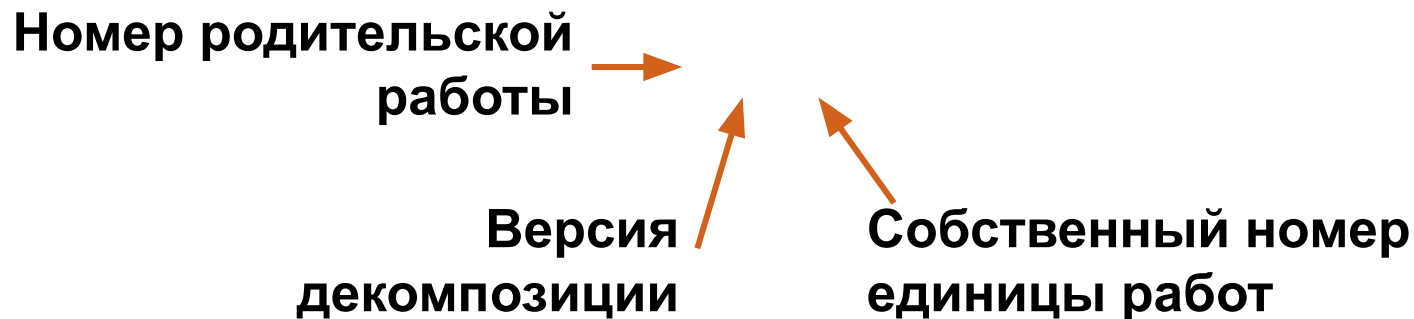
ДЕКОМПОЗИЦИЯ РАБОТ В IDEF3

- В IDEF3 декомпозиция используется для *детализации* работ.
- Методология IDEF3 позволяет декомпонировать работу **многократно**, 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



НУМЕРАЦИЯ РАБОТ В IDEF3

- Номер работы состоит из *номера родительской работы*, *версии декомпозиции* и *собственного номера* работы на текущей диаграмме



СТРУКТУРА МНОЖЕСТВЕННОЙ ДЕКОМПОЗИЦИИ РАБОТ

Первая
декомпозиция
работы 1.2

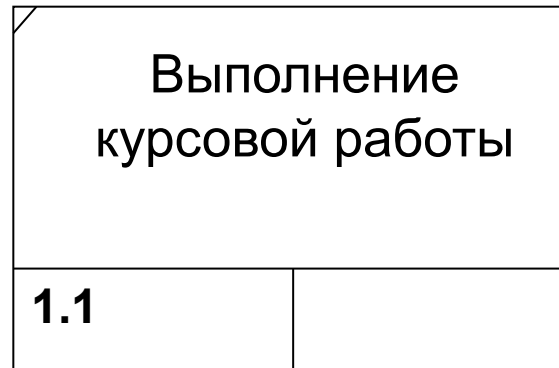


Вторая
декомпозиция
работы 1.2



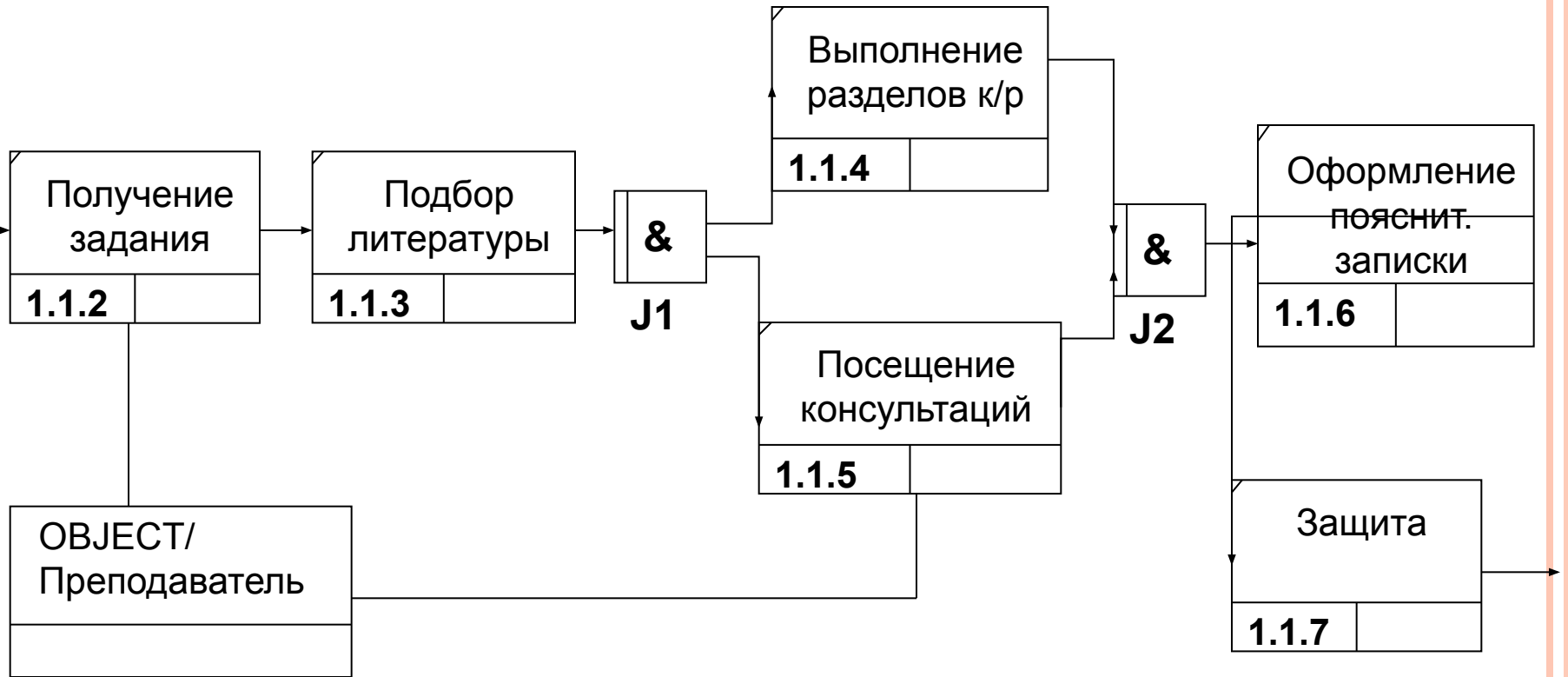
ПРИМЕР ПОСТРОЕНИЯ МОДЕЛИ IDEF3

- Рассмотрим на примере построения динамической модели процесса **«Выполнение курсовой работы»**
- Начнем с построения контекстной диаграммы



ПРИМЕР ПОСТРОЕНИЯ МОДЕЛИ IDEF3

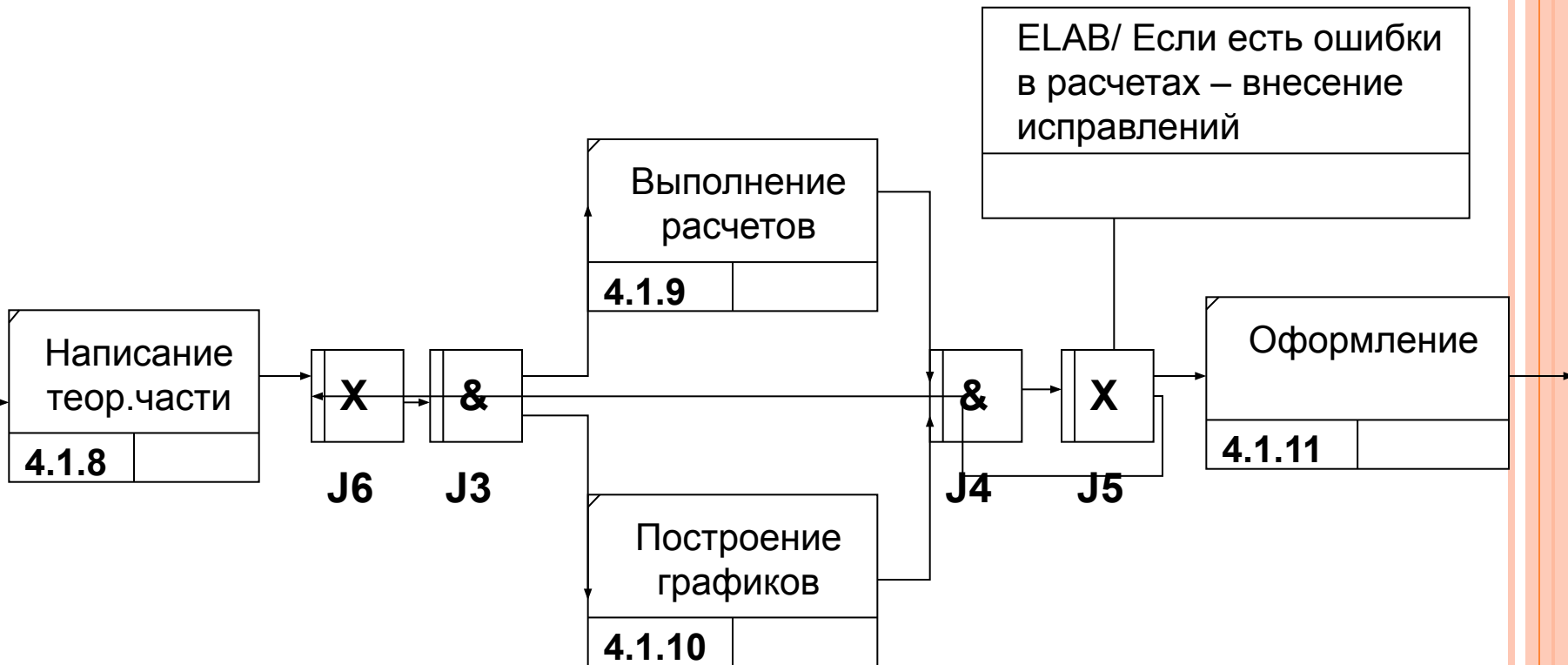
Выполним декомпозицию контекстной диаграммы:



Примечание: Обратите внимание на нумерацию единиц работ. Родительской является работа с собственным номером **1**. Она декомпозируется первый раз, следовательно, версия декомпозиции = **1**, далее следует собственный номер единицы работ в рамках модели (**2-7**).

ПРИМЕР ПОСТРОЕНИЯ МОДЕЛИ IDEF3

Выполним декомпозицию UOW №4 – «Выполнение разделов к/р»



ПРИМЕР ПОСТРОЕНИЯ МОДЕЛИ IDEF3

- Продекомпозируем повторно контекстную диаграмму (в виде **сценария IDEF3** для выполнения курсовой работы по «Информатике и программированию»)



ПОСТРОЕНИЕ ИНФОРМАЦИОННОЙ МОДЕЛИ ПРОЦЕССА ПОСТРОЙКИ САДОВОГО ДОМИКА

1. На основе функциональной модели IDEF0 составим пул – список потенциальных сущностей.

□ **Пул:**

1. Дом
2. Крыша
3. Материалы
4. Проект дома
5. Стены
6. Строители
7. Фундамент
8. Каменщики
9. Плотники
10. Кровельщики
11. Мастера по отделке



МОДЕЛИ ПРОЦЕССА ПОСТРОЙКИ САДОВОГО ДОМИКА

2. Определим сущности



МОДЕЛИ ПРОЦЕССА ПОСТРОЙКИ САДОВОГО ДОМИКА