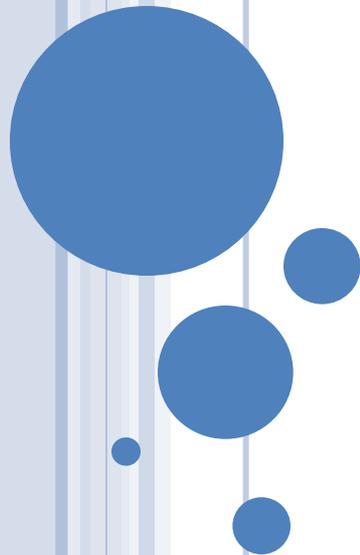


**ТЕМА**  
**РАЗРАБОТКА И ЭКСПЛУАТАЦИЯ**  
**УДАЛЕННЫХ БАЗ ДАННЫХ**



**□ Лекция 5. Классификация инструментальных средств проектирования структуры базы данных**



## ▣ МЕТОДОЛОГИИ СТРУКТУРНОГО МОДЕЛИРОВАНИЯ

- ▣ Диаграммы потоков данных.
- ▣ Нотация Йордана–де Марко.
- ▣ Нотация Гейна–Сарсона
- ▣ Методология SADT (IDEF0)

## ▣ МЕТОДОЛОГИИ ИНФОРМАЦИОННОГО МОДЕЛИРОВАНИЯ

- ▣ Нотация Чена
- ▣ Нотация Мартина
- ▣ Нотация IDEF1X
- ▣ Нотация Баркера
- ▣ Язык инфологического моделирования



## ▣ МЕТОДОЛОГИИ СТРУКТУРНОГО МОДЕЛИРОВАНИЯ

- ▣ **Структурный анализ** – это систематический пошаговый подход к анализу требований и проектированию спецификаций системы независимо от того, является ли она существующей или создается ВНОВЬ.



- Методология структурного анализа и проектирования определяет шаги работы, которые должны быть выполнены, их последовательность, правила распределения и назначения операций и методов.



- В настоящее время успешно используются такие методологии, как структурный системный анализ *Гейна–Сарсона*, структурный анализ и проектирование *Йордана–де Марко*, *SADT* и другие



- **1. Диаграммы потоков данных.**
- **Нотация Йордана–де Марко.**
- **Нотация Гейна–Сарсона**



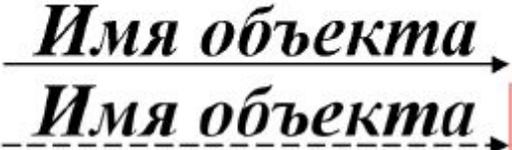
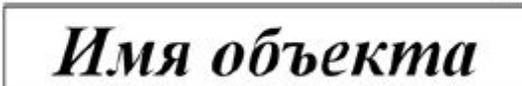
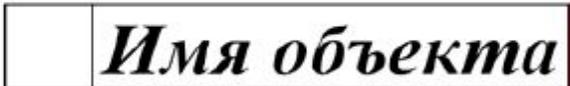
- **Диаграммы потоков данных (*DFD – Data Flow Diagram*)** являются основным средством моделирования функциональных требований проектируемой системы.
- С их помощью эти требования разбиваются на функциональные компоненты (процессы) и представляются в виде сети, связанной потоками данных.
- **Главная цель таких средств** – продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.



- Для изображения *DFD* традиционно используются две различные нотации: Йордана–де Марко и Гейна–Сарсона.
- Диаграммы потоков данных (*DFD – Data Flow Diagramm*) строятся из элементов, представленных в *таблице*



## Конструктивные элементы диаграммы потоков данных

Элемент	Нотация Йордона–де Марко	Гейна-Сарсона
Функция		
Поток данных		
Хранилище данных		
Внешняя сущность		



- *Потоки данных* являются механизмами, использующимися для моделирования передачи информации (или даже физических компонент) из одной части системы в другую.
- Назначение *Функции* состоит в продуцировании выходных потоков из входных в соответствии с действием, задаваемым именем функции.
- *Это имя должно содержать глагол в неопределенной форме с последующим дополнением (например, **ВЫЧИСЛИТЬ МАКСИМАЛЬНУЮ ВЫСОТУ**).*



- ▣ *Хранилище* позволяет на определенных участках определять данные, которые будут сохраняться в памяти между процессами.
- ▣ *Внешняя сущность* представляет сущность вне контекста системы, являющуюся источником или приемником системных данных.
- ▣ Ее имя должно содержать существительное, например, **СКЛАД ТОВАРОВ**

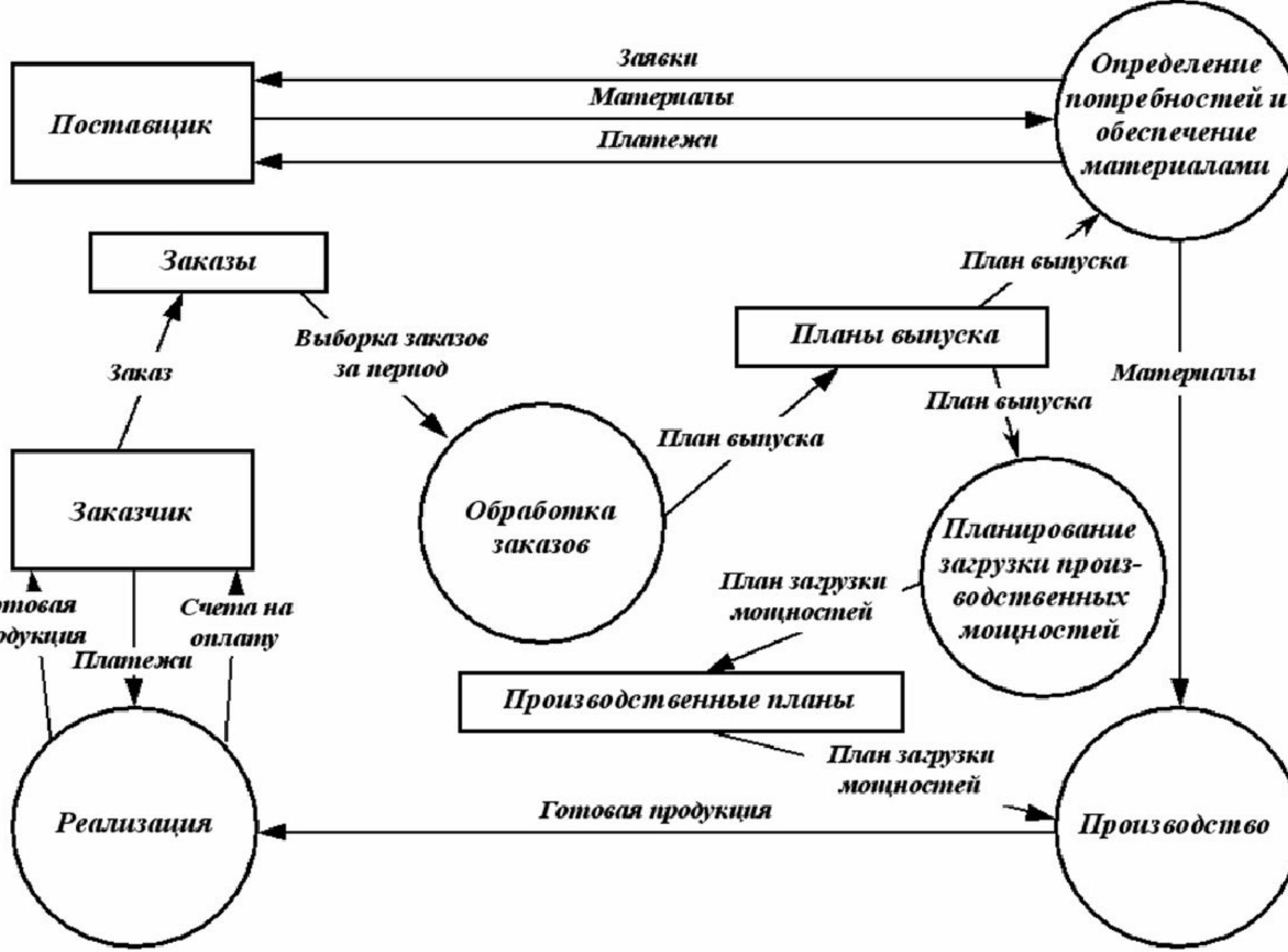


- При интерпретации *DFD*-диаграммы **используются следующие правила:**
- – функции преобразуют входящие потоки данных в выходящие;
- – хранилища данных не изменяют потоки данных, а служат только для хранения поступающих объектов;
- – преобразования потоков данных во внешних сущностях игнорируются.



- Помимо этого, для каждого информационного потока и хранилища определяются связанные с ними элементы данных.
- Каждому элементу данных присваивается имя, также для него может быть указан тип данных и формат.
- Именно эта информация является исходной на следующем этапе проектирования – построении модели «*сущность-связь*».

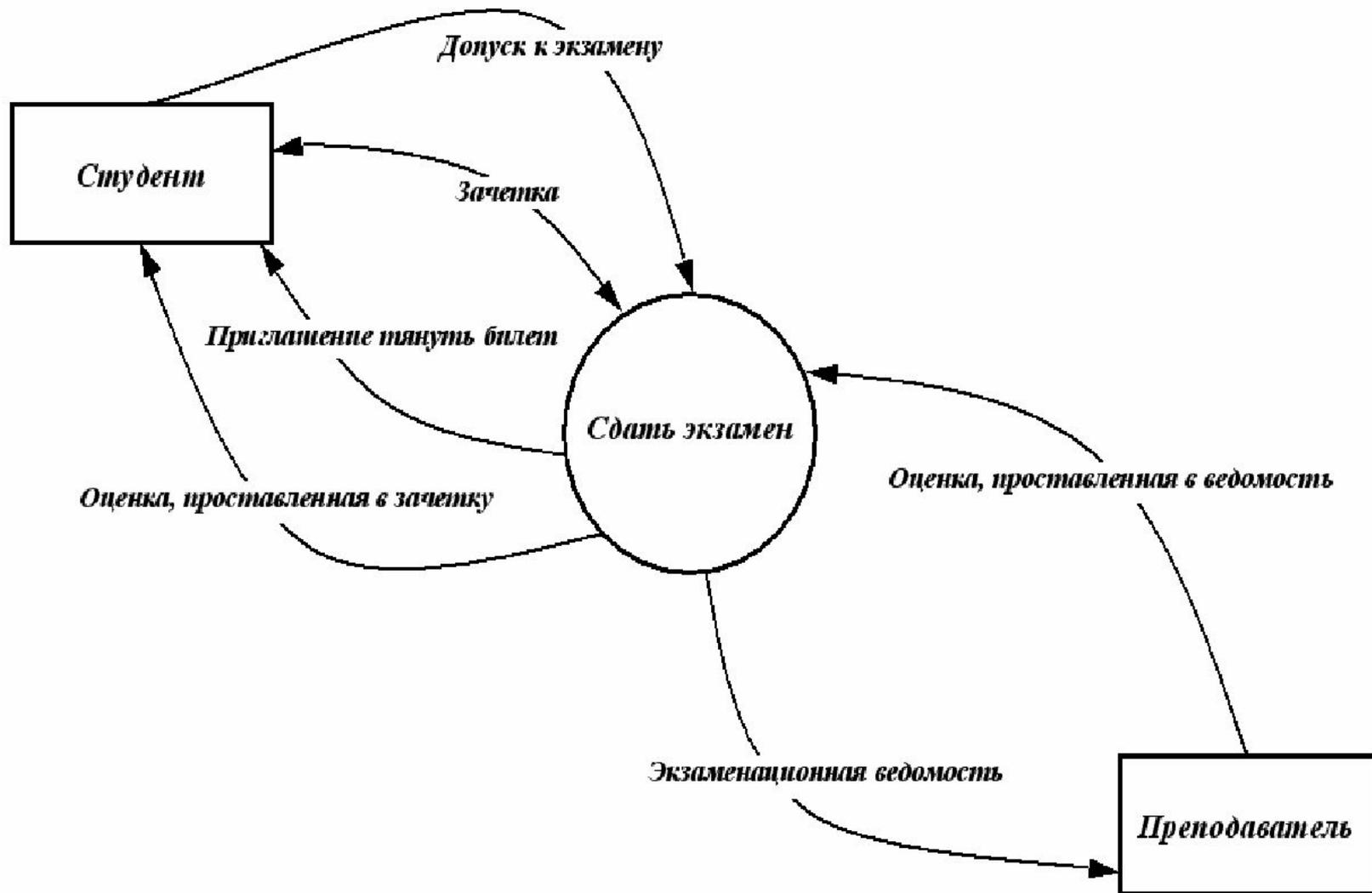




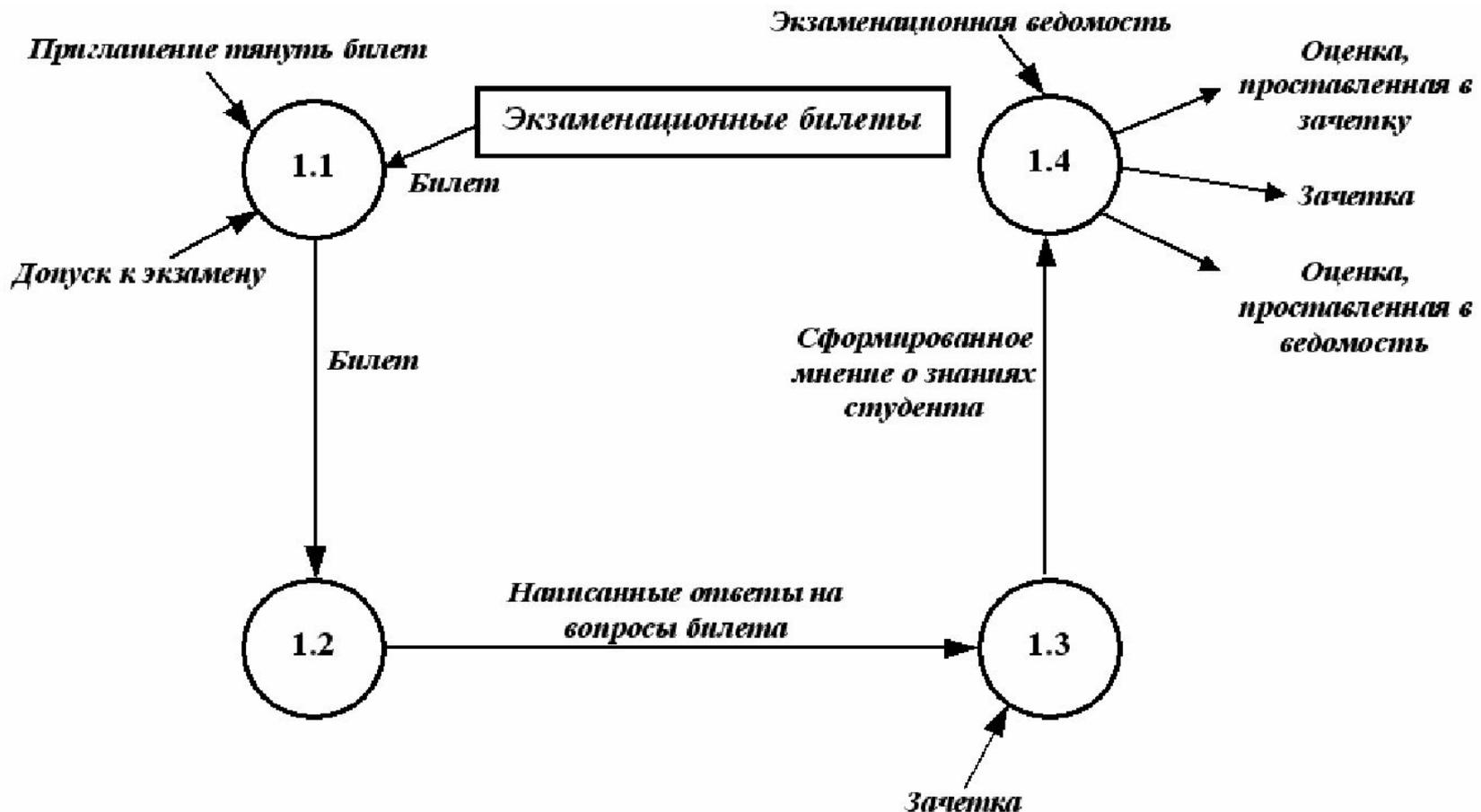
- **Пример 2.** Рассмотрим процесс **СДАТЬ ЭКЗАМЕН**. У нас есть две сущности **СТУДЕНТ** и **ПРЕПОДАВАТЕЛЬ**. Опишем потоки данных, которыми обменивается наша проектируемая система с внешними объектами (*рис. 2*).
- Со стороны сущности **СТУДЕНТ** опишем информационные потоки.
- Для сдачи экзамена необходимо, чтобы у **СТУДЕНТА** была **ЗАЧЕТКА**, а также чтобы он имел **ДОПУСК К ЭКЗАМЕНУ**. Результатом сдачи экзамена, т.е. выходными потоками будут **ОЦЕНКА ЗА ЭКЗАМЕН** и **ЗАЧЕТКА**, в которую будет проставлена **ОЦЕНКА**.
- Со стороны сущности **ПРЕПОДАВАТЕЛЬ** информационные потоки следующие: **ЭКЗАМЕНАЦИОННАЯ ВЕДОМОСТЬ**, согласно которой будет известно, что **СТУДЕНТ** допущен до экзамена, а также официальная бумага, куда будет занесен результат экзамена, т.е. **ОЦЕНКА ЗА ЭКЗАМЕН**, проставленная в **ЭКЗАМЕНАЦИОННУЮ ВЕДОМОСТЬ**.



# РИС. 2 DFD-ДИАГРАММА ПРОЦЕССА «СДАТЬ ЭКЗАМЕН»



- Теперь детализируем процесс **СДАЧА ЭКЗАМЕНА**. Этот процесс будет содержать следующие процессы (рис. 3): **ВЫТЯНУТЬ БИЛЕТ** {1.1}, **ПОДГОТОВИТЬСЯ К ОТВЕТУ** {1.2}, **ОТВЕТИТЬ НА БИЛЕТ** {1.3}, **ПРОСТАВЛЕНИЕ ОЦЕНКИ** {1.4}.
- Рис. 3 Декомпозиция 1-го уровня DFD-диаграммы процесса «Сдать экзамен»



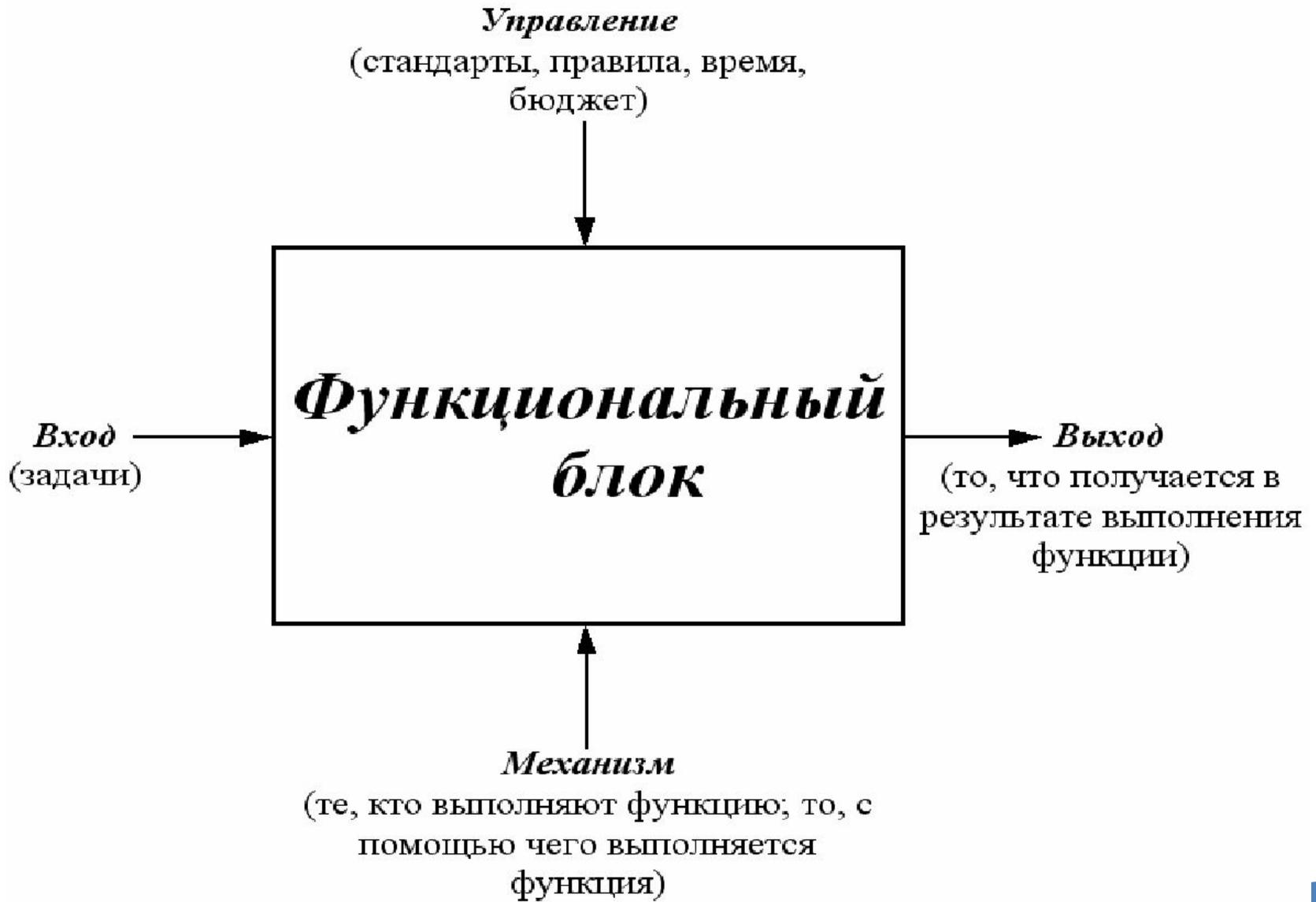
## 2. МЕТОДОЛОГИЯ SADT (IDEF0)

- Методология *SADT* (*Structured Analysis and Design Technique*) разработана Дугласом Т. Россом в 1969–73 годах.



- Она изначально создавалась для проектирования систем более общего назначения по сравнению с другими структурными методами, выросшими из проектирования программного обеспечения.
- ***IDEFO*** (подмножество *SADT*) используется для моделирования бизнес-процессов в организационных системах и имеет развитые процедуры поддержки коллективной работы.





- **Правила интерпретации модели:**
- – функциональный блок (функция) преобразует входные объекты в выходные;
- – управление определяет, когда и как это преобразование может или должно произойти;
- – механизм (исполнитель) осуществляет это преобразование.



- Каждый блок *IDEFO*-диаграммы может быть представлен несколькими блоками, соединенными интерфейсными дугами, на диаграмме следующего уровня.
- Эти блоки представляют подфункции (подмодули) исходной функции.
- Каждый из подмодулей может быть декомпозирован аналогичным образом.

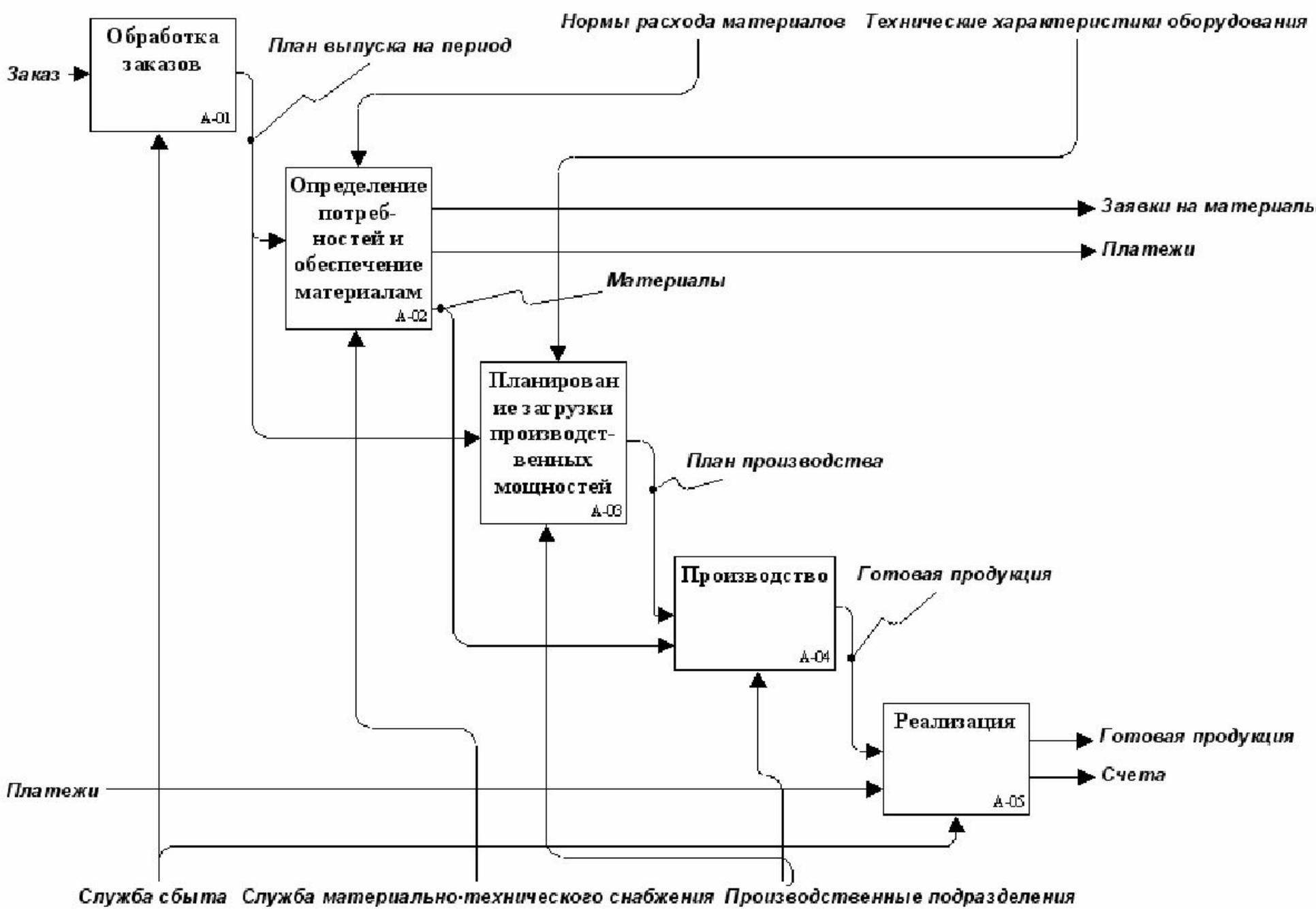


- Методология *SADT* может использоваться для моделирования широкого круга систем и определения требований и функций, а затем для разработки системы, которая удовлетворяет этим требованиям и реализует эти функции.
- Для уже существующих систем *SADT* может быть использована для анализа функций, выполняемых системой, а также для указания механизмов, посредством которых они осуществляются.



- Результатом применения методологии *SADT* является модель, которая состоит из диаграмм, фрагментов текстов и глоссария, имеющих ссылки друг на друга.





## ▣ МЕТОДОЛОГИИ ИНФОРМАЦИОННОГО МОДЕЛИРОВАНИЯ

- ▣ Важнейшая цель информационной модели заключается в выработке непротиворечивой интерпретации данных и взаимодействий между ними с тем, что необходимо для интеграции, совместного использования и управления целостностью данных.



- В 1983 году в рамках проекта военного ведомства США «Интегрированные системы информационной поддержки» (*ISAM*) была создана методология семантического моделирования данных *IDEF1X* (расширение методологии *IDEF1*), позволяющая логически объединять в сеть неоднородные вычислительные системы.



- **Методология *IDEF1X*** – один из подходов к семантическому моделированию данных, основанный на концепции *Сущность-связи (Entity- Relationship)*, это инструмент для анализа информационной структуры систем различной природы.



- Методология *IDEF1X* предназначена для построения концептуальной схемы реляционной базы данных, которая была бы независимой от программной платформы её конечной реализации.
- Эта информация является необходимым дополнением функциональной *IDEF0*-модели, детализирует объекты, которыми манипулируют функции системы.
- Концептуально *IDEF1X*-модель можно рассматривать как проект логической схемы базы данных для проектируемой системы.



- *IDEF1X* использует понятия сущностей, атрибутов, отношений и ключей.
- Стандарт и методология *IDEF1X* является специализированным инструментом, предназначенным для разработчиков реляционных баз данных.



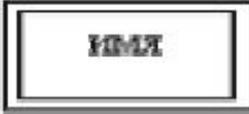
- Наибольшее распространение получили следующие нотации, используемые при построении *ER*-диаграмм: нотация *Чена*, нотация *Мартина*, нотация *IDEF1X*, нотация *Баркера*.



## 1. Нотация Чена

- В *таблице* приведены конструктивные элементы концептуальной схемы базы данных в нотации Чена.

Конструктивные элементы нотации Чена

№	Элемент диаграммы	Обозначает	№	Элемент диаграммы	Обозначает
1		независимая сущность	6		атрибут
2		зависимая сущность	7		первичный ключ
3		родительская сущность в иерархической связи	8		внешний ключ (понятие внешнего ключа вводится в реляционной модели данных)

4		СВЯЗЬ	9		многозначный атрибут
5		идентифицирующая связь	10		получаемый (наследуемый) атрибут в ие- рархических связях



- Связь соединяется с ассоциируемыми сущностями линиями. Возле каждой сущности на линии, соединяющей ее со связью, цифрами указывается класс принадлежности (*рис.*)

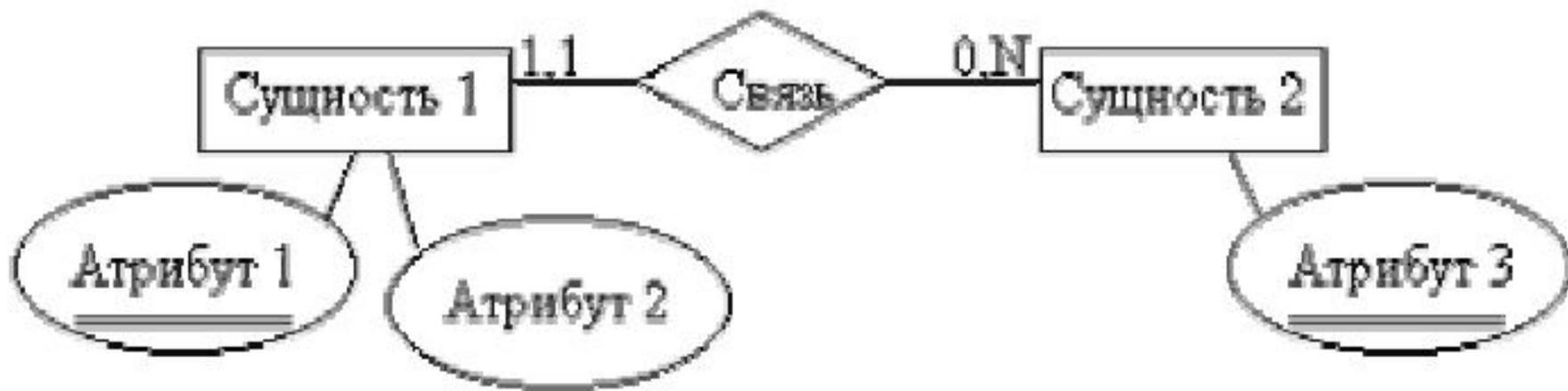
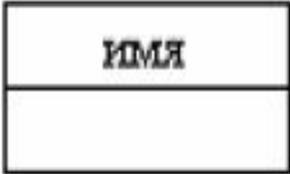


Рис. 6. Пример фрагмента концептуальной схемы в нотации Чена

## 2. НОТАЦИЯ МАРТИНА

### Конструктивные элементы нотации Мартина

Элемент диаграммы	Обозначает
	независимая сущность
	зависимая сущность
	родительская сущность в иерархической связи

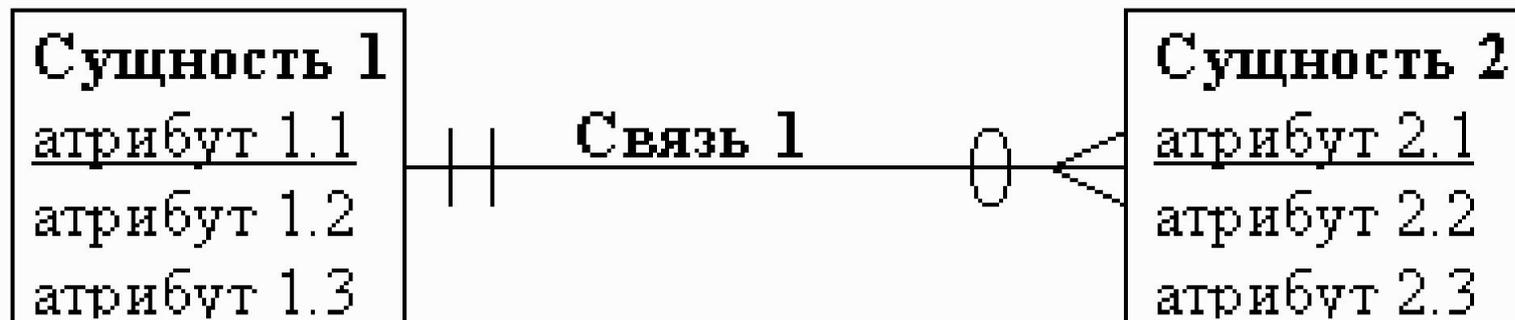
- Список атрибутов приводится внутри прямоугольника, обозначающего сущность.
- Ключевые атрибуты подчеркиваются.
- Связи изображаются линиями, соединяющими сущности, вид линии в месте соединения с сущностью определяет кардинальность связи



### Обозначения связей в нотации Мартина

№	Обозначение	Кардиналь- ность	№	Обозначение	Кардиналь- ность
1		нет	4		M,N
2		1,1	5		0,N
3		0,1	6		1,N





Имя связи указывается на линии ее обозначающей



### 3. НОТАЦИЯ IDEF1X

Таблица 7

Обозначения сущностей в нотации IDEF1X

Элемент диаграммы	Обозначает
	независимая сущность
	зависимая сущность

Таблица 8

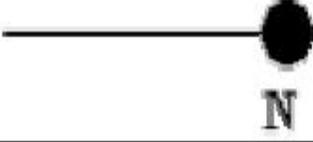
Обозначения связей в нотации IDEF1X

Элемент диаграммы	Обозначает
	Идентифицирующая связь
	Неидентифицирующая связь



Таблица 9

Обозначения кардинальностей связей в нотации IDEF1X

Элемент диаграммы	Обозначает
	1,1
	0,М
	0,1
	1,М
	точно $N$ ( $N$ - произвольное число)



- Список атрибутов приводится внутри прямоугольника, обозначающего сущность.
- Атрибуты, составляющие ключ сущности, группируются в верхней части прямоугольника и отделяются горизонтальной чертой

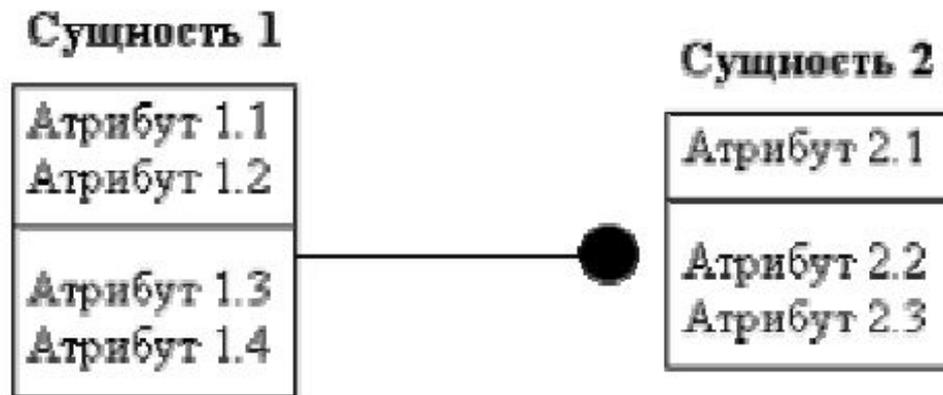
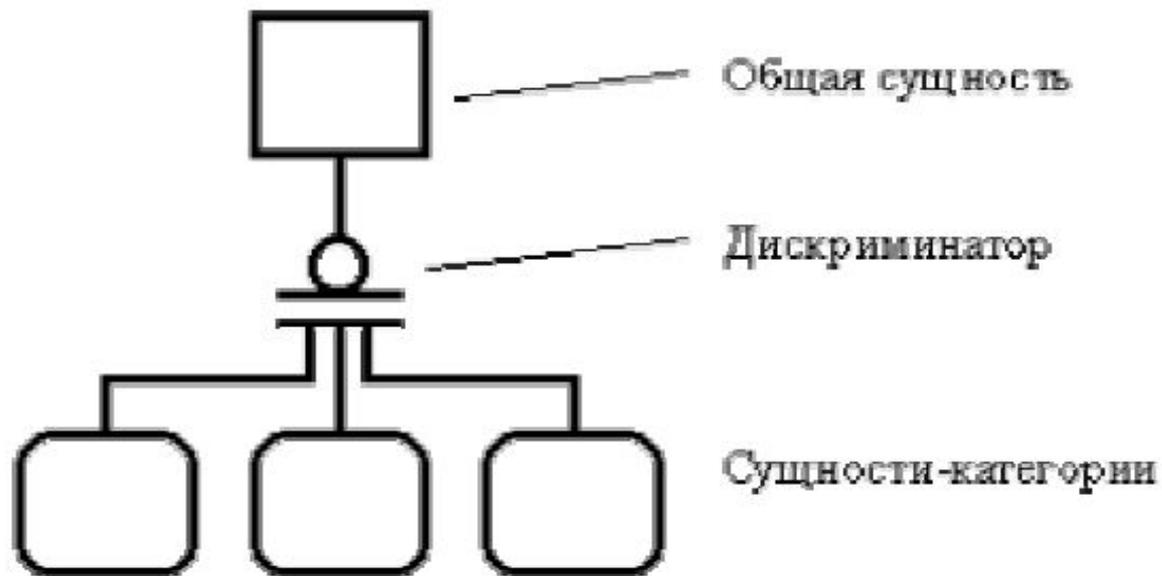


Рис. 8. Пример фрагмента концептуальной схемы в нотации IDEF1X



- Кроме того, в IDEF1X вводится понятие *отношение категоризации*, по смыслу эквивалентное рассмотренной нами иерархической связи.



**Рис. 9. Отношение полной категоризации в нотации IDEF1X**



#### □ 4. Нотация Баркера

- Сущности обозначаются прямоугольниками, внутри которых приводится список атрибутов.
- Ключевые атрибуты отмечаются символом # (решетка).
- Связи обозначаются линиями с именами, место соединения связи и сущности определяет кардинальность связи

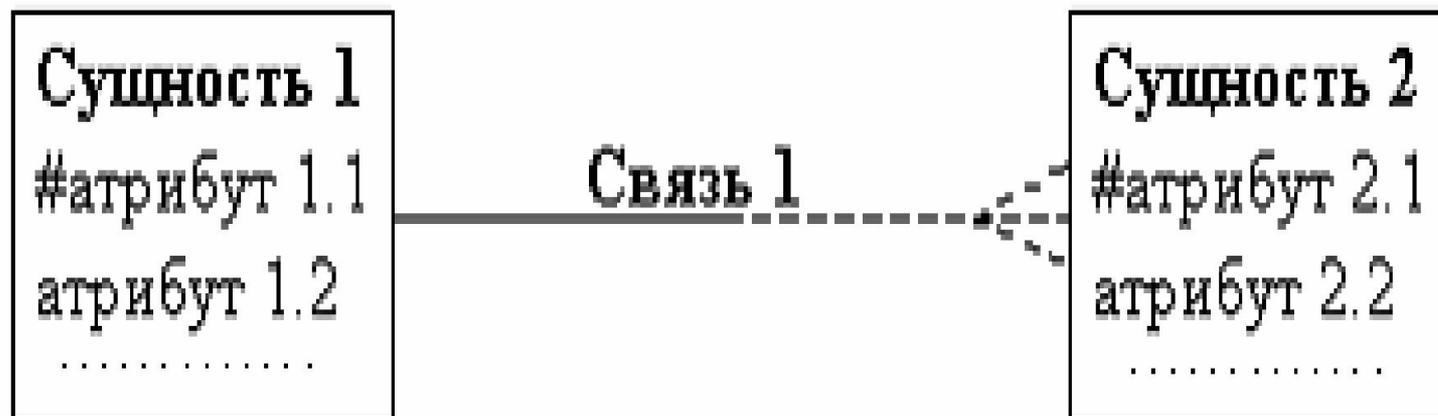


Рис. 11. Пример фрагмента концептуальной схемы в нотации Баркера



- Для обозначения отношения категоризации вводится элемент «дуга»

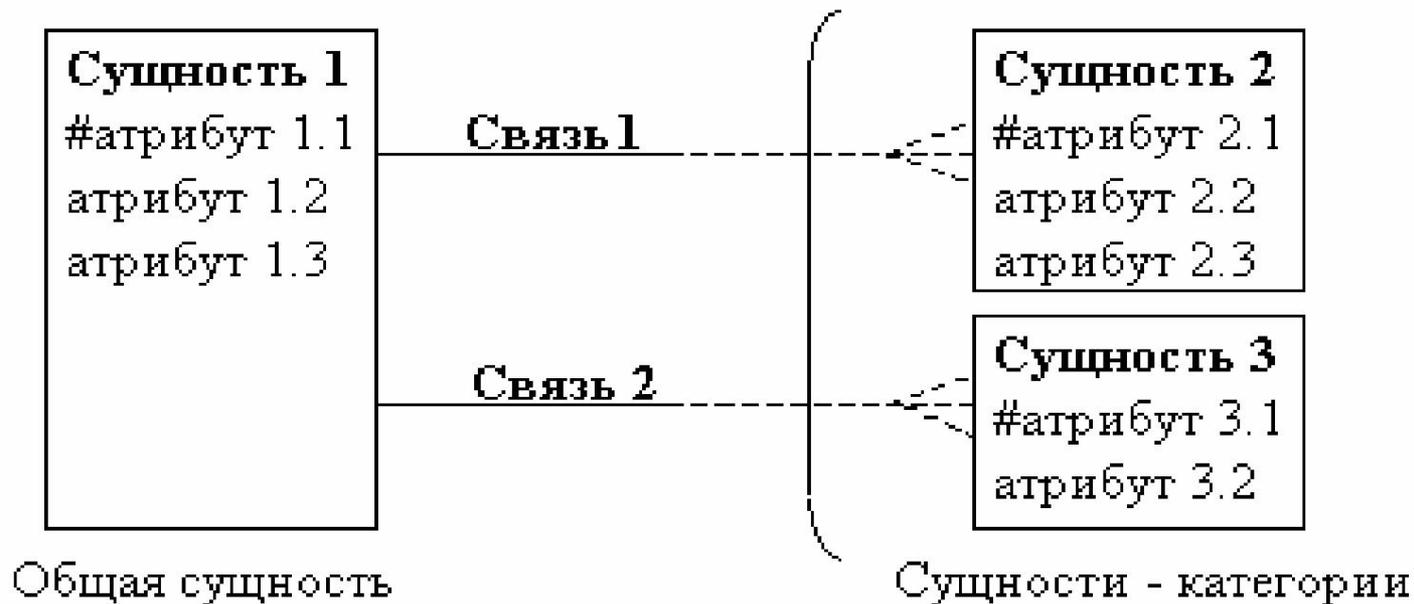
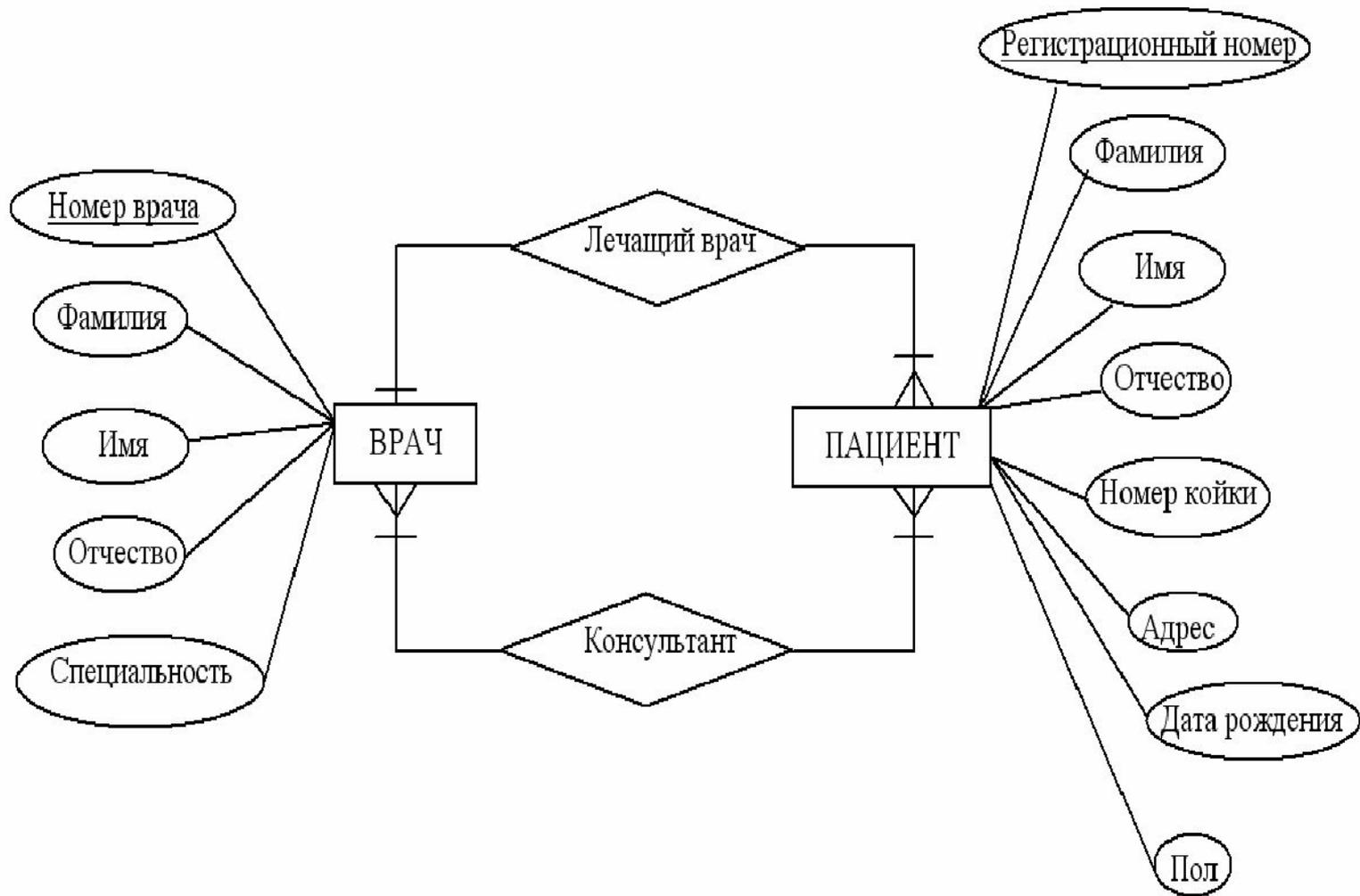


Рис. Обозначение категоризации в нотации Баркера



- **5. Язык инфологического моделирования**
- Язык *ER*-диаграмм используется для построения небольших моделей и иллюстрации отдельных фрагментов больших.
- Чаще же применяется менее наглядный, но более содержательный *язык концептуального моделирования (ЯКМ)*, в котором сущности и связи представляются предложениями вида:
  - **СУЩНОСТЬ (атрибут 1, атрибут 2 , ..., атрибут n)**
  - **СВЯЗЬ [СУЩНОСТЬ S1, СУЩНОСТЬ S2, ...]**
  - **(атрибут 1, атрибут 2, ..., атрибут n),**
- где *S* – степень связи, а атрибуты, входящие в ключ, должны быть отмечены с помощью подчеркивания.





**Рис. Фрагмент концептуальной схемы в нотации Чена**



- Так, рассмотренный выше пример множества связей между сущностями , может быть описан на *ЯКМ* следующим образом:
  
- **Врач** (Номер\_врача, Фамилия, Имя, Отчество, Специальность)
- **Пациент** (Регистрационный\_номер, Номер койки, Фамилия, Имя, Отчество, Адрес, Дата рождения, Пол)
- **Лечащий\_врач** [Врач (1,1), Пациент (1,М)]
- (Номер\_врача, Регистрационный\_номер)
- **Консультант** [Врач (1,М), Пациент (1, N)]
- (Номер\_врача, Регистрационный\_номер).

