

**Университет машиностроения**

**Кафедра «Автоматика и процессы управления»**

**Дисциплина**

**Информационные технологии**

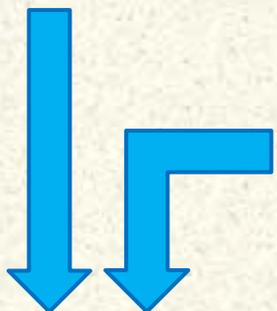
**2 семестр**

**Тема 11**

**Моделирование  
информационных систем  
управления**

# Моделирование системы управления

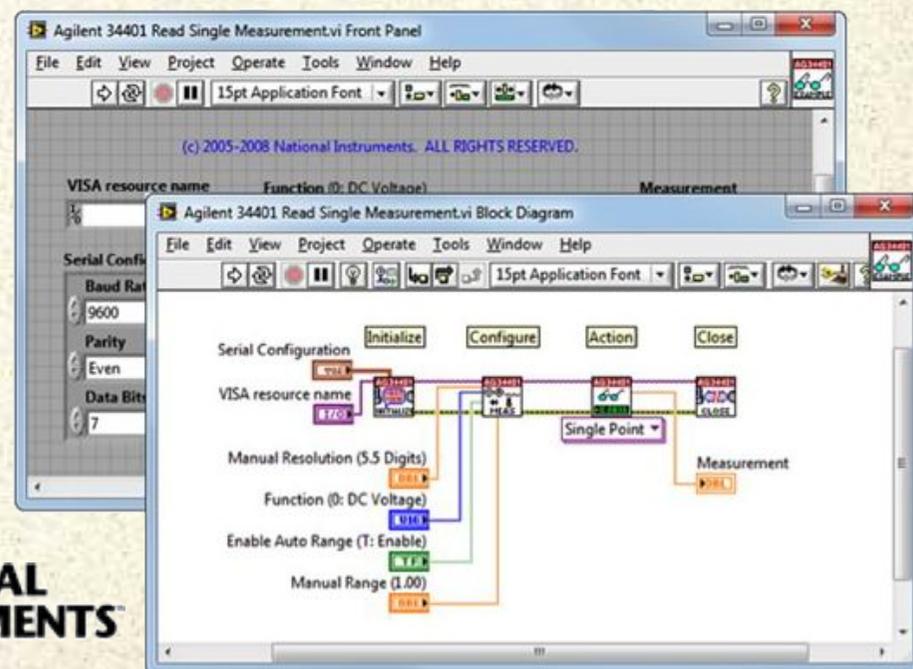
ТАУ (Теория Автоматического Управления) описывает принципы, законы и алгоритмы, лежащие в основе процессов управления техническими системами



Графическое моделирование помогает «увидеть» работу будущей системы, структура которой выражена в блоках и линиях

Графические языки визуального программирования

**NI LabView** — модульная программная среда для разработки приложений по обработке данных и проектированию логики встраиваемых систем от National Instruments



# Графическое моделирование

Описание информационной системы (в том числе АСУТП) с помощью языка графических примитивов, обозначающих конкретную функцию, элемент, действие или состояние системы

- ➔ Структурные диаграммы (блок-схемы)
- ➔ Метод CRC-карточек
- ➔ Техника структурного анализа SADT
  - ➔ Функциональное моделирование IDEF0
  - ➔ Моделирование структур данных IDEF1 / 1X
- ➔ Язык UML

**Структурные диаграммы**

**Элементы и  
правила  
построения  
блок-схем**

## Определение

**Блок-схема** является формой представления алгоритма с помощью графических символов. Графические символы, их размеры, а также правила построения блок-схем определены государственными стандартами.

**ГОСТ 19.701-90.** Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.

**ГОСТ 19.002-80.** Схемы алгоритмов и программ. Правила выполнения.

**ГОСТ 19.003-80.** Схемы алгоритмов и программ. Обозначения условные графические

# Элементы блок-схем (международная традиция)

**Процесс.** Выполнение операции или группы операций, в результате чего изменяется значение, форма представления или расположения данных. Внутри символа или же в виде комментария на естественном языке или в виде формулы записываются действия, которые производятся при выполнении операции или группы операций.



**Предопределенный процесс.** Использование ранее созданных и отдельно описанных алгоритмов или программ (процедур, функций, программных модулей). Символ служит для указания обращения к процедурам, функциям, программным модулям.



# Элементы блок-схем (продолжение)

**Решение.** Выбор направления выполнения алгоритма или программы в зависимости от некоторых переменных условий.

Символ используется для изображения унифицированных структур:

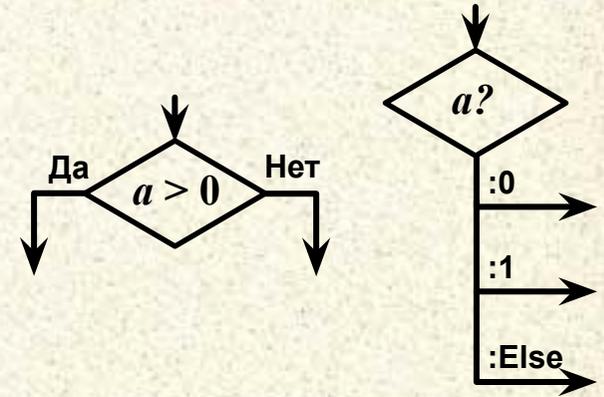
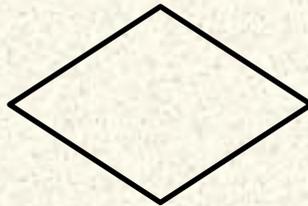
РАЗВИЛКА ПОЛНАЯ

РАЗВИЛКА НЕПОЛНАЯ

ВЫБОР

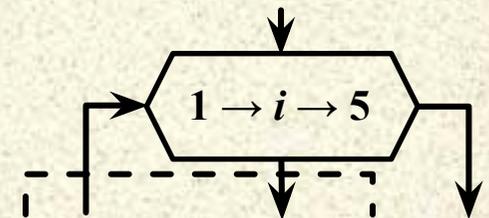
ЦИКЛ-ДО

ЦИКЛ-ПОКА



**Модификация.** Выполнение операций, меняющих команды или группу команд, изменяющих программу.

Символ используется для изображения унифицированной структуры ЦИКЛ С ПАРАМЕТРОМ. Внутри символа записывается параметр цикла с указанием начального и конечного значений, а также шаг изменения цикла, если он не равен единице.

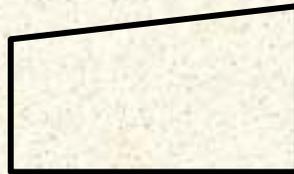


# Элементы блок-схем (продолжение)

**Ввод-вывод** . Операция обмена данными с внешним устройством хранения, ввода-вывода, базой данных, элементами управления или временным хранилищем в оперативной памяти.



**Ручной ввод**. Ввод данных оператором в процесс обработки при помощи устройства, непосредственно сопряженного с компьютером (например, клавиатура).

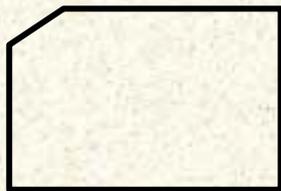


**Дисплей**. Ввод-вывод данных в случае, если непосредственно подключенное к процессору устройство воспроизводит данные и позволяет оператору вносить изменения в процессе их обработки.

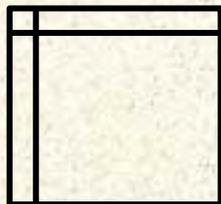


# Элементы блок-схем (продолжение)

**Карта.** (*устар.*) Ввод-вывод данных с использованием в качестве носителя малой ёмкости, стандартизированного дизайна и, как правило, однократной записи – перфоркарты, RFID-метик или штрих-кода.



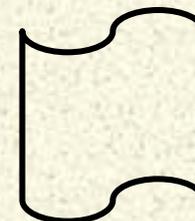
**Память.** Ввод - вывод данных в хранилище, организованное в оперативной памяти или на ином устройстве, допускающим произвольный доступ



**Документ.** Ввод - вывод данных, носителем которых служит бумага.



**Лента.** (*устар.*) Ввод-вывод данных в «человеко-читаемом» виде на носитель или устройство, имитирующее бумажную ленту (перфоленту, кассовую ленту, стример, «бегущую строку», последовательный канал передачи данных).



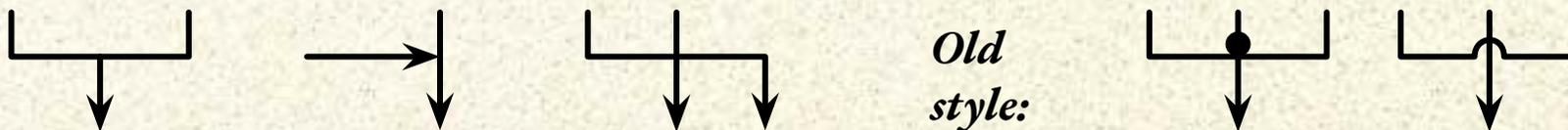
# Элементы блок-схем (продолжение)

**Линия потока.** Указание последовательности связей между символами.



## **Правила изображения линий потока:**

- 1) линии потока должны быть параллельны линиям внешней рамки блок-схемы (границам листа, на котором изображена блок-схема);
- 2) направление линии потока сверху вниз и слева направо принимается за основное и стрелками не обозначается, в остальных случаях направление линии потока обозначается стрелками;
- 3) изменение направления линии потока производится под углом 90 градусов;
- 4) слияние (объединение) линий (кроме тривиальных случаев) должно обозначаться **узлом** или **концевыми стрелками** на линиях потоков.



# Элементы блок-схем (продолжение)

**Соединитель.** Указание связи между прерванными линиями потока, связывающими символы. Если блок-схема состоит из нескольких частей, расположенных на одной странице, то линия потока одной части заканчивается символом СОЕДИНИТЕЛЬ, а линия потока на продолжении блок-схемы начинается с этого же символа. Внутри символов СОЕДИНИТЕЛЬ ставятся одинаковые порядковые номера, соответствующие разорванной линии потока

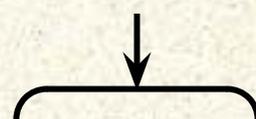
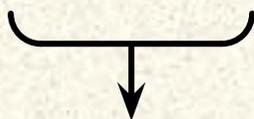


**Межстраничный соединитель.** Указание связи между разъединенными частями схем алгоритмов и программ, расположенных на разных листах. Данный символ служит для тех же целей, что и соединитель, но при расположении частей блок-схемы на разных страницах.

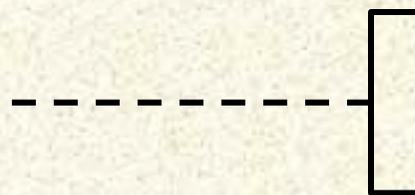


# Элементы блок-схем (окончание)

**Пуск - останов.** Начало, конец, прерывание процесса обработки данных или выполнения программы.



**Комментарий.** Связь между элементами схемы и пояснениями. Позволяет включать в блок-схему пояснения, формулы и другую информацию.



**Метод CRC-карточек**

**Class-Responsibility-Collaboration**

**Класс-Взаимодействия-Сотрудничество**

**или**

**Класс-Ответственность-Связи**

# Шаблон CRC-карточки

Class Name	
Responsibilities	Collaborators

## Пример CRC-карточки в варианте «Класс-Ответственность-Связи»

Контролер	
Обработка ввода Распределение управления	Вид Модель

# Пример CRC-модели

Enrollment	
Mark(s) received Average to date Final grade Student Seminar	Seminar

Transcript	
**See the prototype** Determine average mark	Student Seminar Professor Enrollment

Student Schedule	
**See the prototype**	Seminar Professor Student Enrollment Room

Room	
Building Room number Type (Lab, class, ...) Number of Seats Get building name Provide available time slots	Building

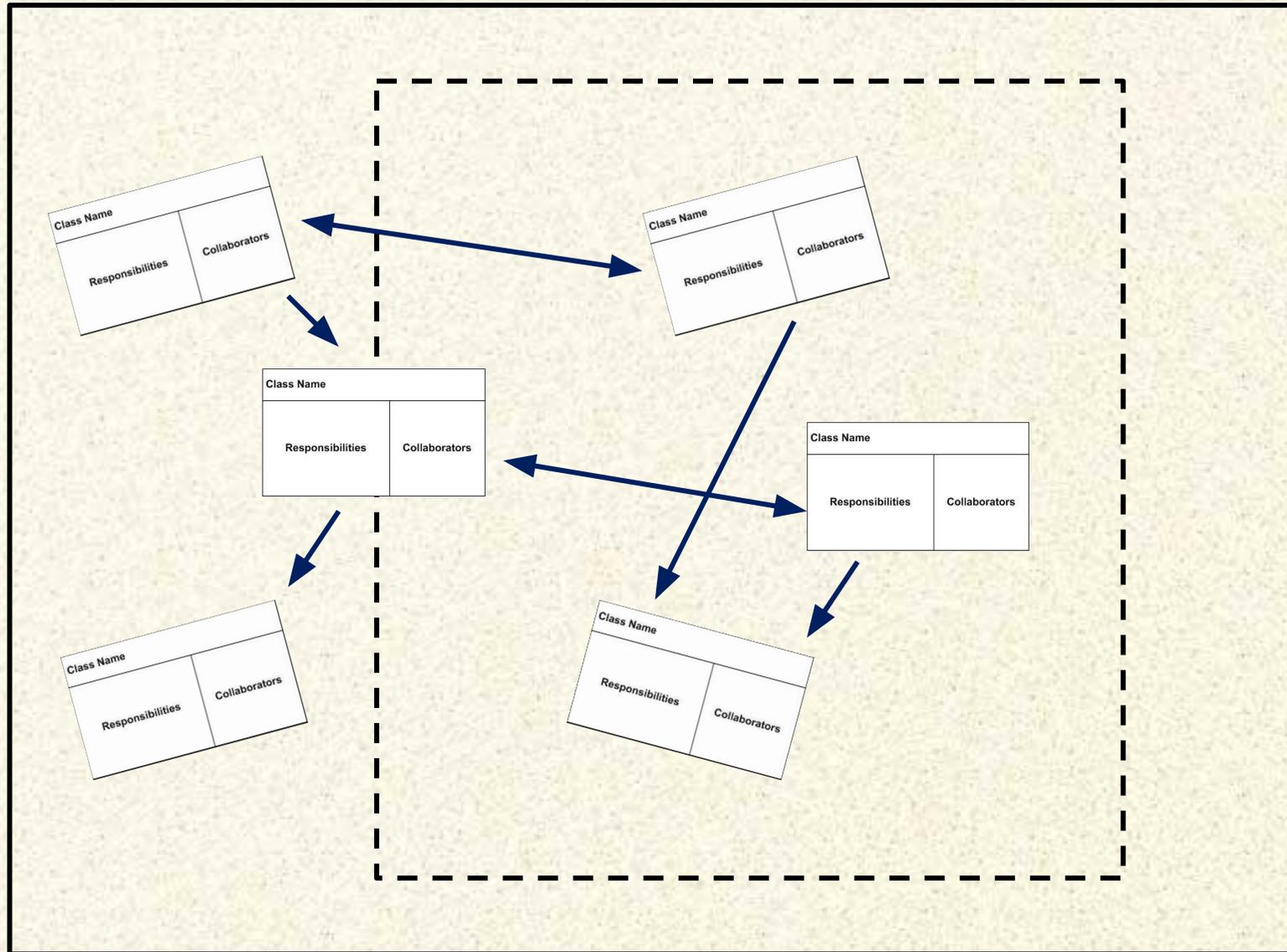
Professor	
Name Address Phone number Email address Salary Provide information Seminars instructing	Seminar

Seminar	
Name Seminar number Fees Waiting list Enrolled students Instructor Add student Drop student	Student Professor

Student	
Name Address Phone number Email address Student number Average mark received Validate identifying info Provide list of seminars taken	Enrollment

Building	
Building Name Rooms Provide name Provide list of available rooms for a given time period	Room

# Рабочее пространство CRC моделирования



# Принципы CRC-моделирования

- ➔ Принцип «Малых приращений»
- ➔ Четкое определение границы системы
- ➔ Перекомпозиция классов до выделения явной функции

**Технология структурного анализа и проектирования (SADT)**

**Проектирование  
Функциональности**

**Стандарты  
IDEF**

**IDEF — методологии семейства ICAM (Integrated Computer-Aided Manufacturing) для решения задач моделирования сложных систем, позволяет отображать и анализировать модели деятельности широкого спектра сложных систем в различных разрезах**

**IDEF1 — Information Modeling — методология моделирования информационных потоков внутри системы, позволяющая отображать и анализировать их структуру и взаимосвязи.**

**IDEF1X (IDEF1 Extended) — Data Modeling — методология моделирования баз данных на основе модели «сущность-связь». Применяется для построения информационной модели, которая представляет структуру информации, необходимой для поддержки функций производственной системы или среды.**

## Семейство стандартов

В настоящий момент к семейству IDEF можно отнести следующие стандарты:

- IDEF0 — Function Modeling — методология функционального моделирования. С помощью наглядного графического языка IDEF0 изучаемая система предстает перед разработчиками и аналитиками в виде набора взаимосвязанных функций (функциональных блоков — в терминах IDEF0). Как правило, моделирование средствами IDEF0 является первым этапом изучения любой системы. Методологию IDEF0 можно считать следующим этапом развития хорошо известного графического языка описания функциональных систем SADT (Structured Analysis and Design Technique);
- IDEF1 — Information Modeling — методология моделирования информационных потоков внутри системы, позволяющая отображать и анализировать их структуру и взаимосвязи;
- IDEF1X (IDEF1 Extended) — Data Modeling — методология построения реляционных структур (баз данных), относится к типу методологий «Сущность-взаимосвязь» (ER — Entity-Relationship) и, как правило, используется для моделирования реляционных баз данных, имеющих отношение к рассматриваемой системе;

- IDEF2 — Simulation Model Design — методология динамического моделирования развития систем. В связи с весьма серьезными сложностями анализа динамических систем от этого стандарта практически отказались, и его развитие приостановилось на самом начальном этапе. В настоящее время присутствуют алгоритмы и их компьютерные реализации, позволяющие превращать набор статических диаграмм IDEF0 в динамические модели, построенные на базе «раскрашенных сетей Петри» (CPN — Color Petri Nets);
- IDEF3 — Process Description Capture — Документирование технологических процессов,  
IDEF3 — методология документирования процессов, происходящих в системе (например, на предприятии), описываются сценарий и последовательность операций для каждого процесса. IDEF3 имеет прямую взаимосвязь с методологией IDEF0 — каждая функция (функциональный блок) может быть представлена в виде отдельного процесса средствами IDEF3;
- IDEF4 — Object-Oriented Design — методология построения объектно-ориентированных систем, позволяют отображать структуру объектов и заложенные принципы их взаимодействия, тем самым позволяя анализировать и оптимизировать сложные объектно-ориентированные системы;

- **IDEF5 — Ontology Description Capture — Стандарт онтологического исследования сложных систем.** С помощью методологии IDEF5 онтология системы может быть описана при помощи определенного словаря терминов и правил, на основании которых могут быть сформированы достоверные утверждения о состоянии рассматриваемой системы в некоторый момент времени. На основе этих утверждений формируются выводы о дальнейшем развитии системы и производится её оптимизация;
- **IDEF6 — Design Rationale Capture — Обоснование проектных действий.** Назначение IDEF6 состоит в облегчении получения «знаний о способе» моделирования, их представления и использования при разработке систем управления предприятиями. Под «знаниями о способе» понимаются причины, обстоятельства, скрытые мотивы, которые обуславливают выбранные методы моделирования. Проще говоря, «знания о способе» интерпретируются как ответ на вопрос: «почему модель получилась такой, какой получилась?» Большинство методов моделирования фокусируются на собственно получаемых моделях, а не на процессе их создания. Метод IDEF6 акцентирует внимание именно на процессе создания модели;

- **IDEF7 — Information System Auditing — Аудит информационных систем.** Этот метод определён как востребованный, однако так и не был полностью разработан;
- **IDEF8 — User Interface Modeling — Метод разработки интерфейсов взаимодействия оператора и системы (пользовательских интерфейсов).** Современные среды разработки пользовательских интерфейсов в большей степени создают внешний вид интерфейса. IDEF8 фокусирует внимание разработчиков интерфейса на программировании желаемого взаимного поведения интерфейса и пользователя на трех уровнях: выполняемой операции (что это за операция); сценарии взаимодействия, определяемом специфической ролью пользователя (по какому сценарию она должна выполняться тем или иным пользователем); и, наконец, на деталях интерфейса (какие элементы управления, предлагает интерфейс для выполнения операции);

- IDEF9 — Scenario-Driven IS Design (Business Constraint Discovery method) — Метод исследования бизнес ограничений был разработан для облегчения обнаружения и анализа ограничений в условиях которых действует предприятие. Обычно, при построении моделей описанию ограничений, оказывающих влияние на протекание процессов на предприятии уделяется недостаточное внимание. Знания об основных ограничениях и характере их влияния, закладываемые в модели, в лучшем случае остаются неполными, несогласованными, распределенными нерационально, но часто их вовсе нет. Это не обязательно приводит к тому, что построенные модели нежизнеспособны, просто их реализация столкнется с непредвиденными трудностями, в результате чего их потенциал будет не реализован. Тем не менее в случаях, когда речь идет именно о совершенствовании структур или адаптации к предсказываемым изменениям, знания о существующих ограничениях имеют критическое значение;
- IDEF10 — Implementation Architecture Modeling — Моделирование архитектуры выполнения. Этот метод определён как востребованный, однако так и не был полностью разработан;

## Семейство стандартов

(окончание)

- IDEF11 — Information Artifact Modeling. Этот метод определён как востребованный, однако так и не был полностью разработан;
- IDEF12 — Organization Modeling — Организационное моделирование. Этот метод определён как востребованный, однако так и не был полностью разработан;
- IDEF13 — Three Schema Mapping Design — Трёхсхемное проектирование преобразования данных. Этот метод определён как востребованный, однако так и не был полностью разработан;
- IDEF14 — Network Design — Метод проектирования компьютерных сетей, основанный на анализе требований, специфических сетевых компонентов, существующих конфигураций сетей. Также он обеспечивает поддержку решений, связанных с рациональным управлением материальными ресурсами, что позволяет достичь существенной экономии.

**Проектирование  
Хранилищ данных**

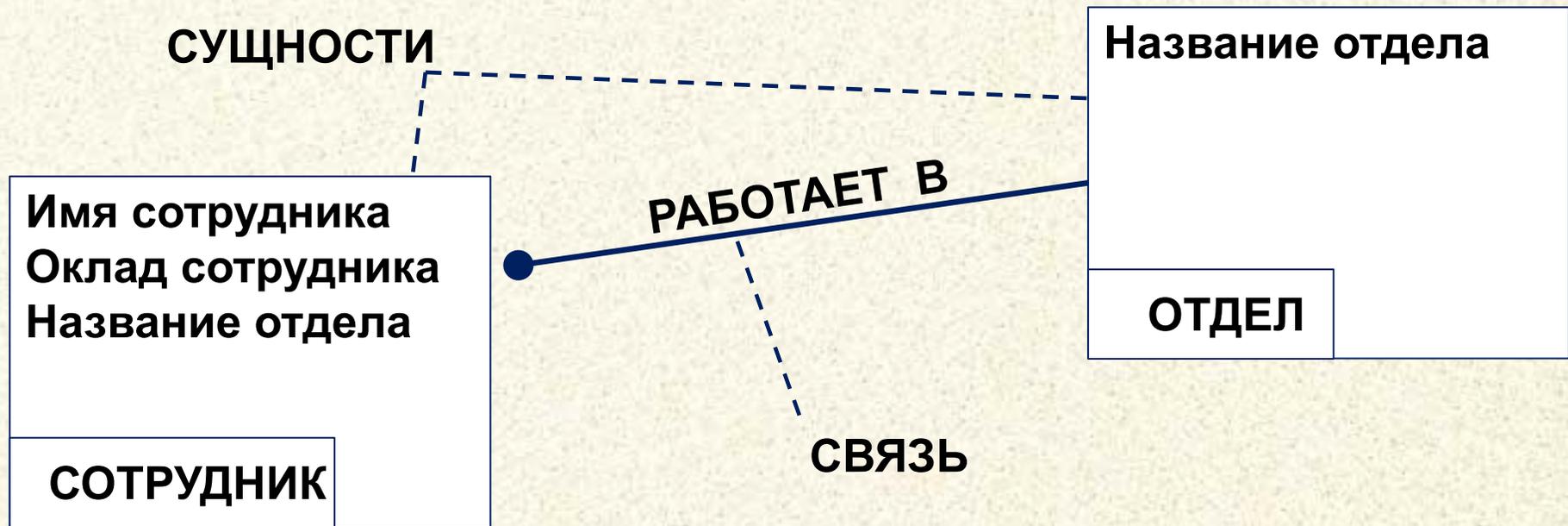
**IDEF1**

**IDEF1x**

**DFD**

# Методика проектирования хранилищ данных

Пример: система учёта персонала (бухгалтерия)



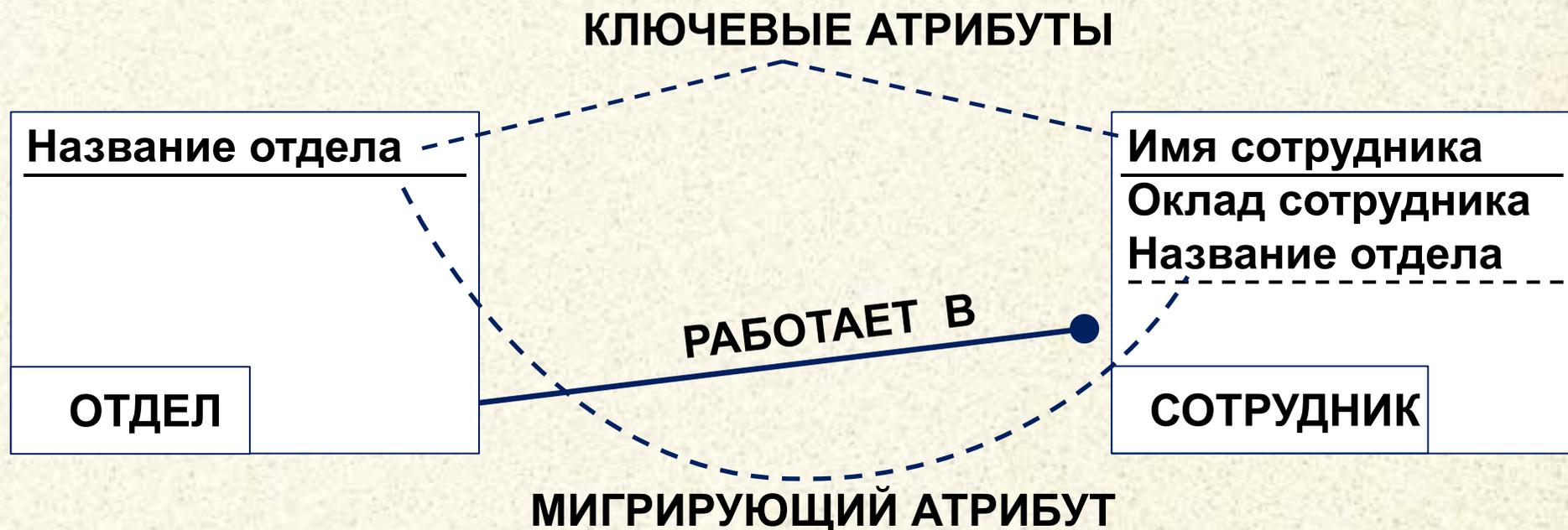
**Сущности** – абстракции (шаблоны) используемых объектов

**Атрибуты** – параметры (характеристики, свойства) объектов

Установка значений атрибутов приводит к созданию

**реализации сущности**, т.е. появлению **отображения реального объекта** в информационной системе

# Использование связи между сущностями

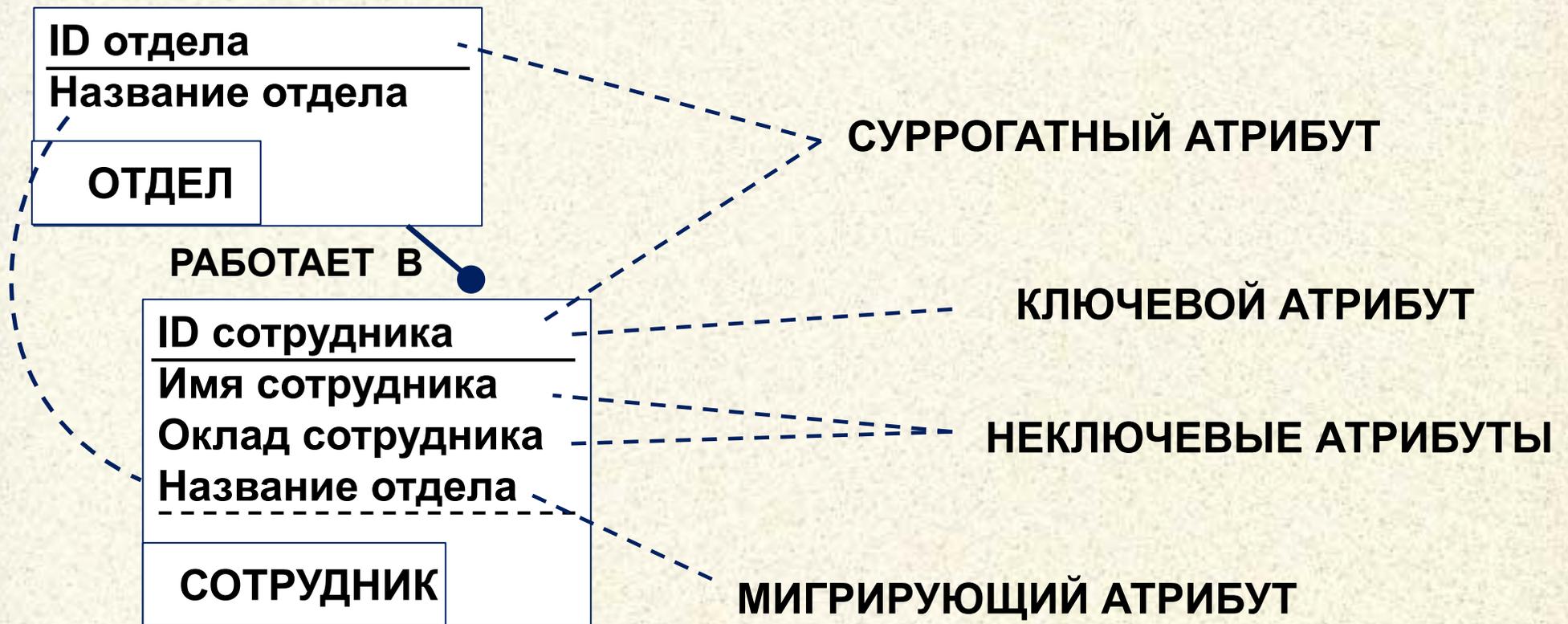


Пример связи между сущностями по схеме **один-ко-многим**

Один экземпляр первой сущности (отдел) связан с несколькими экземплярами второй сущности. Причем первая сущность называется **родительской**, а вторая - **дочерней**

Значение **мигрирующего (внешнего) атрибута** не хранится в записи, а получается через связь сущностей

# Идентификация сущностей



**Суррогатный атрибут (ID)** отсутствует у объекта-прототипа и создается для решения технических задач

**Естественный ключ** образуется самими данными

**Составной ключ** образуется сочетанием нескольких атрибутов

**Суррогатный ключ** – искусственное образование для обеспечения уникальности объекта

# Схема данных



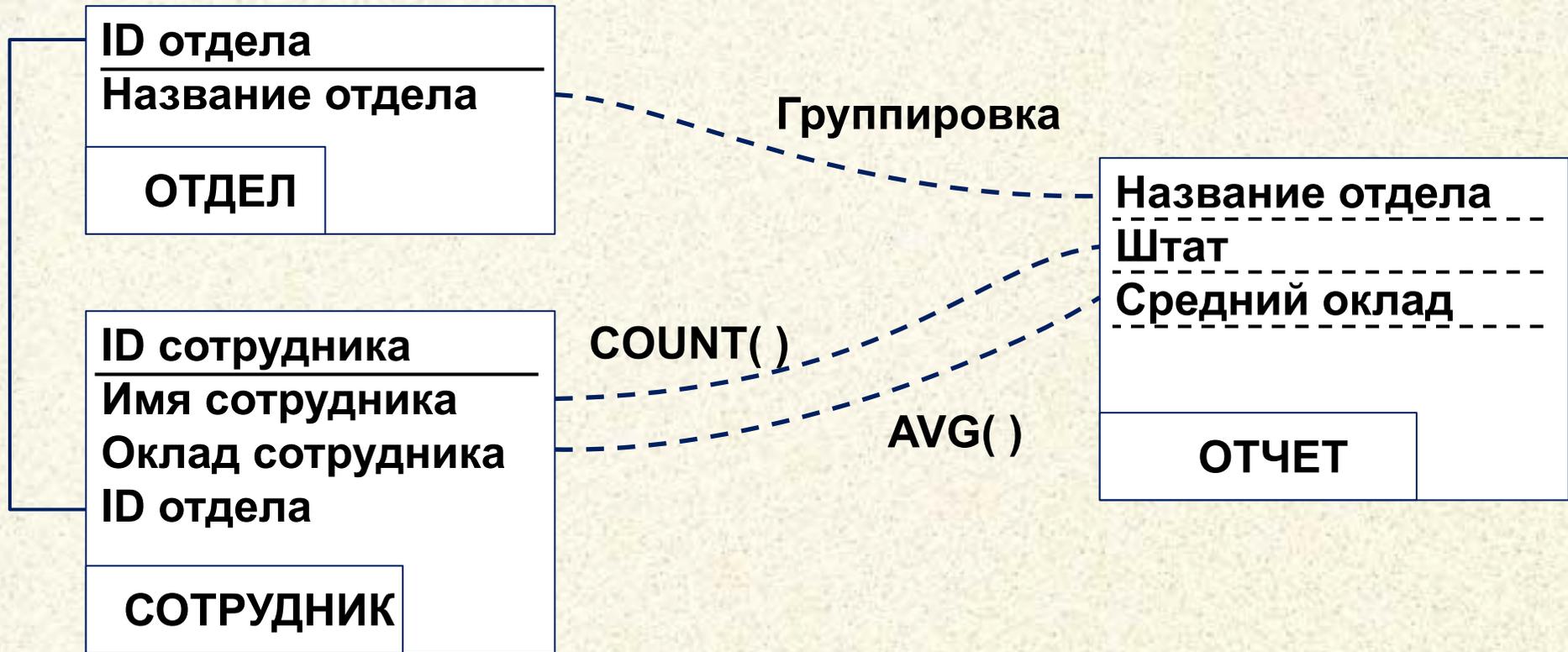
## Отделы

ID отдела	Название отдела
1	Бухгалтерия
2	Адм.-хозяйственный
3	Статистики
4	Прогнозирования

## Сотрудники

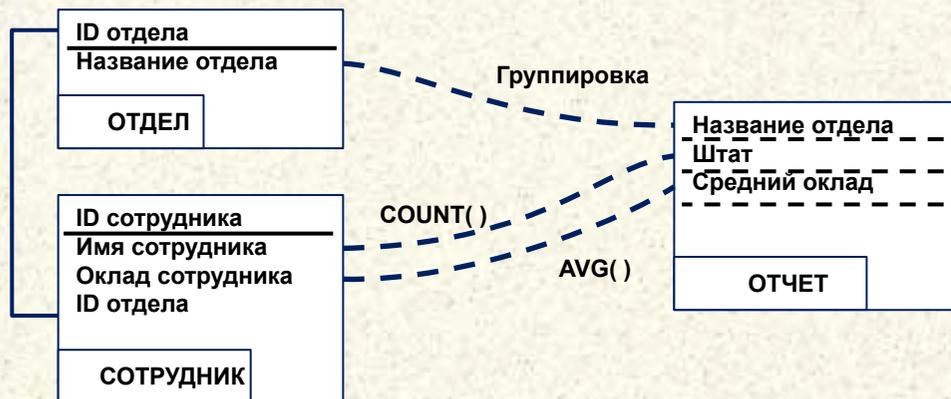
ID сотрудника	Имя сотрудника	ID отдела
1	Иванов И.И.	1
2	Петров П.П.	1
3	Сидоров С.С.	2
4	Кольцов А.С.	3
5	Сергеев Б.А.	2

# Анализ потоков данных



**Диаграммы потоков данных (DFD – Data Flow Diagram)** отражают перемещение данных и способы их обработки

# Пример реализации схемы потоков данных



Вид запроса на выборку данных на языке SQL:

```
SELECT COUNT([Имя сотрудника]),  
       AVG([Оклад сотрудника])  
FROM [ОТДЕЛ], [СОТРУДНИК]  
GROUP BY [Название отдела];
```

Вид запроса на выборку данных в конструкторе запросов Ms Access

Screenshot of Microsoft Access Query Design View showing a 1-to-many relationship between 'ОТДЕЛ' and 'СОТРУДНИК' tables. The 'ОТДЕЛ' table has fields: ID отдела, Название отдела. The 'СОТРУДНИК' table has fields: ID сотрудника, Имя сотрудника, Оклад сотрудника, ID отдела. The relationship is 1 to many (∞).

Поле:	Название отдела	Штат: [СОТРУДНИК].[Имя сотрудника]	Средний оклад: [СОТРУДНИК].[Оклад сотрудника]
Имя таблицы:	ОТДЕЛ		
Групповая операция:	Группировка	Count	Avg
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			
или:			

Результат

Название о	Штат	Средний ок
Деканат	2	35 000,00р.
Кафедра	2	22 500,00р.
Ректорат	1	100 000,00р.

Университет машиностроения

Кафедра «Автоматика и процессы управления»

Блок дисциплин

Информатика и информационные технологии

***Спасибо за внимание !!!***

**Содержание:**

- **Метод CRC-карточек**
- **Проектирование хранилищ данных (IDEF1/1x в SADT)**
- **Технология структурного анализа и проектирования (SADT)**
- **UML**

...

*Контакты:*

[mami.testolog.ru](mailto:mami.testolog.ru)

[timid@mami.ru](mailto:timid@mami.ru)

[inform437@gmail.com](mailto:inform437@gmail.com)