

---

# **Моделирование данных « Модель «сущность-связь**

---

---

# Рассматриваемые вопросы:

- **Элементы модели «сущность-связь»**
  - **Диаграммы «сущность-связь»**
  - **Слабые сущности**
  - **Подтипы сущностей**
  - **Пример ER-диаграммы**
  - **Диаграммы «сущность-связь» а стиле UML**
-

# Элементы модели «сущность-связь»

## Сущность

- Класс сущностей
- Экземпляр сущности

## Атрибуты

- Композитные атрибуты
- Многозначные атрибуты

## Идентификаторы

- Уникальные/неуникальные
- Композитные

## Связи

- Классы связей
- Экземпляры связей
- Рекурсивные связи

# ЭЛЕМЕНТЫ МОДЕЛИ «СУЩНОСТЬ-СВЯЗЬ»

## Сущность

*Сущность (entity) – это некоторый объект, идентифицируемый в рабочей среде пользователя, нечто такое, за чем пользователь хотел бы наблюдать.*

**Обозначение средствами в UML- диаграммах:**

*Сущность обозначается* 

# Элементы модели «сущность-связь»

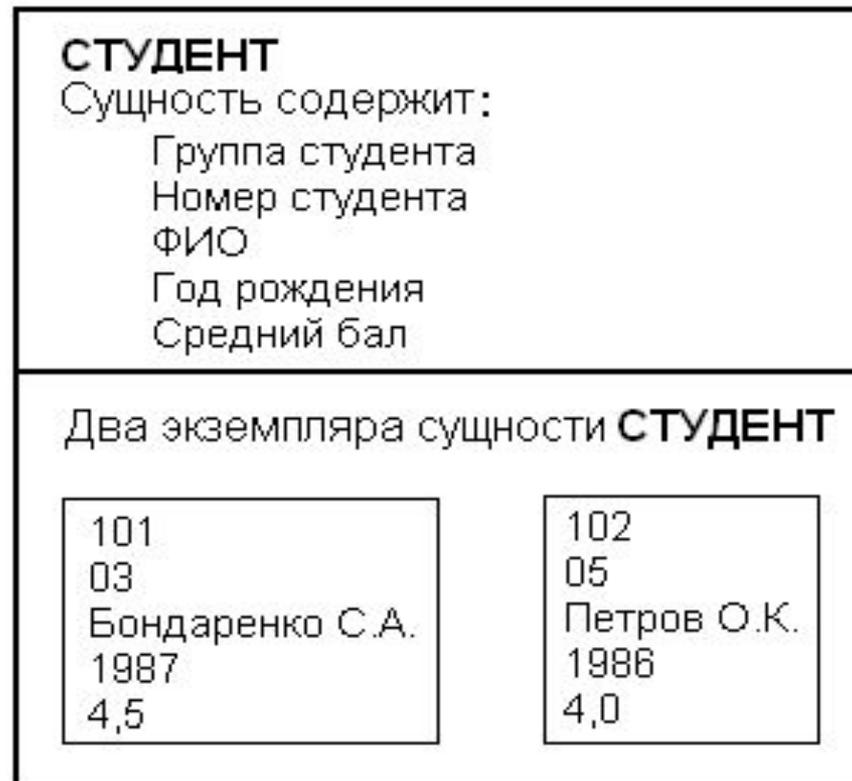
**Класс сущностей** (*entity classes*) – это совокупность сущностей, описывается структурой или форматом сущностей, составляющих этот класс.

**Экземпляр сущности** (*an instance*) представляет конкретную сущность

Обычно класс сущностей держит множество экземпляров сущности.

# Элементы модели «сущность-связь»

## Пример сущности СТУДЕНТ



# Элементы модели «сущность-связь»

## Атрибуты

**Атрибуты (свойства)** – описывают характеристики сущности.

**Пример композитного атрибута:** Адрес, состоящий из группы атрибутов {Улица, Город, Индекс}.

**Пример многозначного атрибута:** атрибут Имя студента сущности ПРЕПОДАВАТЕЛЬ, который может содержать имена нескольких обучаемых им студентов.

# Элементы модели «сущность-связь»

## Идентификаторы

**Идентификаторы** (*identifiers*) – атрибуты, с помощью которых экземпляры сущностей именуются, или идентифицируются.

Если идентификатор является **уникальным**, его значение будет указывать на один и только один экземпляр сущности.

Если идентификатор является **неуникальным**, его значение будет указывать на некоторое множество экземпляров.

Идентификаторы, состоящие из нескольких атрибутов, называются **композиционными идентификаторами** (*composite identifiers*).

# ЭЛЕМЕНТЫ МОДЕЛИ «СУЩНОСТЬ-СВЯЗЬ»

## СВЯЗИ

*Взаимоотношения сущностей выражаются СВЯЗЯМИ.*

**Классы связей** (*relationship classes*) — это взаимоотношения между классами сущностей.

**Экземпляры связи** (*relationship instances*) — взаимоотношения между экземплярами сущностей

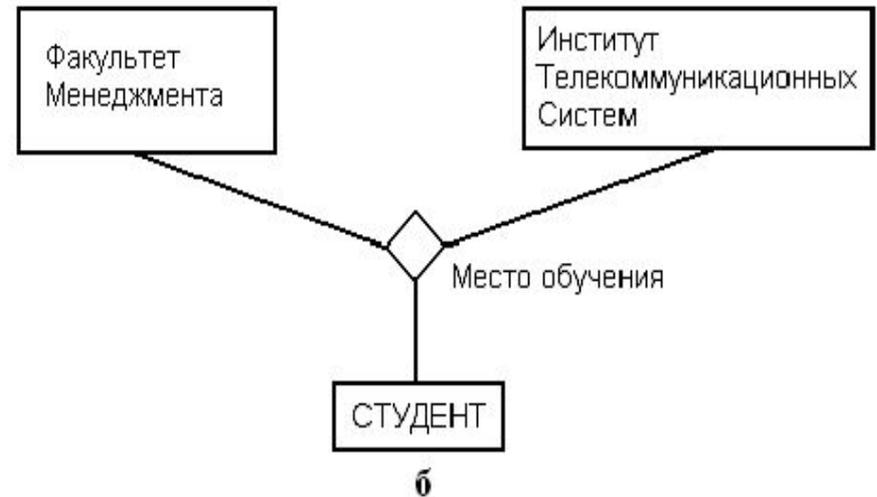
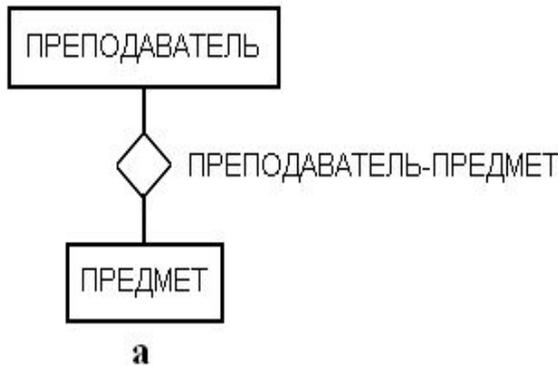
**Степень связи** (*relationship degree*) — число классов сущностей, участвующих в связи.

**Обозначение средствами в UML-диаграммах:**

Связь обозначается 

# Элементы модели «сущность-связь»

Примеры различных степеней связи:  
а – связь степени 2, б – связь степени 3.



Связи степени 2 весьма распространены, их часто называют еще **бинарными связями** (*binary relationships*).

# Элементы модели «СУЩНОСТЬ-СВЯЗЬ»

## Три типа бинарных связей

Обозначение средствами в UML- диаграммах:

Связь  $1:1$  («один к одному») обозначается



Связь  $1:N$  («один к N» или «один ко многим») –



Связь  $N:M$  (читается «N к M» или «многие ко многим») –



Связь обладания в обобщенном виде, когда не указан конкретный тип связи -



Числа внутри ромба, символизирующего связь, обозначают максимальное количество сущностей на каждой стороне связи. Эти *ограничения* называются **максимальными кардинальными числами**, а совокупность из двух таких ограничений для обеих сторон связи называется **максимальной кардинальностью** (*maximum cardinality*) связи.

# ЭЛЕМЕНТЫ МОДЕЛИ «СУЩНОСТЬ-СВЯЗЬ»

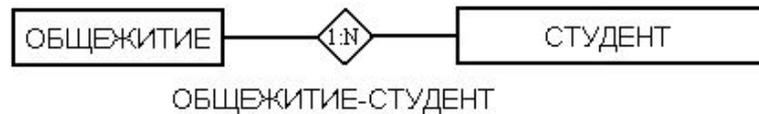
Пример бинарных связей: а – бинарная связь 1:1,

б – бинарная связь 1:N, в – бинарная связь N:M,

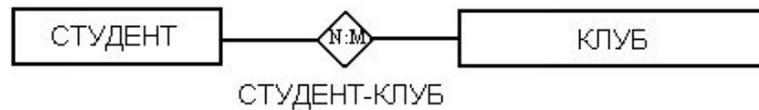
г – представление связи с помощью разветвлений.



**а**



**б**



**в**



**г**

# Диаграммы «сущность-связь»

Схемы бинарных связей, изображенных выше, называются диаграммами «сущность-связь», или **ER-диаграммами** (*entity-relationship diagrams, ER-diagrams*).

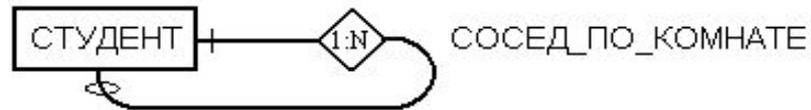
Для указания **минимальной кардинальности** (*minimum cardinality*) существует несколько способов. Один из них, продемонстрирован ниже.

## Связь с указанной минимальной кардинальностью

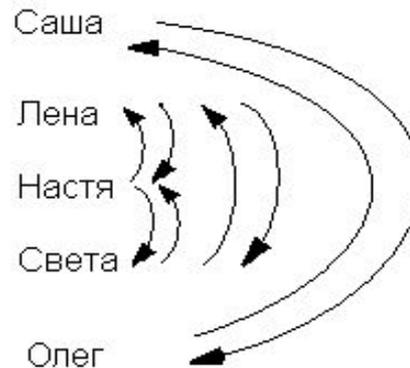


# Диаграммы «СУЩНОСТЬ-СВЯЗЬ»

Связи между сущностями одного и того же класса называются иногда **рекурсивными связями** (*recursive relationships*).



а

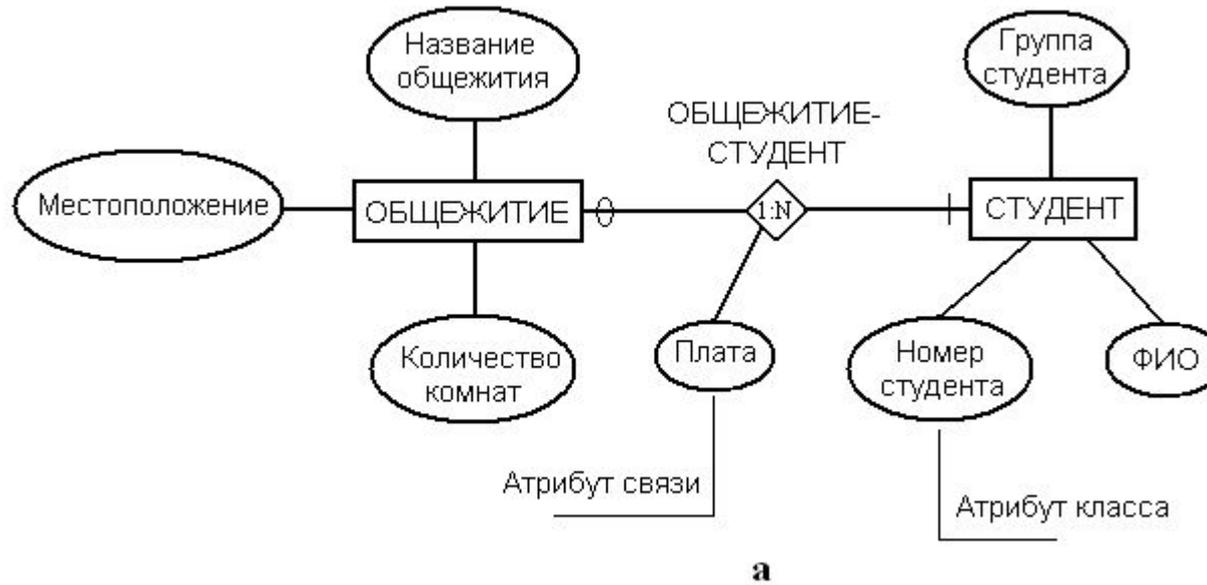


б

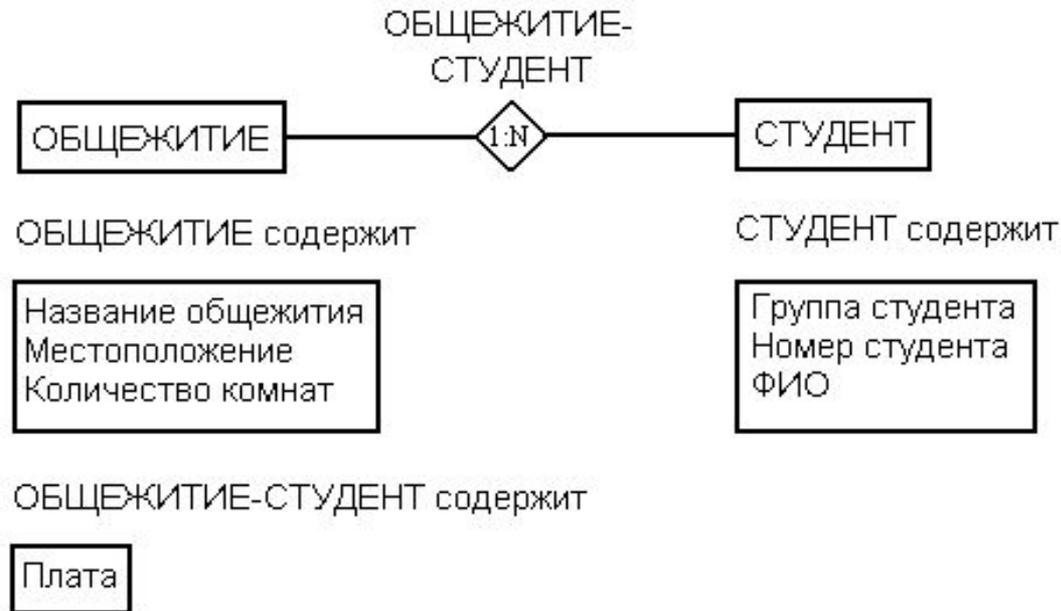
# Диаграммы «сущность-связь»

## Изображение атрибутов в диаграммах «сущность-связь»

В некоторых версиях ER-диаграмм *атрибуты* обозначаются *эллипсами*, соединенными с сущностью или связью, которой они принадлежат.



# Диаграммы «сущность-связь»



б

Изображение свойств на диаграммах «сущность-связь»:

**а** – указание на диаграмме; **б** – отдельное перечисление.

---

# Слабые сущности

**Слабые сущности** (*weak entity*) - сущности, которые могут существовать в базе данных только в том случае, если в ней присутствует сущность некоторого другого типа.

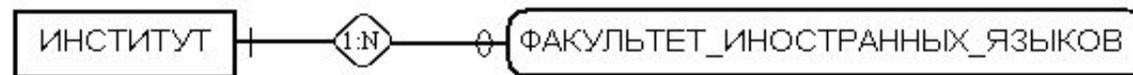
Сущность, не являющаяся слабой, называется **сильной сущностью** (*strong entity*).

**Идентификационно-зависимые сущности** (*ID-dependent entities*) - это такие сущности, идентификаторы которых содержат идентификатор другой сущности.

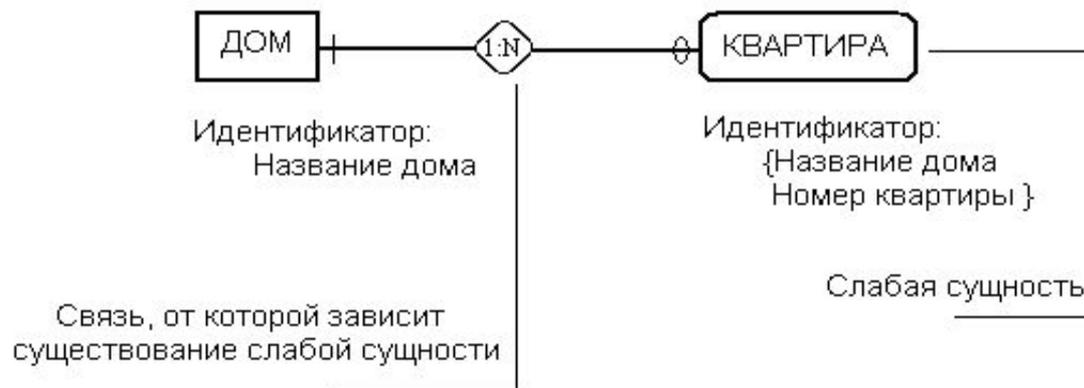
---

# Слабые сущности

**Слабые сущности: а – пример слабой сущности, б – пример идентификационно-зависимой сущности.**



**а**



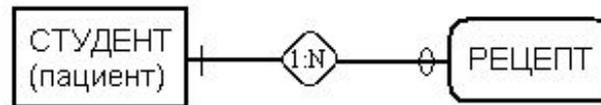
**б**

Слабая сущность

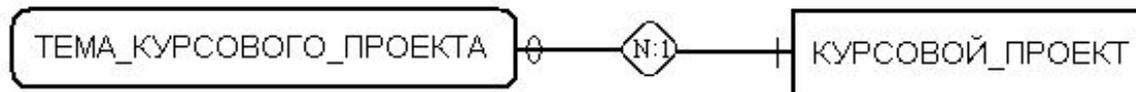
# Слабые сущности

Чтобы *сущность* можно было отнести к разряду **слабых**, она должна *логически зависеть от другой сущности*.

## Пример обязательных сущностей



**а**



Идентификатор:  
{Название курсового проекта,  
Название задачи}

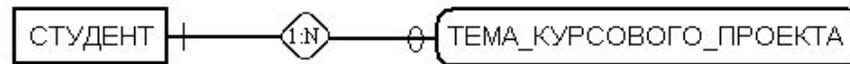
Идентификатор:  
Название курсового проекта

**б**

# Слабые сущности

*Многозначные атрибуты* представляются в модели «сущность-связь» путем создания новой слабой сущности и построения связи вида «один ко многим».

## Представление многозначных атрибутов с помощью слабых сущностей



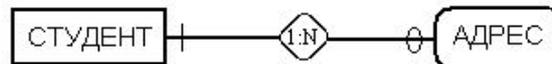
СТУДЕНТ содержит:

Группа студента  
Номер студента  
и прочие атрибуты

ТЕМА\_КУРСОВОГО\_ПРОЕКТА содержит:

Тема курсового проекта

**а**



СТУДЕНТ содержит:

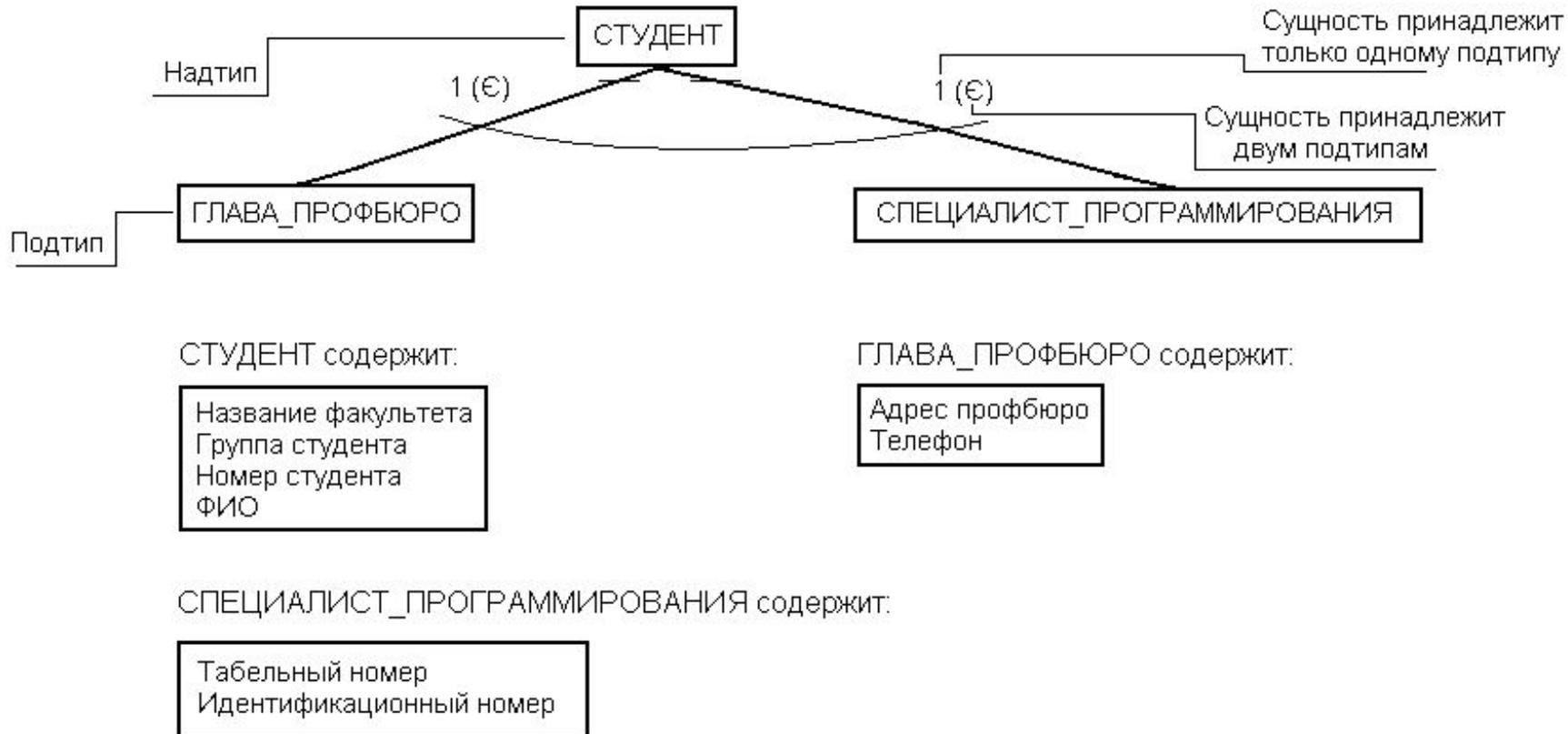
Группа студента  
Номер студента  
и прочие атрибуты

АДРЕС содержит:

Улица  
Город  
Индекс

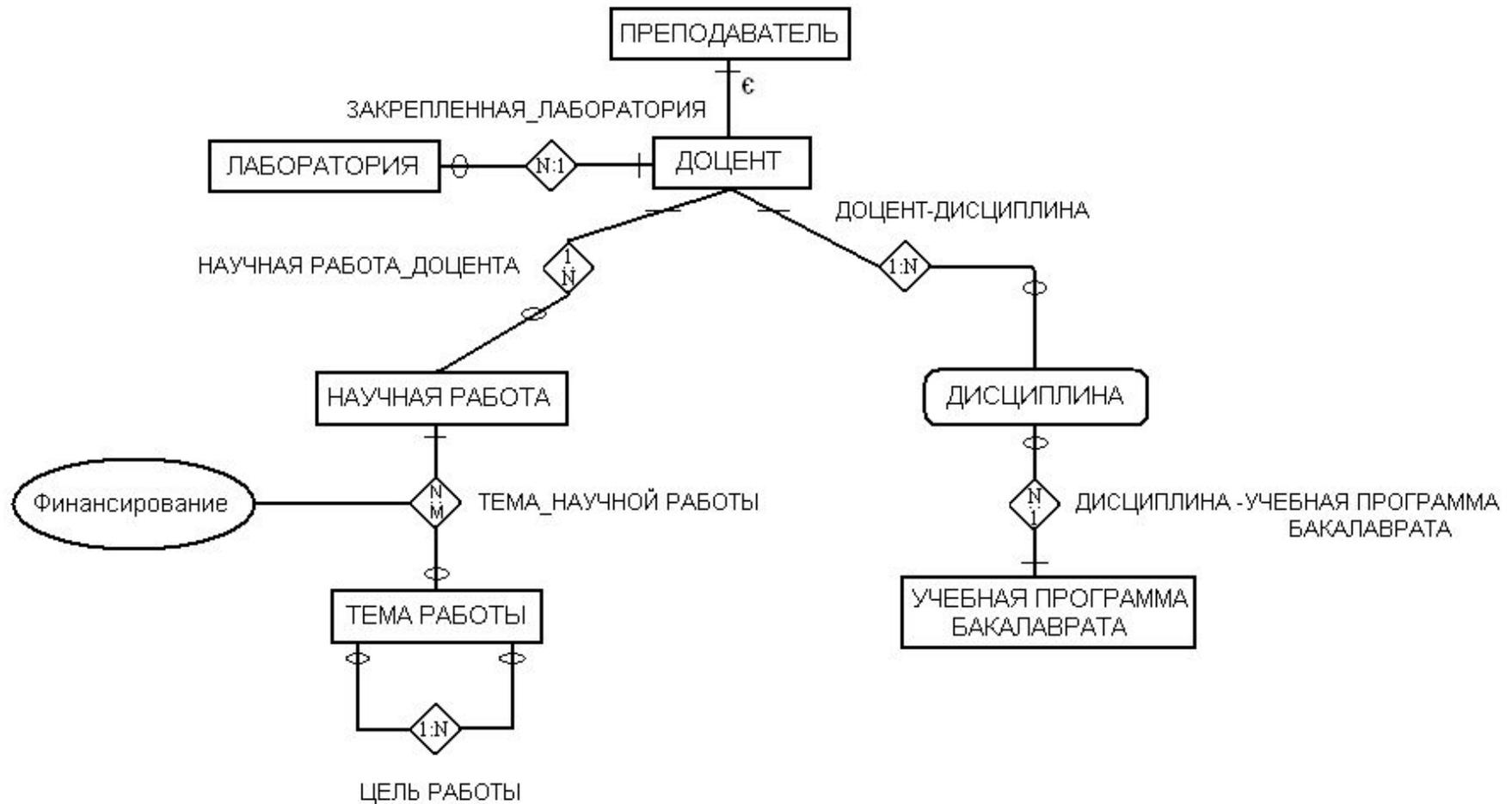
**б**

# Подтипы сущностей



Иерархии генерализации имеют специальную характеристику, называемую **наследованием** (*inheritance*), которая означает, что подтипы классов сущностей наследуют атрибуты от надтипа.

# Пример ER-диаграммы



---

# Диаграммы «сущность-связь» в стиле UML

**Унифицированный язык моделирования**  
*(UML, Unified Model Language) - это набор  
структур и методик для моделирования и  
проектирования объектно-ориентированных  
программ (ООП) и приложений.*

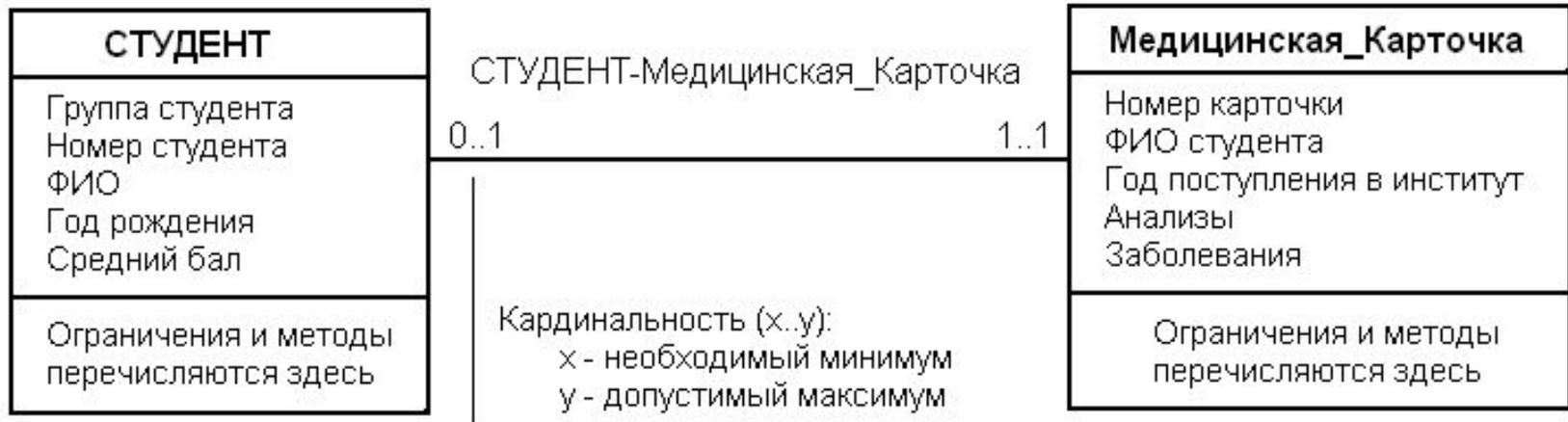
---

# Диаграммы «сущность-связь» в стиле UML

## Сущности и связи в UML

Представления различных типов связей в UML:

**а** – связь 1:1, **б** – связь 1:N, **в** - связь N:M.



**а**

# Диаграммы «сущность-связь» в стиле UML



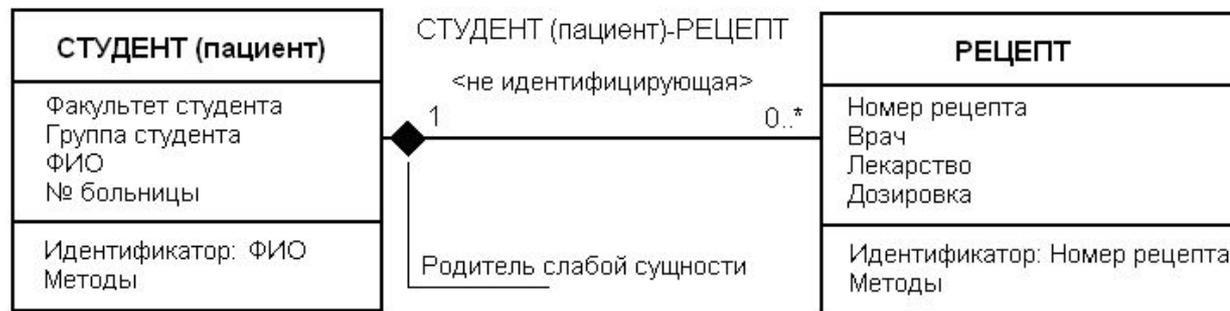
б



в

# Диаграммы «сущность-связь» в стиле UML

## Представление слабых сущностей



а



б

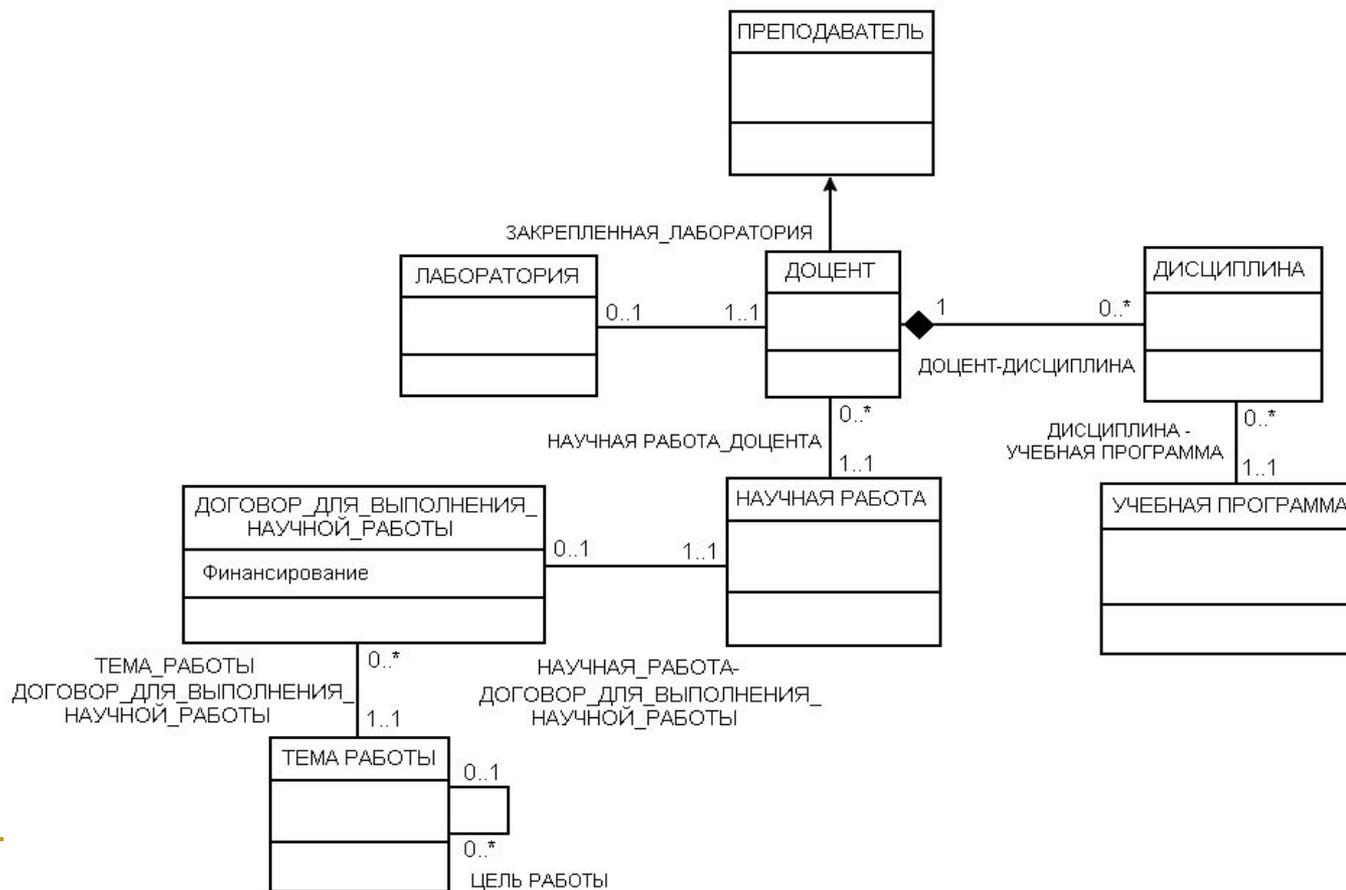
# Диаграммы «сущность-связь» в стиле UML

## Представление подтипов



# Диаграммы «сущность-связь» в стиле UML

## UML-версия диаграммы «сущность-связь»



# Диаграммы «сущность-связь» в стиле UML

## Конструкции ООП, введенные языком UML

- Классы всех сущностей, которые должны храниться в базе данных, помечаются стереотипом «Persistent» (устойчивый)
- UML допускает назначение атрибутов классам сущностей
- UML использует объектно-ориентированную нотацию для обозначения видимости атрибутов и методов
  - «+» - открытые
  - «#» - защищенными
  - «-» - закрытыми

# Диаграммы «сущность-связь» в стиле UML

**Открытым** (*public*) называется такой *атрибут*, который может читаться и изменяться любым методом любого объекта.

Термин **защищенный** (*protected*) означает, что *атрибут* или *метод* доступен только для методов данного класса и его подклассов.

А термин **закрытый** (*private*) указывает на то, что соответствующий *атрибут* или *метод* доступен только для методов данного класса.

- В UML задаются ограничения и методы.

# Диаграммы «сущность-связь» в стиле UML

Представление классов сущностей в UML с помощью  
конструкций ООП

