

# Разработка технического задания к ИС

Управление данными

*Разработка технического задания* – перевод понимания разработчиком предметной области в требования, которым должна удовлетворять создаваемая ИС.

*Техническое задание* – документ, содержащий спецификацию требований к создаваемой ИС.

*Этапы разработки технического задания к ИС:*

1. Формирование требований к ИС.
2. Прототипирование интерфейса пользователя ИС.
3. Разработка концептуальной модели данных.

*Пример:*

ИС должна обеспечивать учет и выдачу информации:

- о пассажирах всех рейсов;
- о расписании рейсов и фактической отправке самолетов;
- о техническом обеспечении и экипажах рейсов.

## 1. Формирование требований к ИС:

1. Определение видения, границ и назначения ИС:
  - Краткое описание ИС.
  - Преимущества, которые получит заказчик при внедрении ИС.
2. Определение функциональных требований к ИС:
  - Функциональная модель ИС.
  - Описание функциональных требований к ИС.
  - Описание данных и бизнес-правил.
3. Определение нефункциональных требований к ИС.
  - Перечень нефункциональных требований к ИС.

## 2. Прототипирование интерфейса пользователя:

*Интерфейс пользователя* (User Interface, UI) – средства, доступные пользователю для работы с данными и управления ИС;

– включает:

- формы,
- диалоговые окна,
- отчеты,
- меню.

*Форма* – окно, предназначенное для представления, ввода и редактирования данных удобным и привычным для пользователя способом и направленное на решение законченной функциональной задачи (варианта использования).

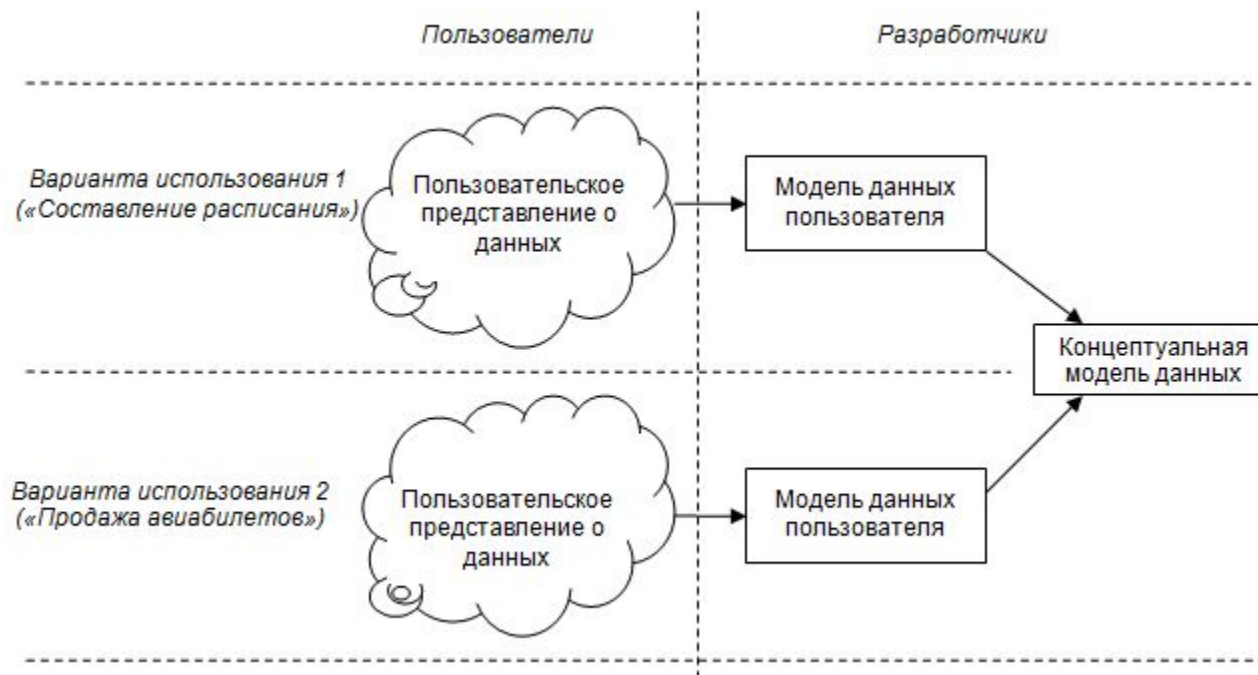
*Элемент управления* – графический объект, размещенный в форме и предназначенный для изображения данных, выполнения операций или просто для красоты.

Инструментальные средства для разработки прототипов UI:

- MS Visio – только рисование интерфейса;
- GUI Machine – не только рисование интерфейса, но и возможность «покликать».

## 3. Разработка концептуальной модели данных:

**Концептуальная модель данных** – концептуальное представление хранимой информации, используемой в проектируемой ИС;  
– первоначальный проект БД.



Задача разработчика – построить модель данных пользователя, определяющую объекты, информация о которых должна храниться в БД, их структуру и связи между ними.

Концептуальная модель данных должна поддерживать все пользовательские представления о данных, т.к. БД – это единое хранилище информации для всей ИС.

### 3. Разработка концептуальной модели данных

**Модель «сущность – связь», ER-диаграмма, ERD (Entity Relationship Diagram)** – средство моделирования данных.

Ключевые элементы ER-диаграммы:

- сущности;
- атрибуты;
- идентификаторы;
- связи.

1). *Сущность* (Entity) – реальный или абстрактный объект, являющийся узловой точкой сбора информации.

*Класс сущностей* – совокупность сущностей одного типа.

*Экземпляр сущности* – конкретная сущность.

Пример: Класс сущностей КЛИЕНТ (НомерКлиента, ИмяКлиента, АдресКлиента, ТелефонКлиента)  
Экземпляр сущности КЛИЕНТ – (123, Иванов, СПб .....

2). *Атрибут* (Attribute) – поименованная характеристика сущности.

*Композитный атрибут* – атрибут, состоящий из группы атрибутов.

*Многозначный атрибут* – атрибут, содержащий несколько значений.

Пример: Композитный атрибут АдресКлиента – {Индекс, Город, Улица, Дом}

Многозначный атрибут ДоверенноеЛицо – {Иванов, Петров, Сидоров}

Композитный и многозначный атрибут ТелефонКлиента – {КодГорода, НомерТелефона}

### 3. Разработка концептуальной модели данных

3). *Идентификатор* (Identifier) – атрибут или совокупность атрибутов, однозначно определяющие каждый экземпляр сущности.

Пример: Паспортные данные, ИНН, Табельный Номер.

*Графическое представление ER-диаграмм:*

- сущности – прямоугольники,
- атрибуты – эллипсы,
- идентификатор – подчеркнутый атрибут.



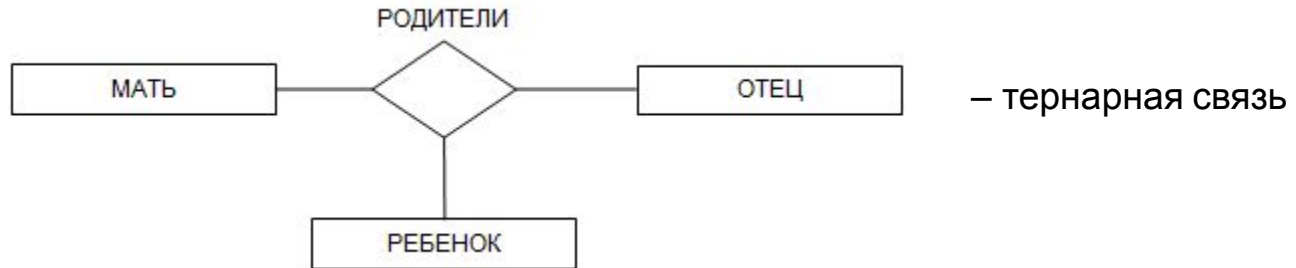
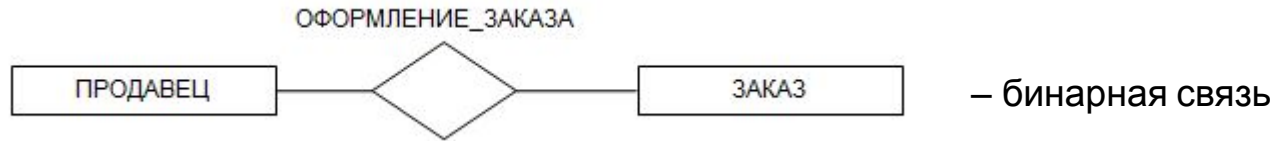
### 3. Разработка концептуальной модели данных

3). *Связь* (Relationship) – моделирует отношение между сущностями.

Свойства связи:

а). Степень связи – число классов сущностей, участвующих в связи.

Пример:







### 3. Разработка концептуальной модели данных

в). Максимальная кардинальность связи – максимальное количество экземпляров сущностей, участвующих в связи.

Максимальные кардинальные числа могут иметь значения, отличные от 1:1, 1:M, M:N.

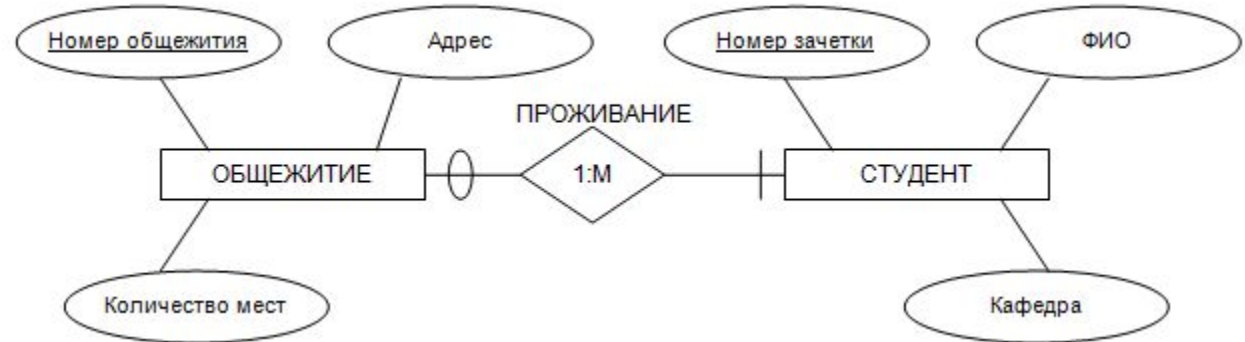
Пример:



г). Минимальная кардинальность связи – минимальное количество экземпляров сущностей, участвующих в связи.

Минимальная кардинальность связи рассматривается как характеристика обязательности или необязательности участия экземпляра сущности в связи.

Пример:



### 3. Разработка концептуальной модели данных

#### Элементы расширенной модели «сущность – связь»

*Подтипы сущности* – сущности, уточняющие варианты типов исходной сущности, являющейся надтипом,  
– наследуют атрибуты от надтипов.

*Взаимоисключаю*

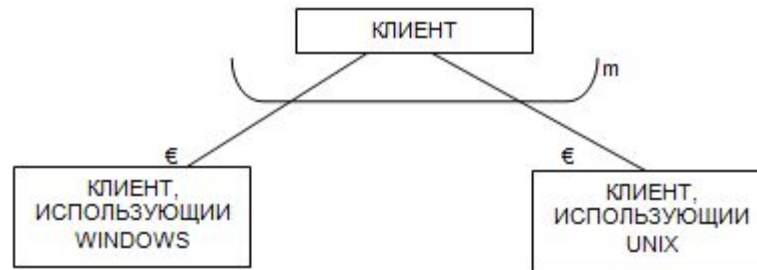
одному подтипу.

Пример:



*Невзаимоисключающие подтипы* – экземпляр надтипа может принадлежать к нескольким подтипам.

Пример:



### 3. Разработка концептуальной модели данных

#### Диаграмма классов UML

*Диаграмма классов* – отражает взаимосвязи между сущностями предметной области, описывает их внутреннюю структуру и типы отношений,  
– совокупность классов с атрибутами и операциями, а также связывающие их отношения.

*Класс (Class)*– абстрактное описание множества однородных объектов.

Графическое изображение  
класса:



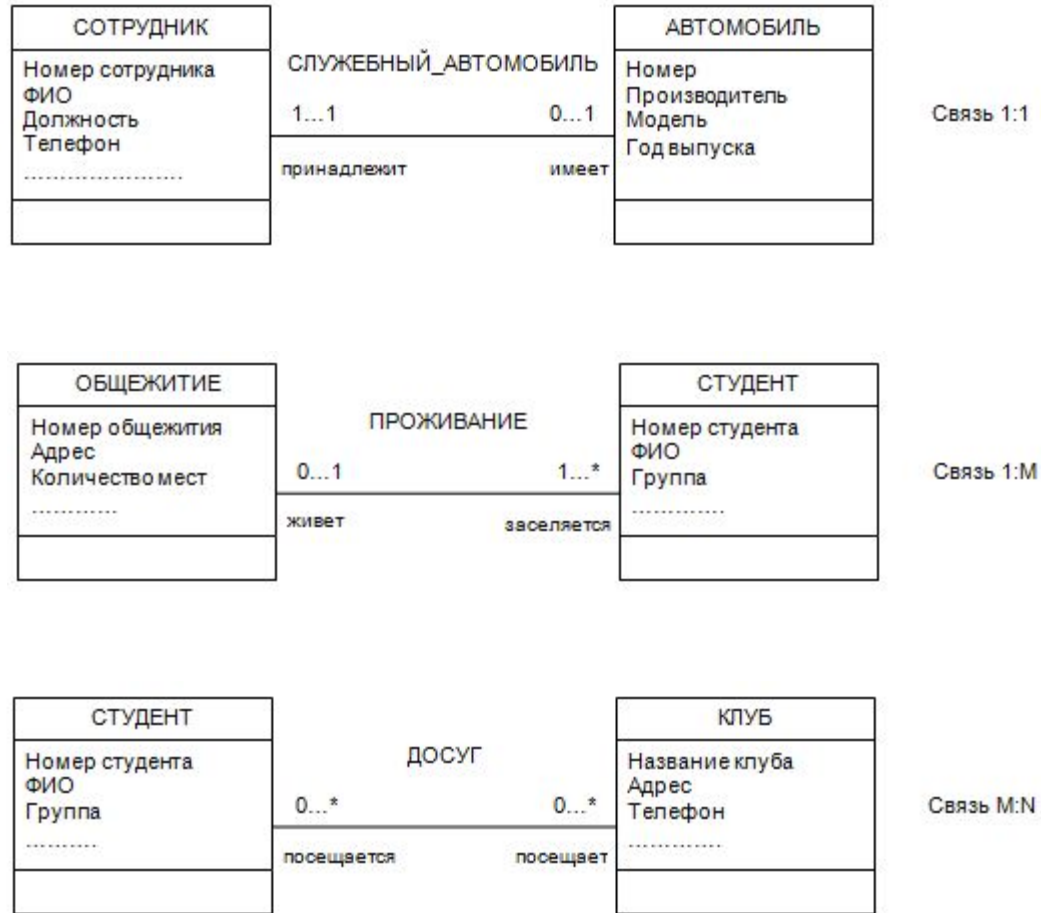
Виды отношений:

1. *Отношение ассоциации* – произвольное отношение между классами.

*Кратность ассоциации* – кардинальность, представленная в формате X..Y,  
где X – необходимый минимум,  
Y – допустимый максимум,  
X и Y – целые числа от 0 до \* (неограниченное значение).

### 3. Разработка концептуальной модели данных

Пример:



### 3. Разработка концептуальной модели данных

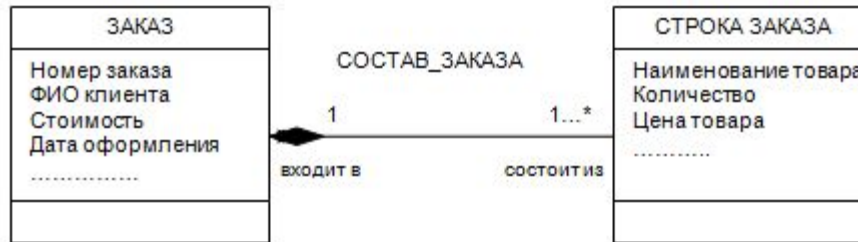
2. *Отношение обобщения* – описывает иерархическое строение классов и наследование их свойств и поведения.

Пример:



3. *Отношение композиции* – для представления связи «часть – целое».

Пример:



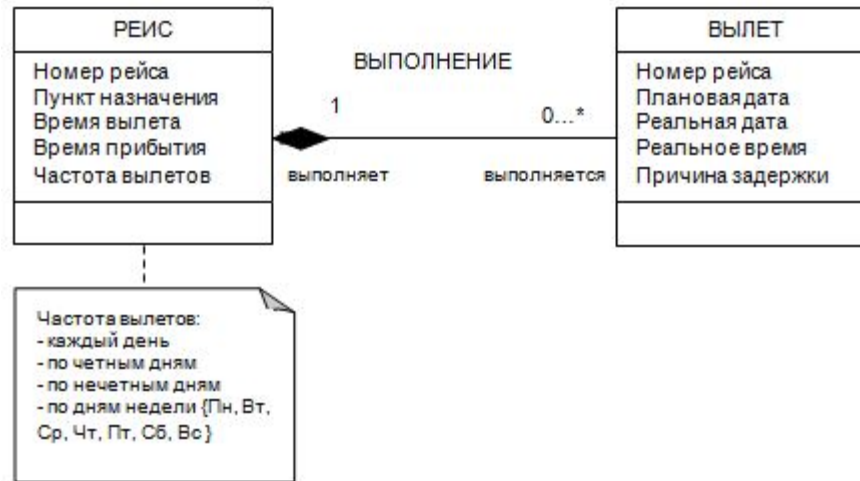
### 3. Разработка концептуальной модели данных

Пример: Модель данных варианта использования «Составление расписания»

Модель «сущность – связь» (ER-диаграмма):



Диаграмма классов UML:



### 3. Разработка концептуальной модели данных

---

Концептуальная модель данных как концептуальное представление хранимой информации обсуждается с заказчиком и с будущими пользователями с целью единого понимания будущей структуры данных в информационной системе.

Последовательность действий:

1. Составить модели данных 2-х видов отдельно для каждой категории пользователей ИС :

- Модель «сущность – связь» (ER-диаграмма).
- Диаграмма классов UML.

2. Объединить созданные модели данных для каждой категории пользователей ИС в единую концептуальную модель данных и представить в 2-х видах:

- Модель «сущность – связь» (ER-диаграмма).
- Диаграмма классов UML.