

Моделирование структурных отношений на UML

1. Диаграмма классов
2. Структурные отношения между классами
3. Отношения ассоциации
4. Отношения агрегации/композиции
5. Отношения зависимости
6. Отношения обобщения (наследования)
7. Применение диаграмм классов

Преподаватель:

Ботов Дмитрий Сергеевич

Предметная область: «Цветы»

- Цветок
- Ромашка
- Роза
- Красная и желтая роза
- Лепесток
- Божья коровка

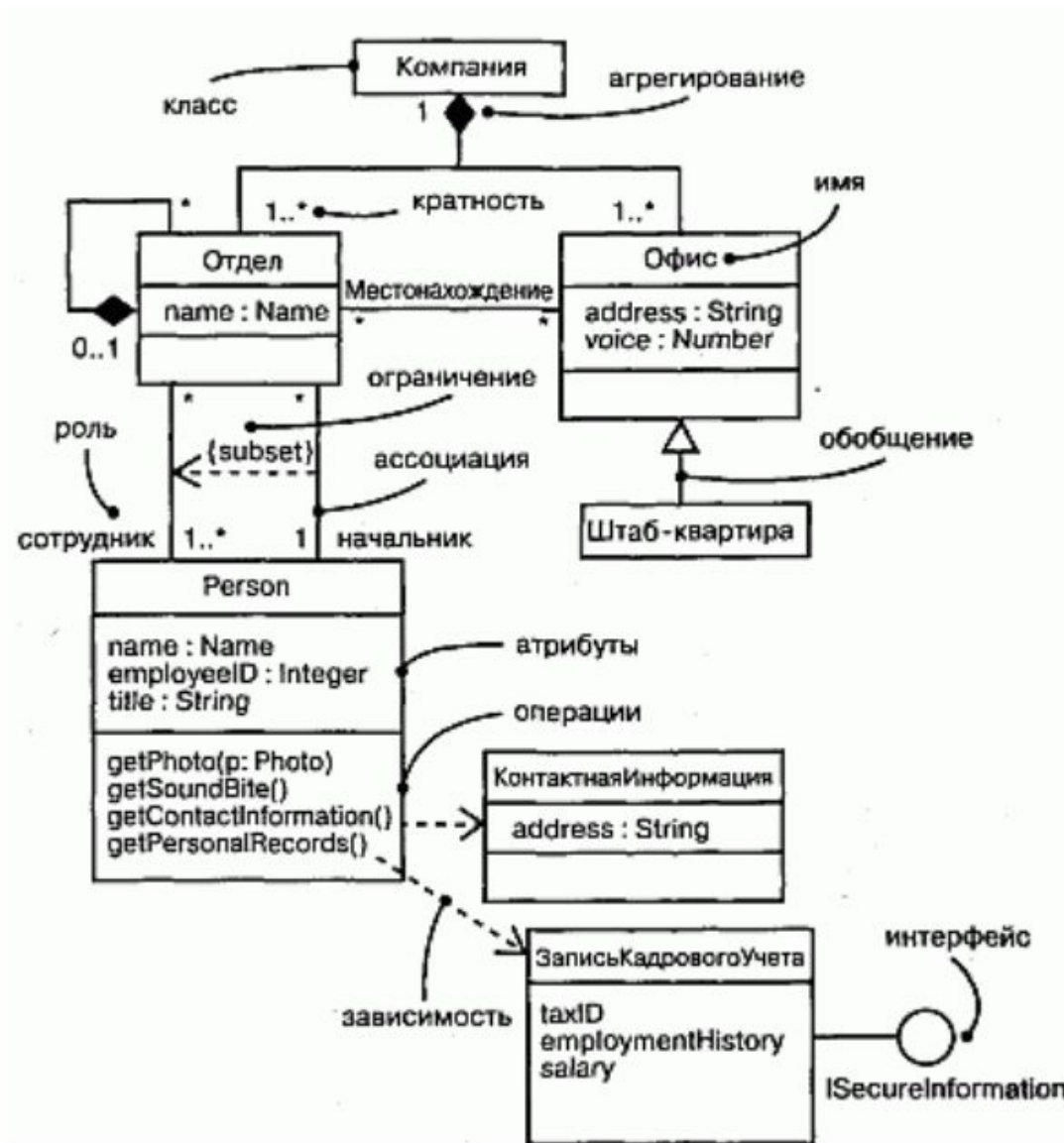


Диаграммы классов в UML

- Диаграммой классов (Class diagram) в UML называют диаграмму, на которой показано множество классов, интерфейсов и структурных отношений между ними.
- **Диаграммы классов** используются для моделирования **статического вида системы** с точки зрения проектирования.

Основные элементы диаграммы классов

- Классы
- Интерфейсы
- Отношения



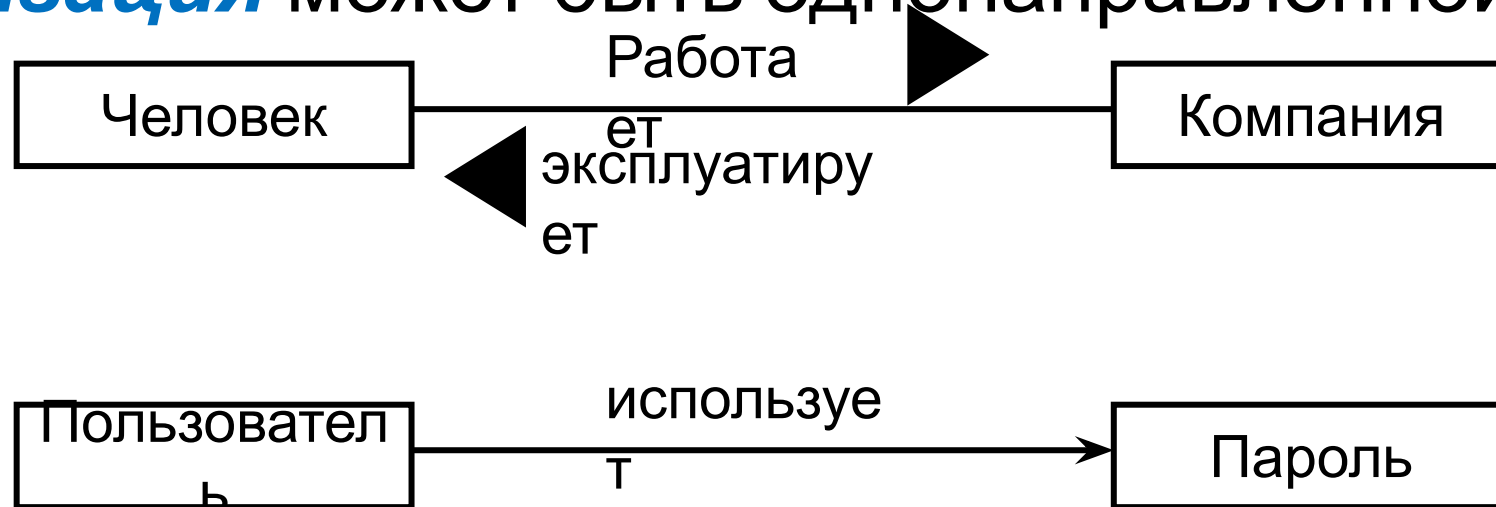
Отношения между объектами

Отношения между объектами – общий термин, охватывающий специфику логических связей между объектами различных классов



Ассоциация

- Отношение между классами и их объектами, которые имеют **равноправное** значение в предметной области
- Позволяет перемещаться от объектов одного класса к объектам другого
- **Навигация** может быть однонаправленной



Ассоциация

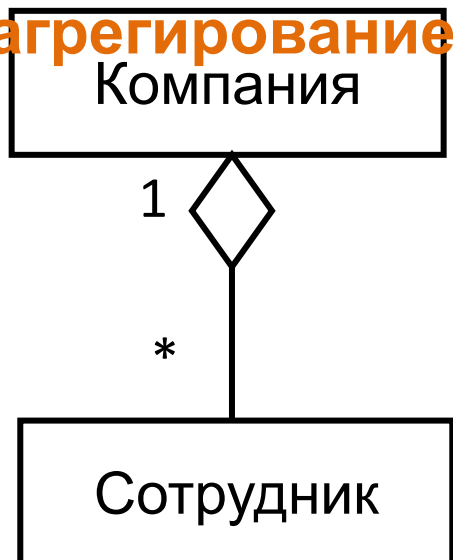
- Каждый класс, который участвует в ассоциации, играет в ней определенную **роль**
- Ассоциацией может быть связано некоторое множество объектов. Число объектов, связанных через одну ассоциацию называют **кратностью**



Ассоциация с агрегированием

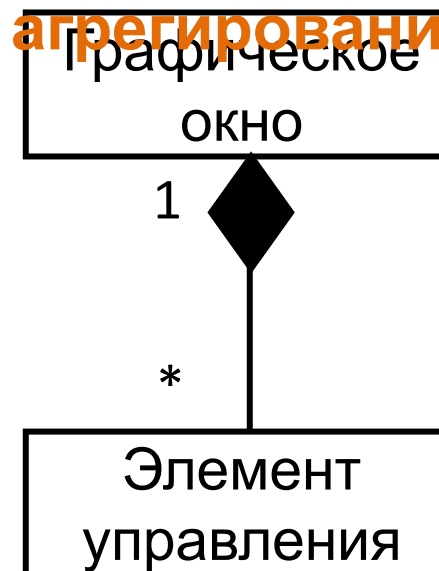
Показывает отношение между объектами типа «**часть-целое**» («**is a part of**»)

Агрегация
(Простое агрегирование)



подчиненные объекты могут существовать независимо от главного объекта

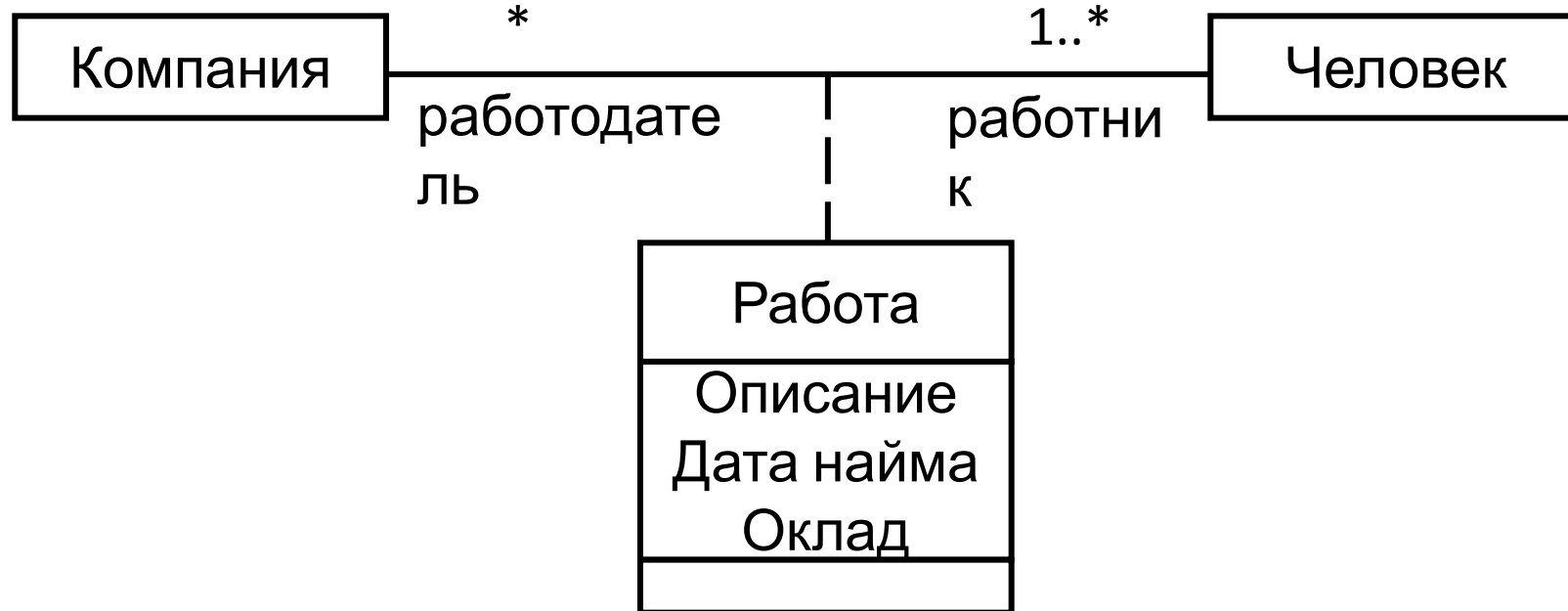
Композиция
(Композитное агрегирование)



время жизни подчиненных объектов зависит от времени жизни главных объектов

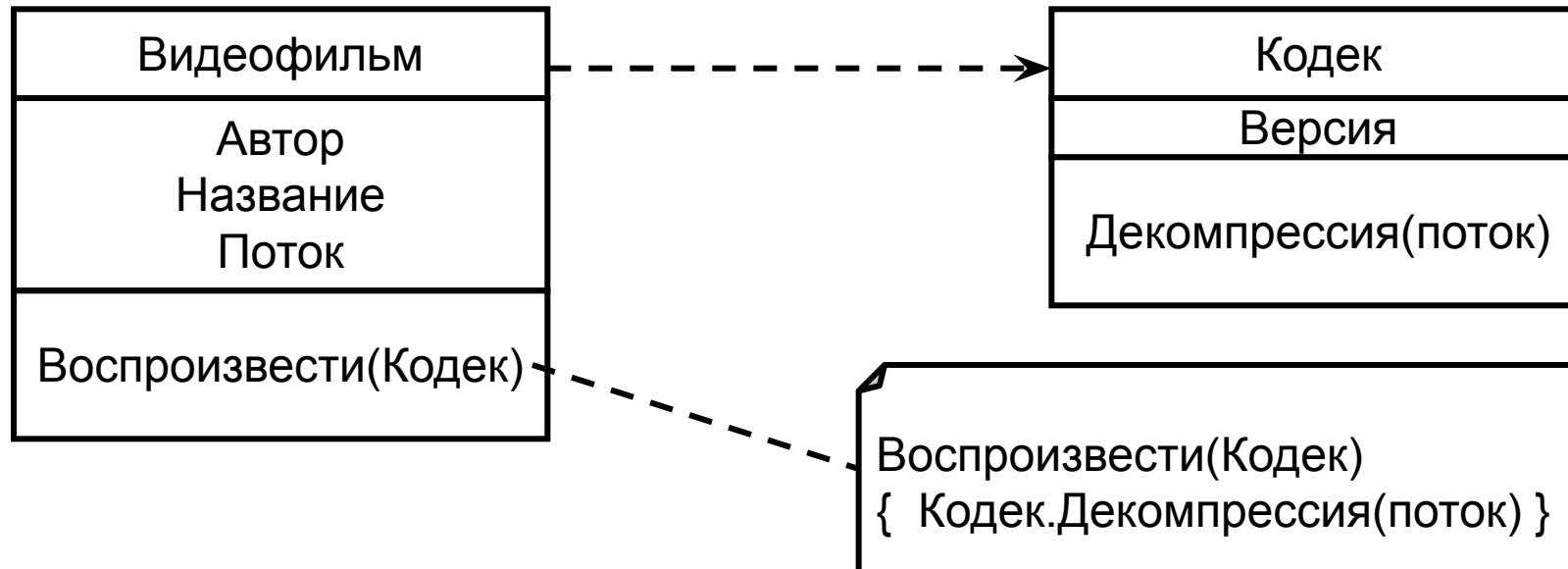
Класс-ассоциация

- Сама ассоциация может быть объектом некоторого класса
- Может быть преобразована к некоторой системе с тремя классами



Зависимость

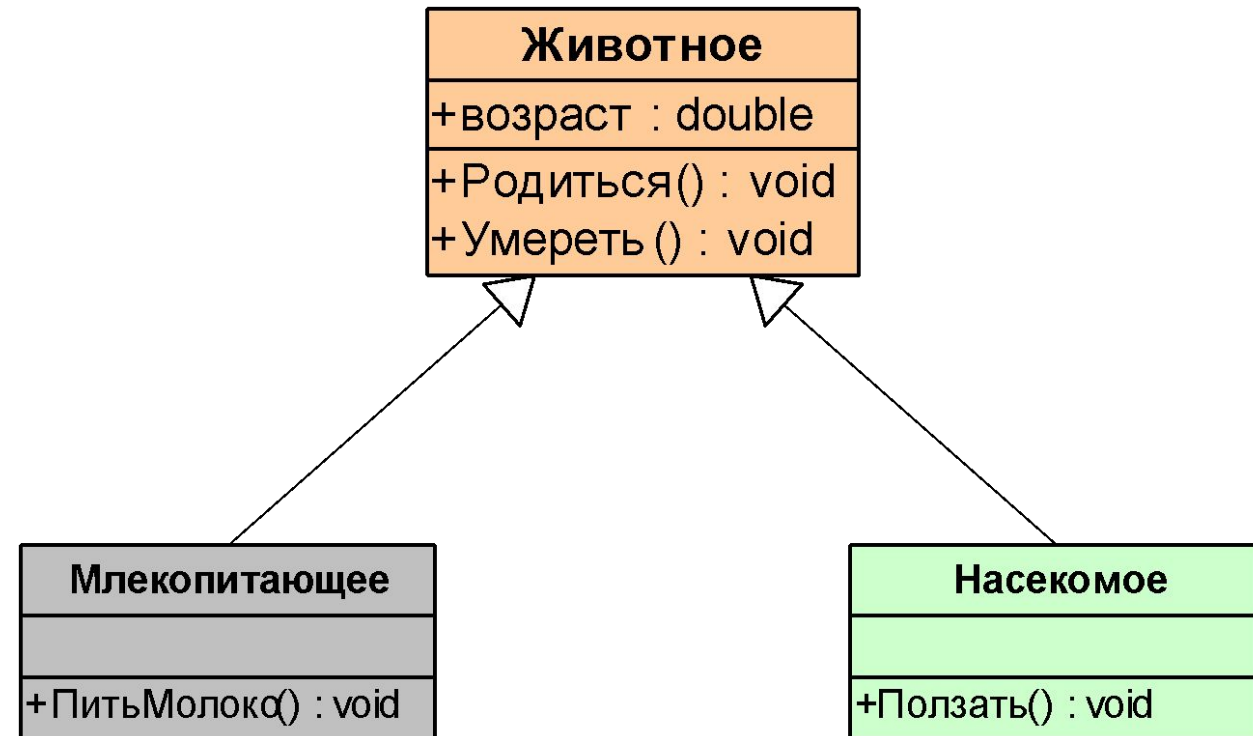
- **Зависимость объектов** возникает, когда реализация класса одного объекта **зависит от спецификаций** операций класса другого объекта
- Чаще всего зависимость проявляется, когда объекты одного класса **используются в качестве параметров методов** другого класса



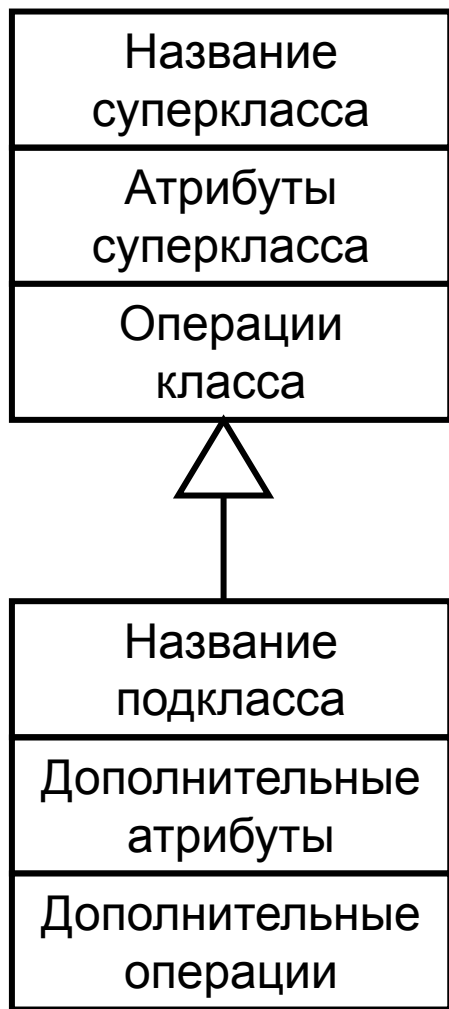
Обобщение / Наследование

Обобщение (Generalization) или **наследование** (Inheritance) – отношение в котором один класс **повторяет структуру и поведение** одного или нескольких классов

Моделирует связь типа «**является**» («**is a**») или «**частное/общее**»



Суперкласс и подкласс



Суперкласс (superclass)

(общая сущность) — класс, структура и поведение которого наследуются и который служит основой для создания новых классов

Подкласс (subclass) (конкретное воплощение общей сущности) — класс, который создается на основе суперкласса

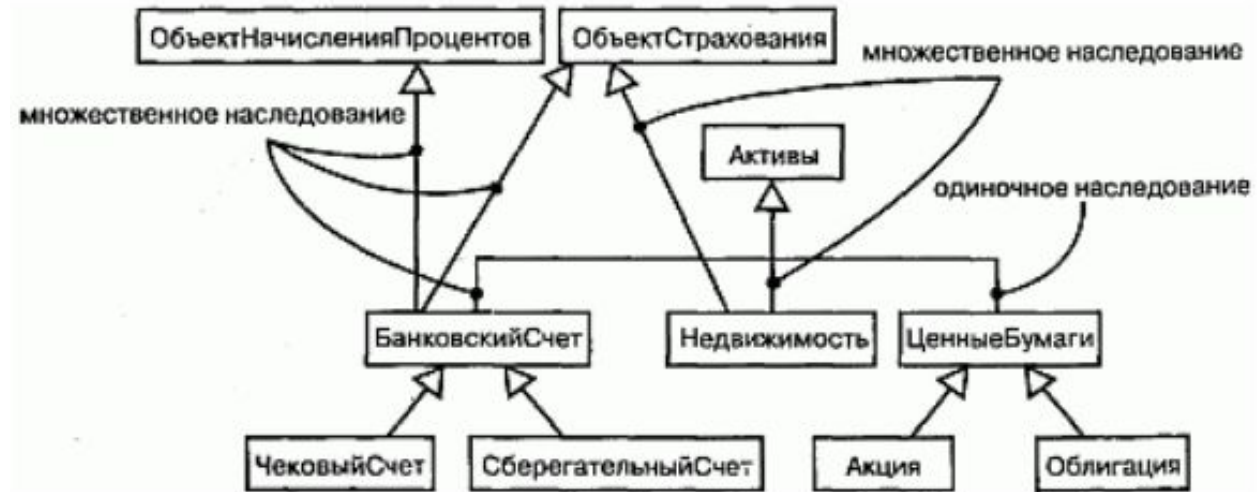
Подкласс включает в себя:

- Все состояние и поведение родительского класса
- Собственное состояние и поведение (специализацию)

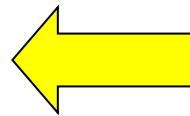
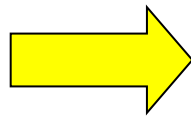
Виды наследования

Наследование может быть:

- **Одиночным:** подкласс повторяет структуру и поведение одного суперкласса
- **Множественным:** подкласс повторяет структуру и поведение нескольких



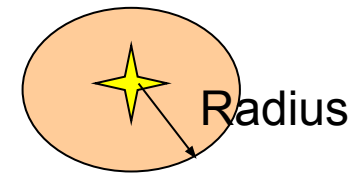
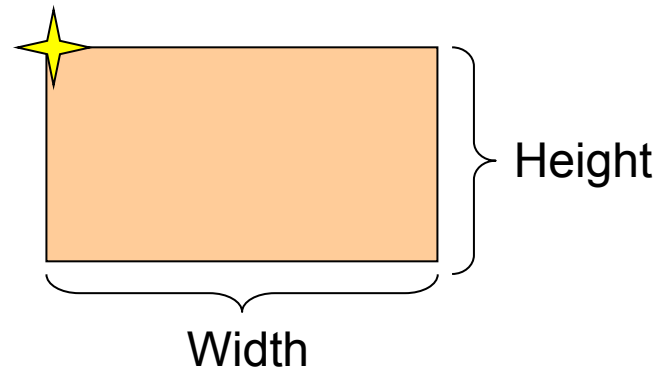
ССОВ



Моделирование обобщения (Г. Буч)

1. Найдите **общие атрибуты, операции и обязанности**, для двух или более классов из данной совокупности
2. Вынесите эти элементы в некоторый общий **суперкласс** (если нужно создайте новый класс)
3. Отметьте в модели, что **подклассы** наследуются от суперкласса, установив между ними отношения обобщения

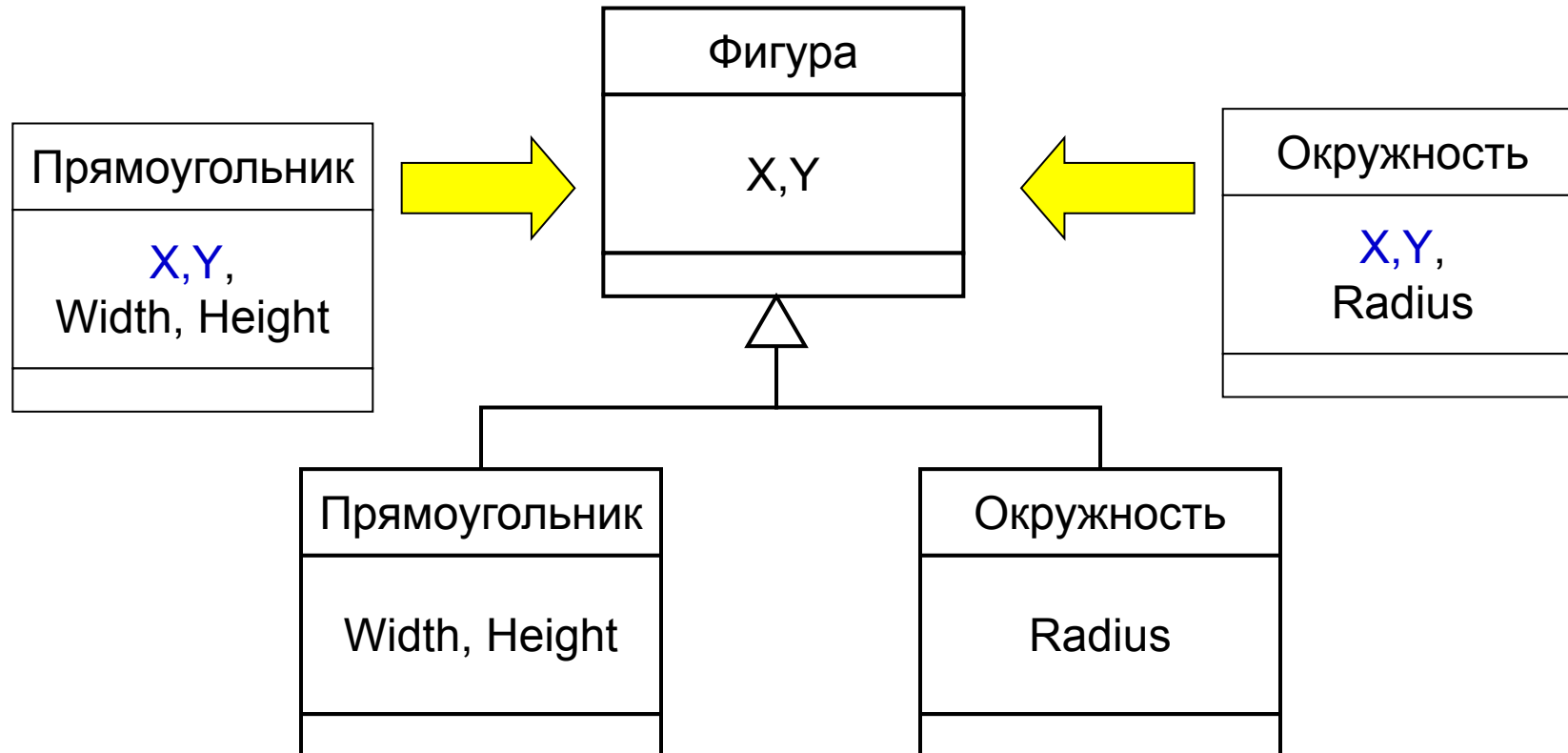
Пример моделирования обобщения



Прямоугольник
X, Y, Width, Height

Окружность
X, Y, Radius

Пример моделирования обобщения



Применение диаграмм классов

Диаграммы классов чаще всего применяются для моделирования:

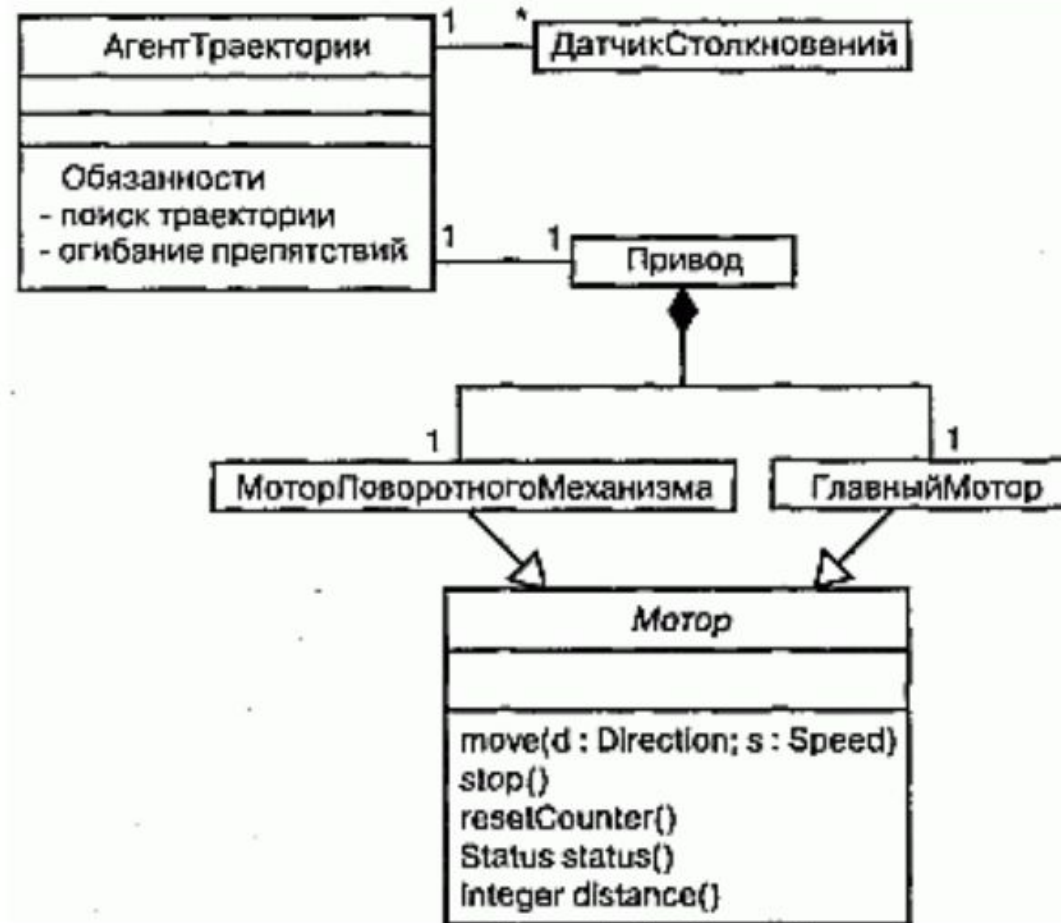
- **словаря предметной области**
- **структуры отдельных модулей (подсистем), фрагментов совместного взаимодействия (коопераций) классов**
- **логической схемы базы данных**
- **классов при прямом или обратном проектировании**

Применение диаграмм классов (1)

- Моделирование структуры отдельных модулей (подсистем), фрагментов совместного взаимодействия (*коопераций*)

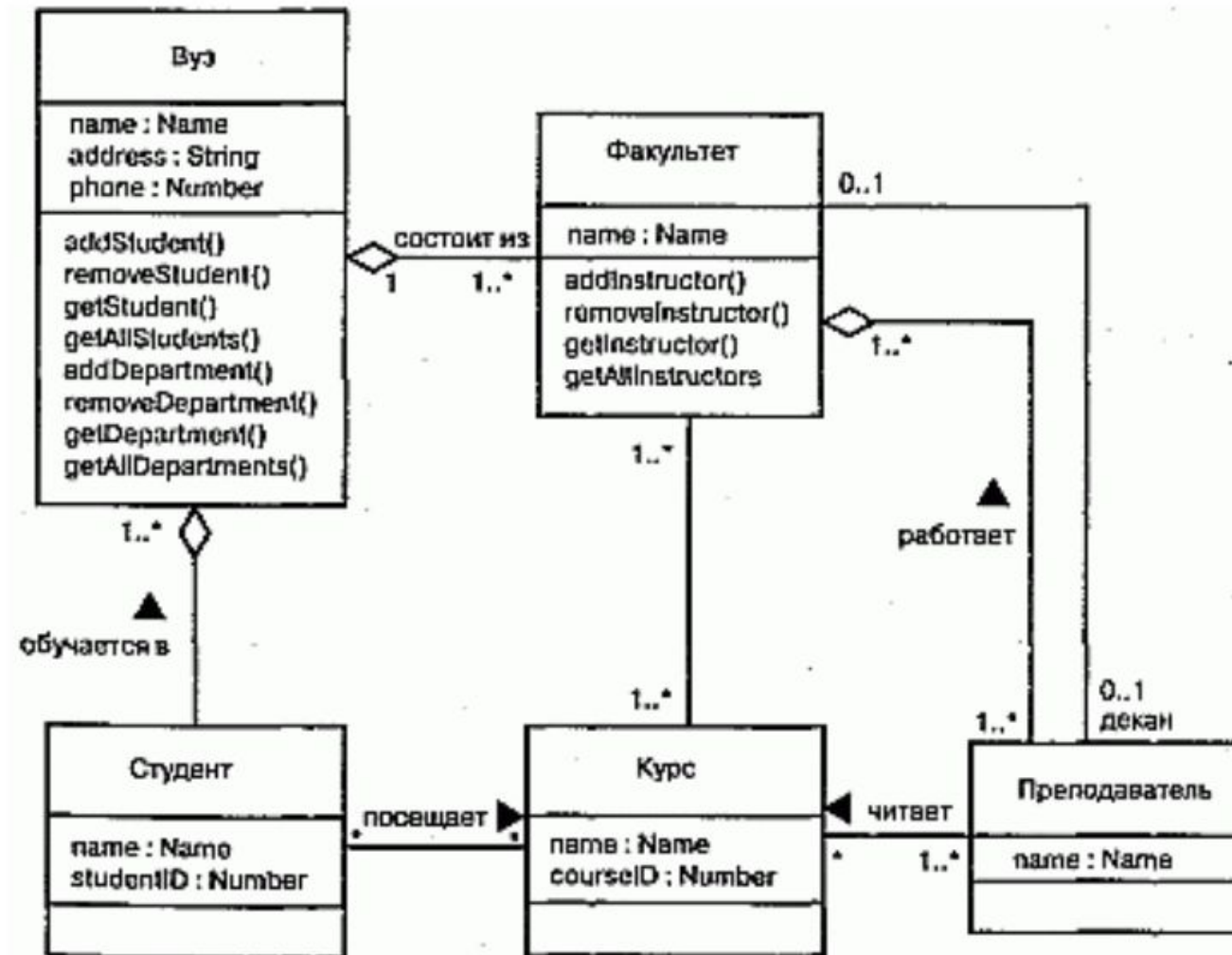
Система
«Автономный
робот»

Кооперация:
движение
по заданной
траектории



Применение диаграмм классов (2)

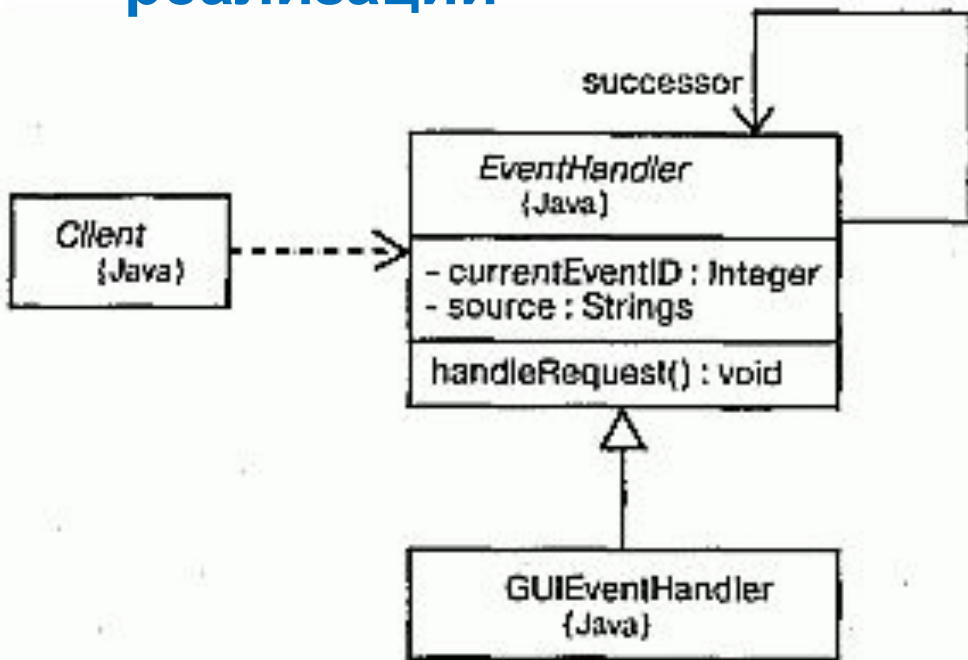
- Проектирование **логической схемы базы данных**



Применение диаграмм классов (3)

- Прямое и обратное проектирование

Модель реализации



Реализация в программном коде

```
public abstract class EventHandler {
    EventHandler successor;
    private Integer currentEventID;
    private String source;
    EventHandler() {}
    public void handleRequest() {}
}
```

Хорошая диаграмма классов

- ✓ заостряет внимание только на одном аспекте
- ✓ содержит лишь элементы, существенные для понимания данного аспекта;
- ✓ показывает детали, соответствующие требуемому уровню абстракции, опуская те, без которых можно обойтись;
- ✓ не настолько проста и лаконична, чтобы ввести читателя в заблуждение относительно важных аспектов.



Резюме: рассмотренные вопросы

- Что такое диаграмма классов? Какие основные элементы диаграммы отображаются при моделировании?
- Какие существуют разновидности структурных отношений между классами?
- Чем характеризуется отношение ассоциации?
- В чем отличие композиции и агрегации?
- В чем суть отношения обобщения? Как соотносятся суперкласс и подклассы? Как моделировать отношение обобщения?
- Как применяются диаграммы классов в моделировании?
- Как построить *хорошую* диаграмму классов?