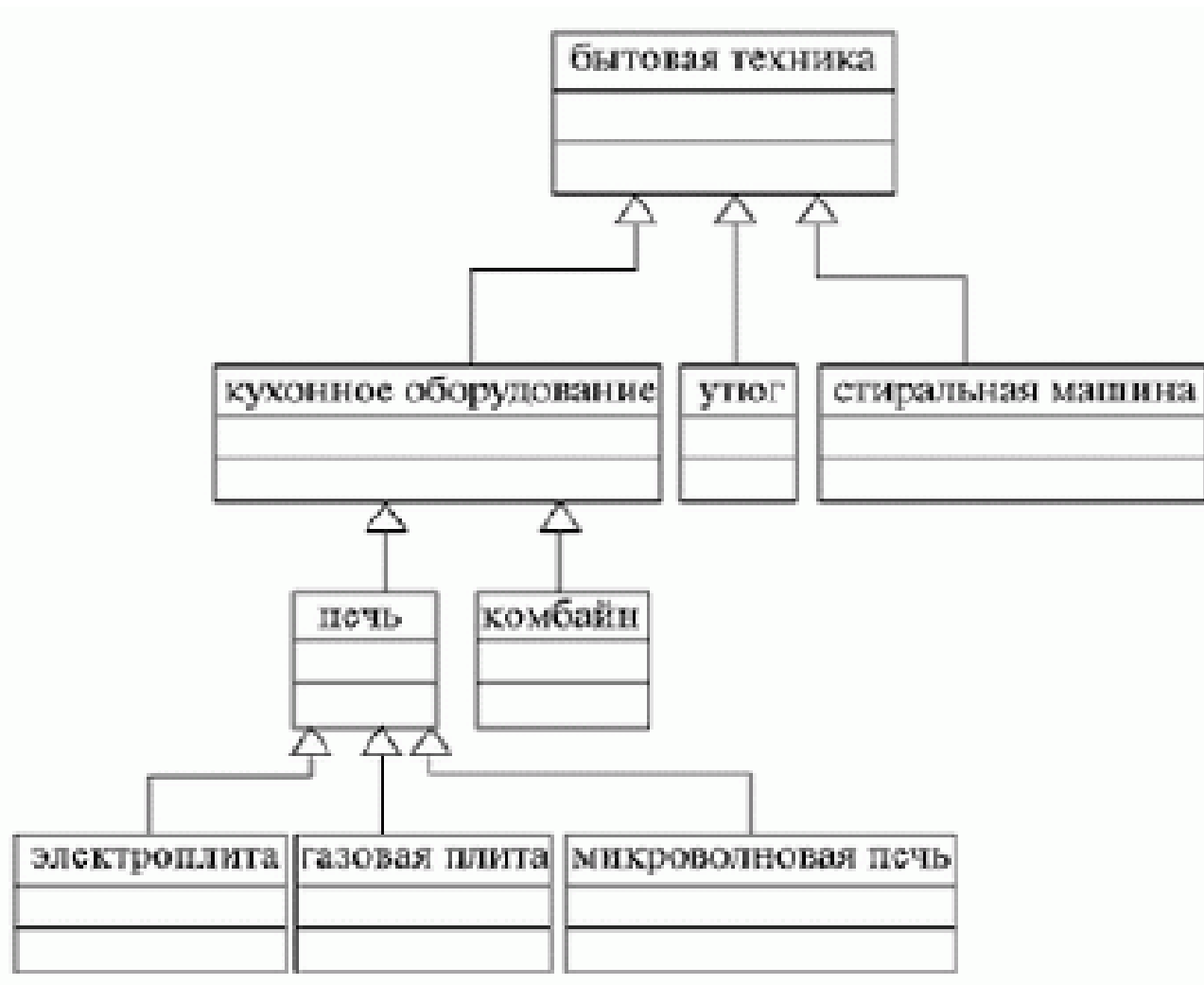
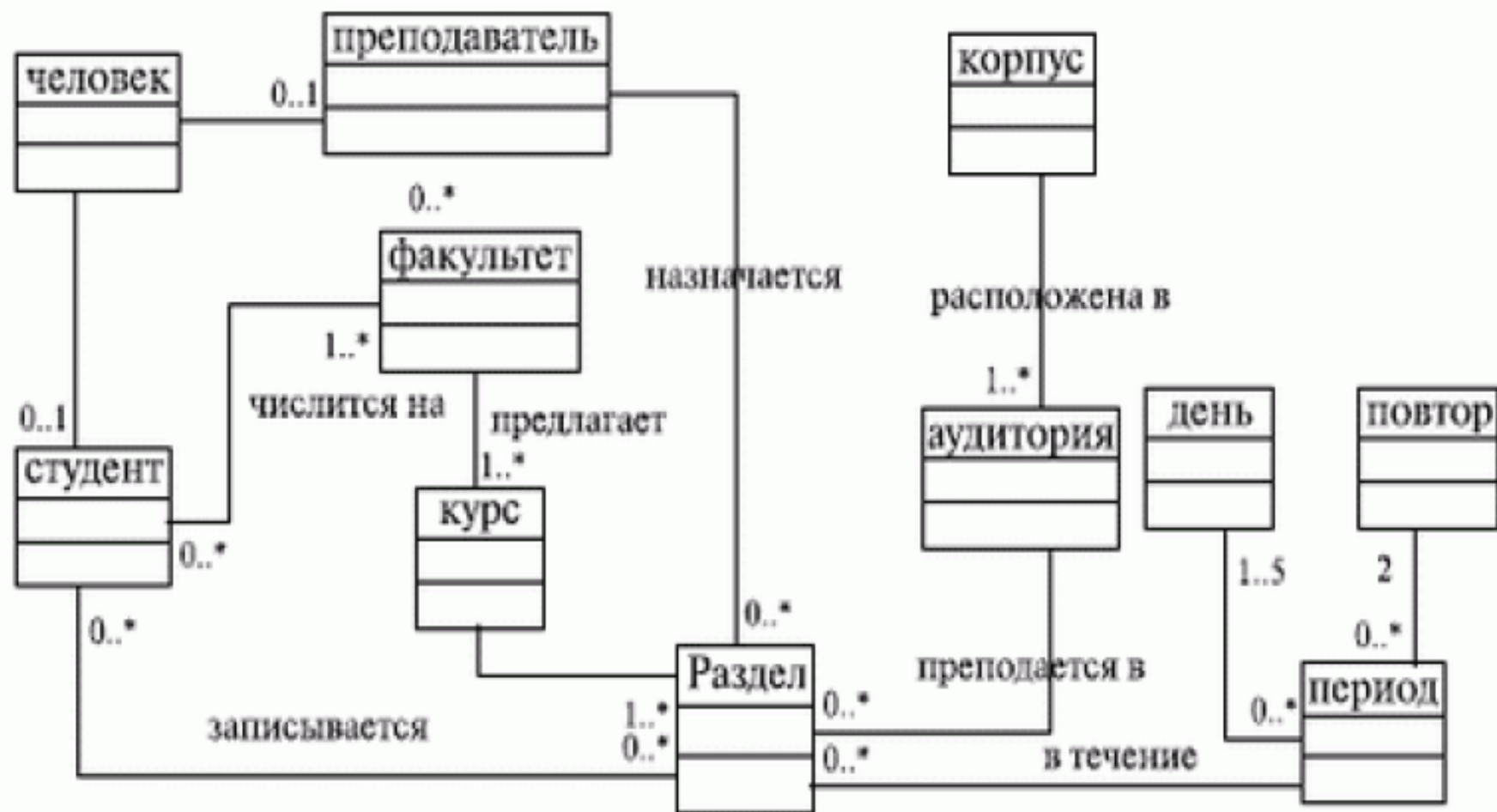


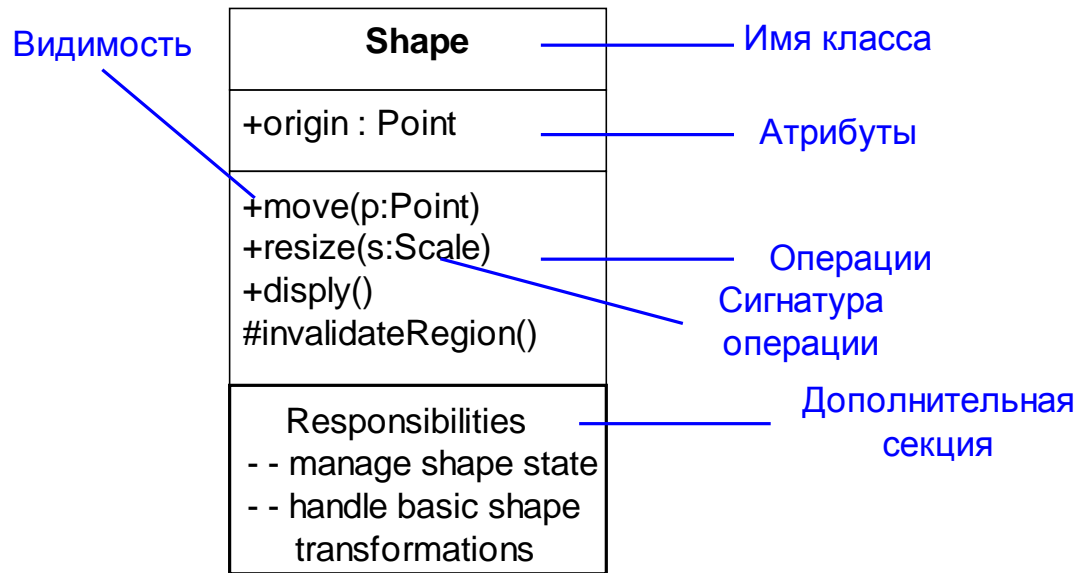
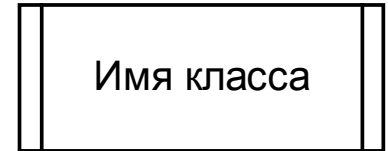
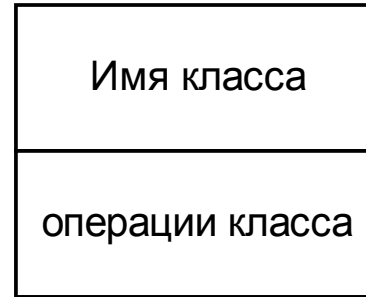
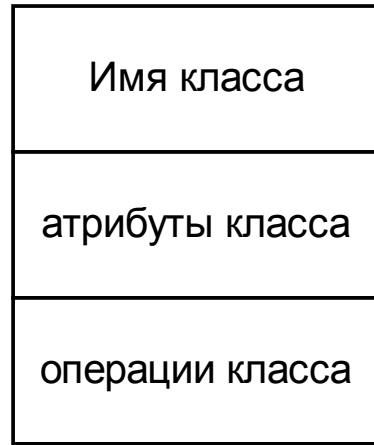
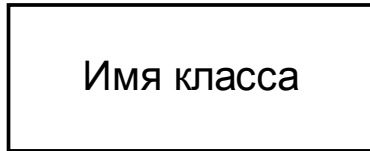
Диаграммы классов







Некоторые курсы не требуют инструктора и рассчитаны на самостоятельное изучение



*<атрибут> ::= [<видимость>] ["/"] <имя> [":"
<тип атрибута>] ["["<кратность>"]"] [=] <з
начение по умолчанию> [{"<модификатор
атрибута> ["," <модификатор атрибута>]*
"}]*

Телевизор

- + Язык экранного меню
- Частота каналов
- + Порядок и именование каналов
- + ...

- Самодиагностика()
- + Включить()
- + Выключить()
- + Поиск каналов()
- Декодирование сигнала()
- + Переключение каналов()
- + ...()

*<кратность> ::= <диапазон-
кратности> ['{' <указатель-
упорядоченности> [';'
<указатель-уникальности>] '}']*

+ имяСотрудника : String {readOnly}

~ датаРождения : Data {readOnly}

/возрастСотрудника : Integer

+ номерТелефона : Integer [1..*] {unique}

– заработнаяПлата : Currency = 500.00

Операции класса

<операция> ::= [<видимость>*]*

<имя операции> '(' [<список**

параметров>] ')' [*':'* [*<тип*

возвращаемого результата>] '{'

<свойство операции> [','* *<свойство**

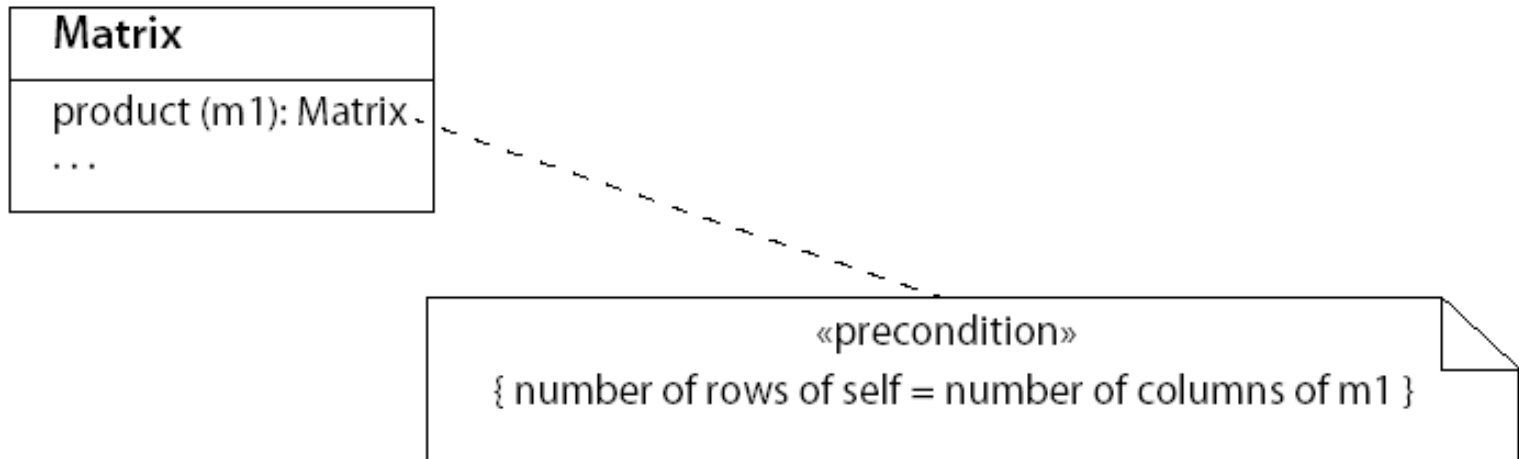
операции>] '}'*]

Табл. Ключевые слова для описания направления передачи параметров

Ключевое слово	Назначение параметра
in	Входной параметр – аргумент должен быть значением, которое используется в операции, но не изменяется
out	Выходной параметр – аргумент должен быть хранилищем, в которое операция помещает значение
inout	Входной и выходной параметр – аргумент должен быть хранилищем, содержащим значение. Операция использует переданное значение аргумента и помещает в хранилище результат
return	Значение, возвращаемое операцией. Такое значение направления передачи устанавливается автоматически для возвращаемого значения

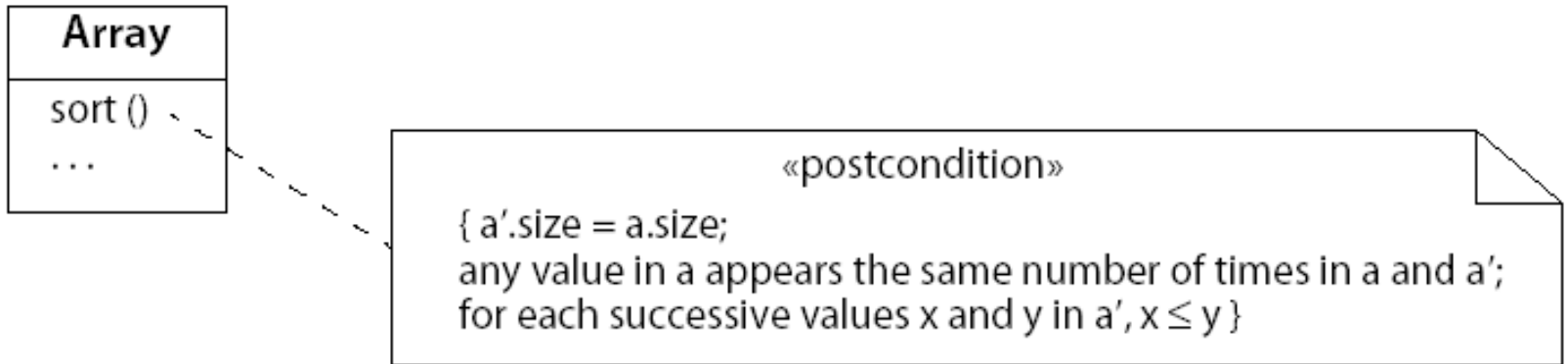
Предусловие (precondition) операции

- – определяет условие, которое должно быть истинным, когда эта операция вызывается



Постусловие (postcondition) операции

- – определяет условие, которое должно быть истинным, когда вызов операции успешно завершился, в предположении, что все предусловия были удовлетворены



Примеры записи операций:

+добавить(in номерТелефона : Integer [*]
{unique})

–изменить(in заработнаяПлата : Currency)

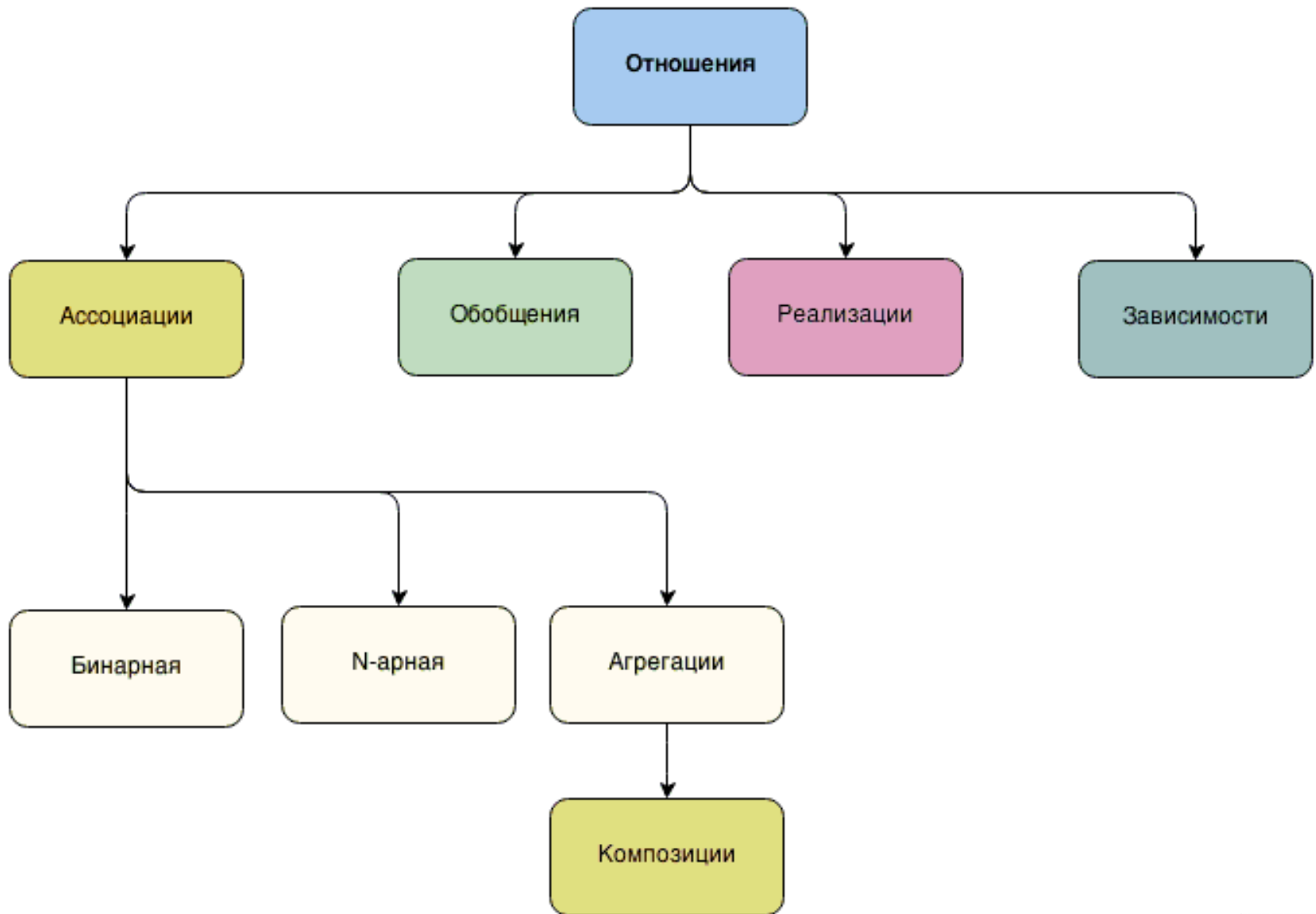
+создать() : Boolean

toString(return : String)

toString() : String

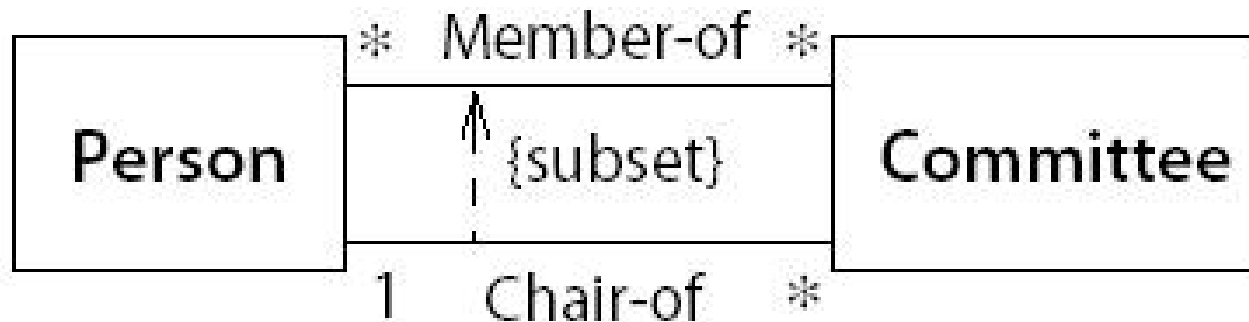
Отношения на диаграмме классов

<i>Relationship</i>	<i>Function</i>	<i>Notation</i>
association	Представление произвольного отношения между экземплярами классов	
generalization	Отношение типа "Общее-Частное", обладающая свойством наследования свойств	
aggregation	Отношение типа "Часть-Целое"	
composition	Более сильная форма отношения типа "Часть-Целое"	
realization	Отношение между спецификацией и ее выполнением	
dependency	Направленное отношение между двумя элементами модели с открытой семантикой	



Строка свойство (property string)

- $\{\text{subset } \langle \text{имя элемента} \rangle\}$ – конец ассоциации представляет собой некоторое подмножество $\langle \text{имя элемента} \rangle$, в качестве которого может выступать имя конца ассоциации или атрибута класса

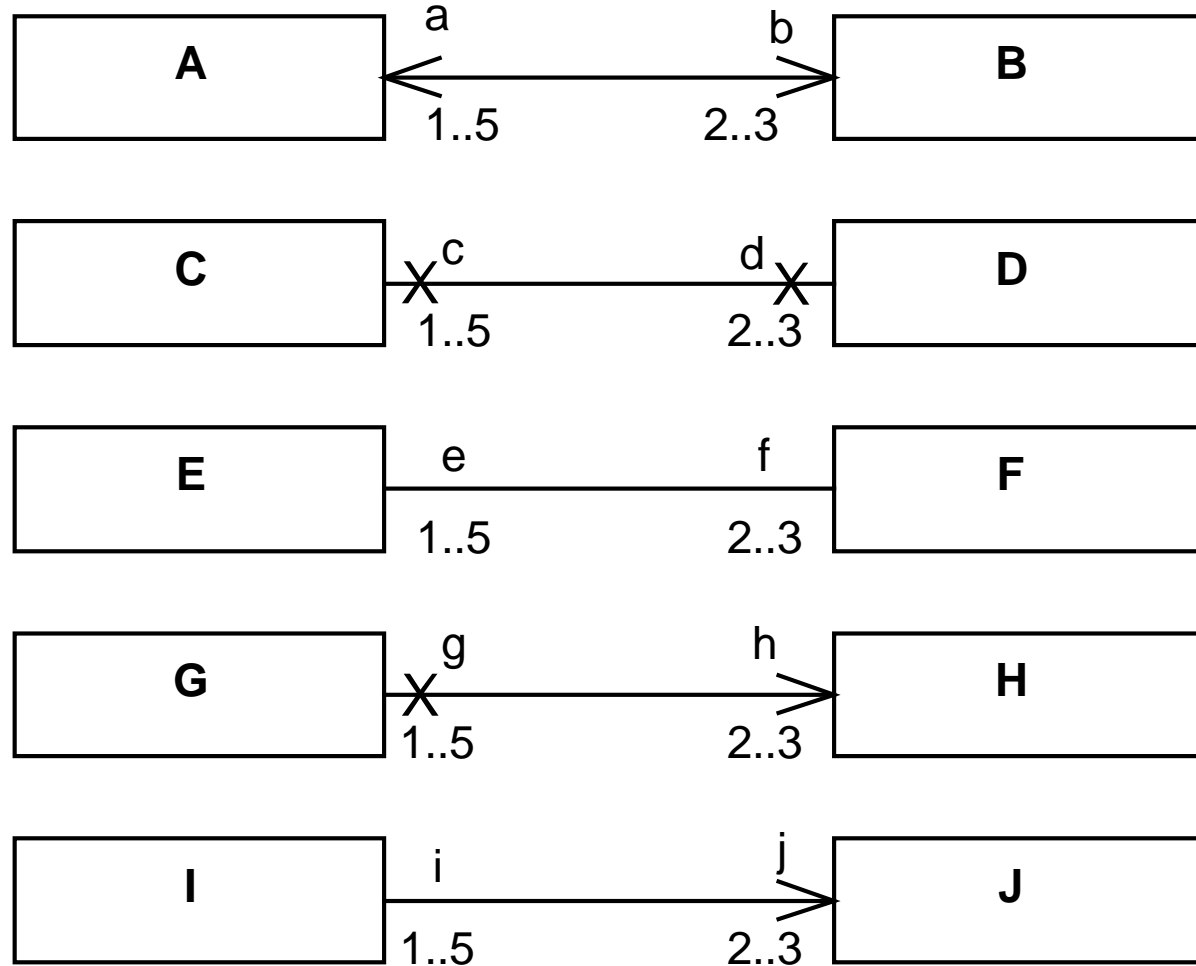


binary constraint: attached to dashed arrow

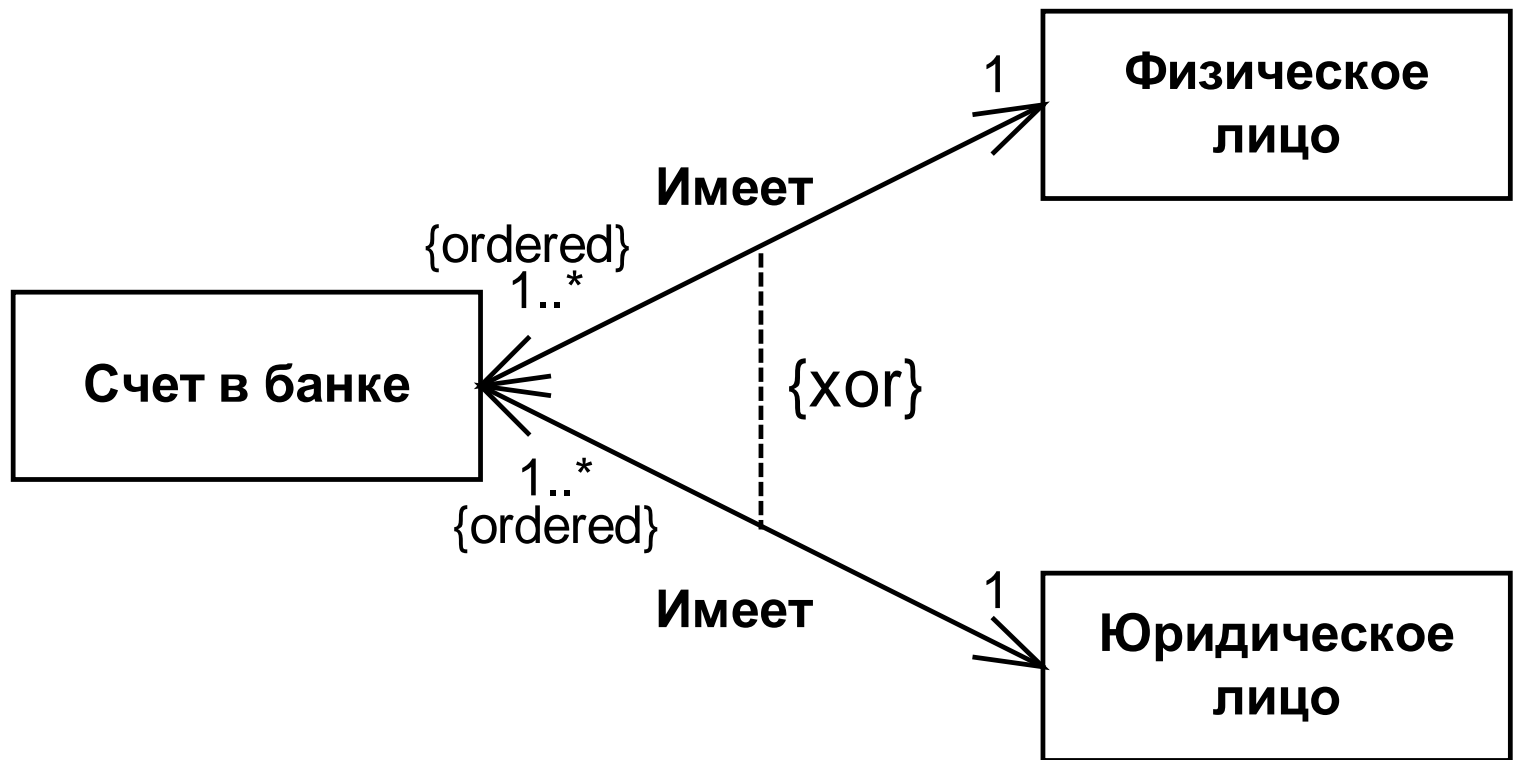
Ассоциация с навигацией и эквивалентное ему представление класса с атрибутом



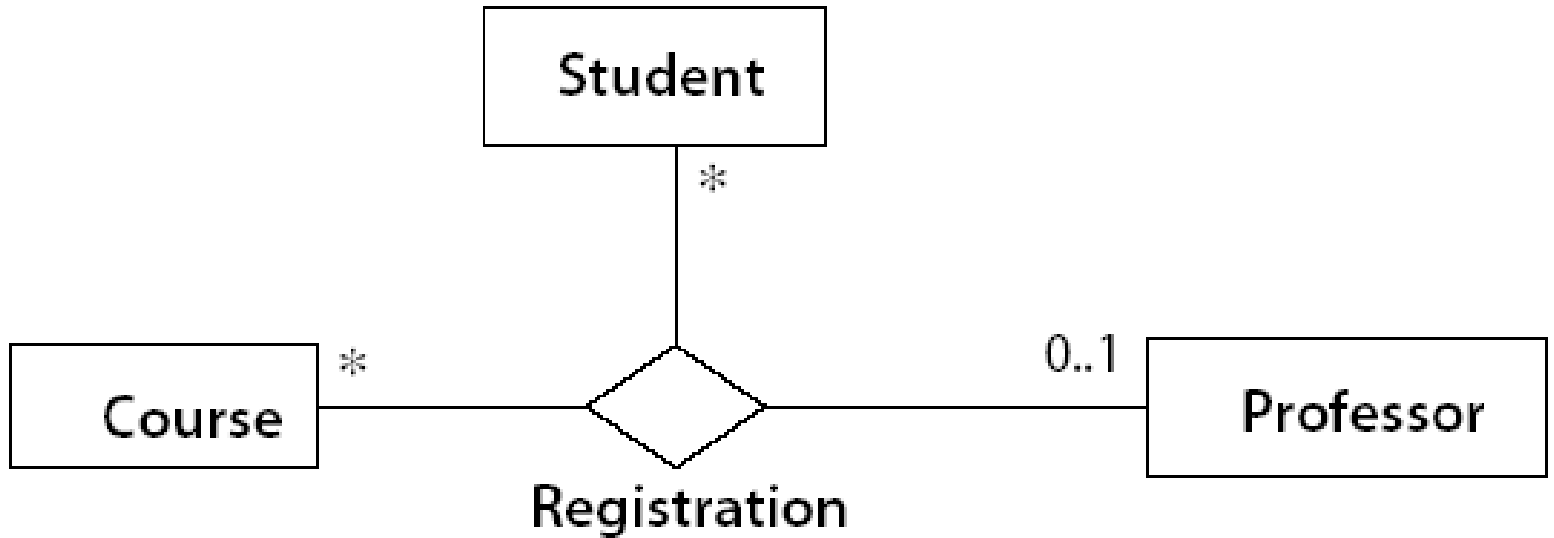
Варианты изображения навигации и кратности у концов ассоциации



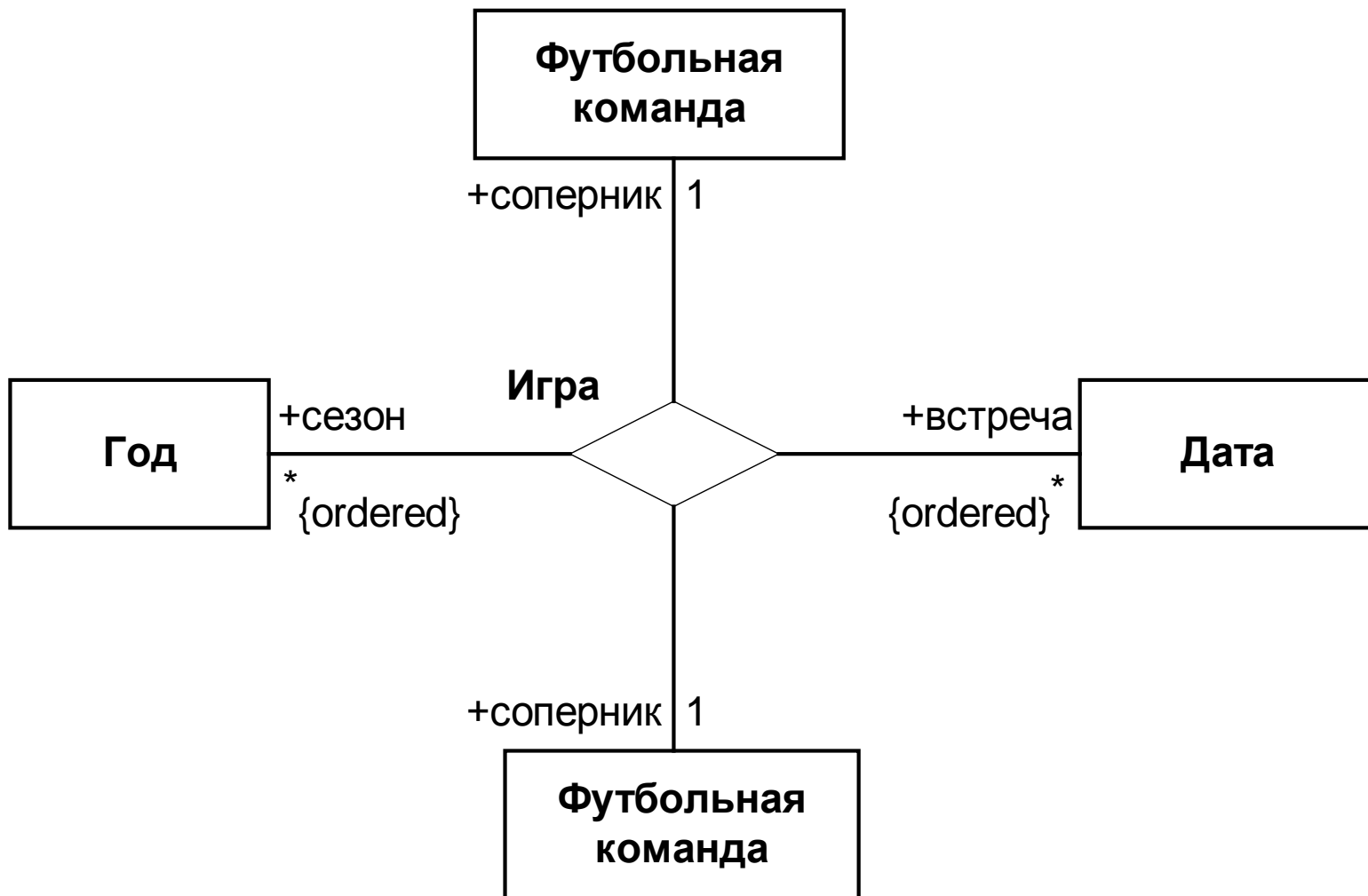
Исключающая ассоциация между тремя классами



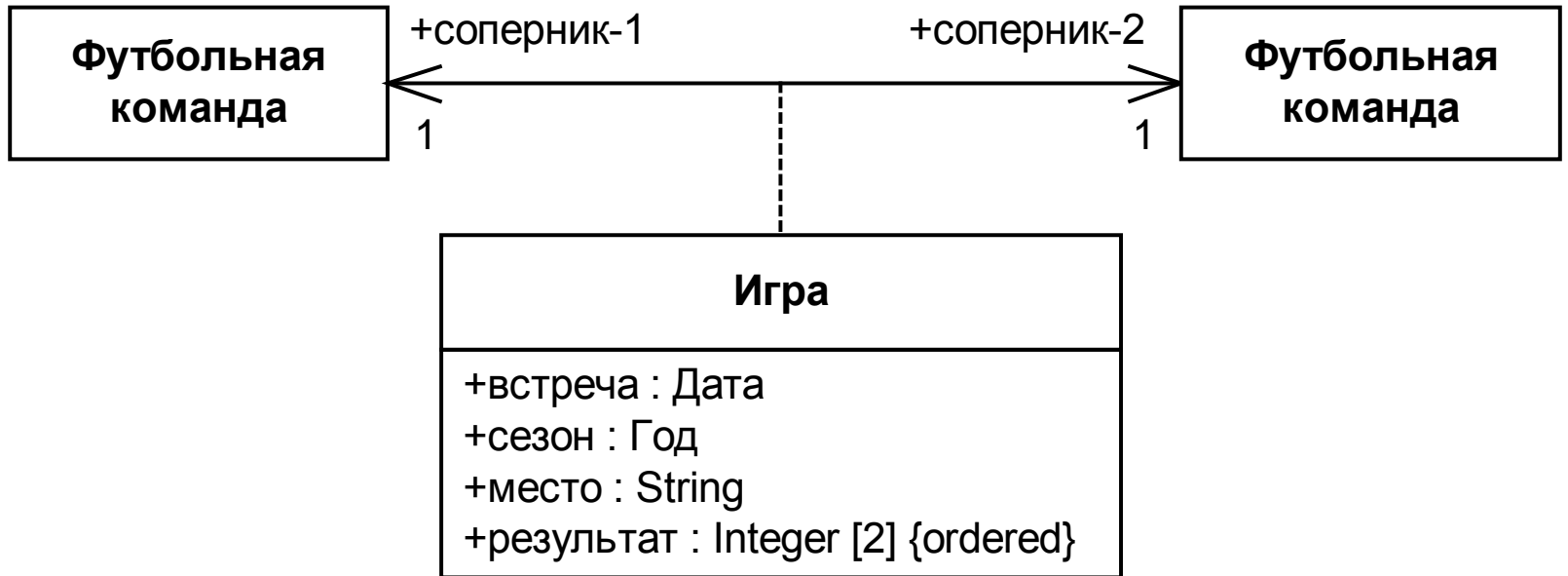
Пример тернарной ассоциации



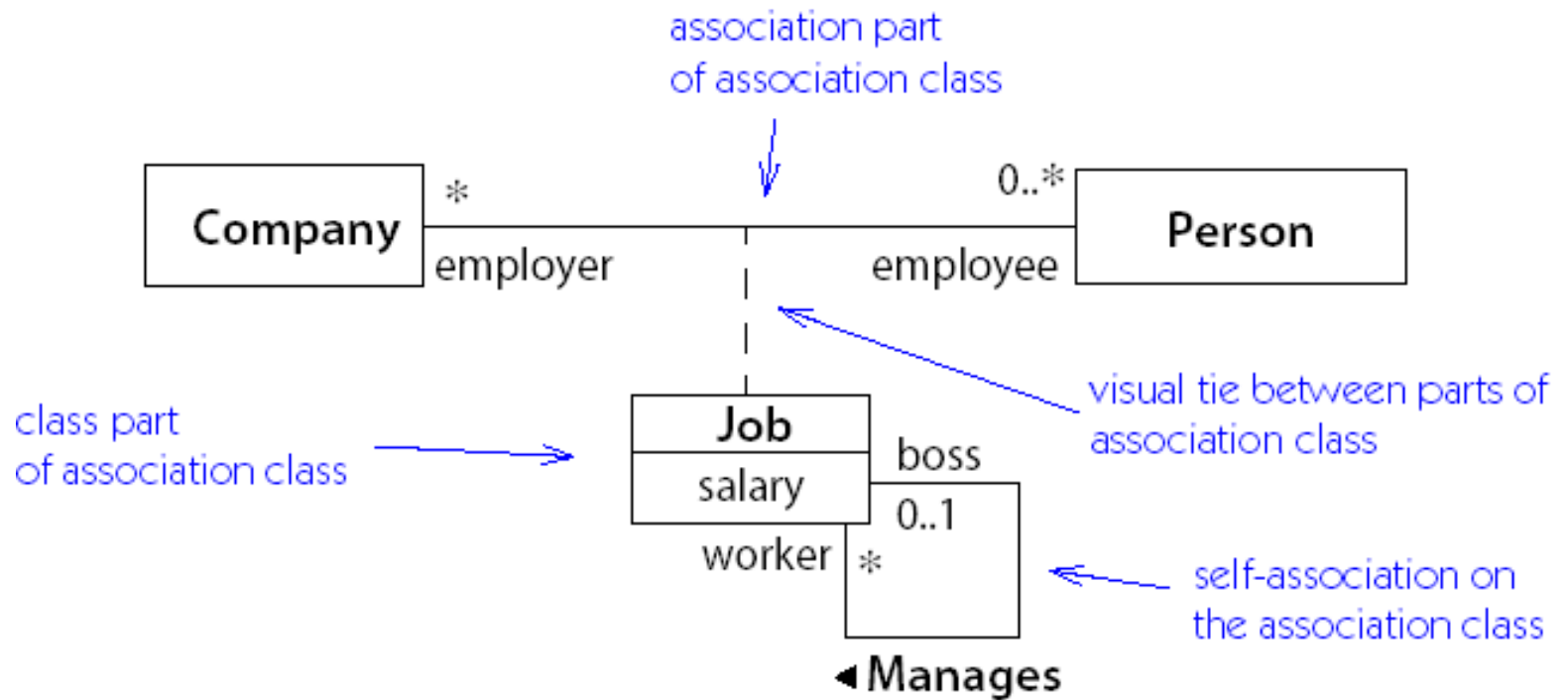
Пример 4-арной ассоциации



Ассоциация класс (*association class*)

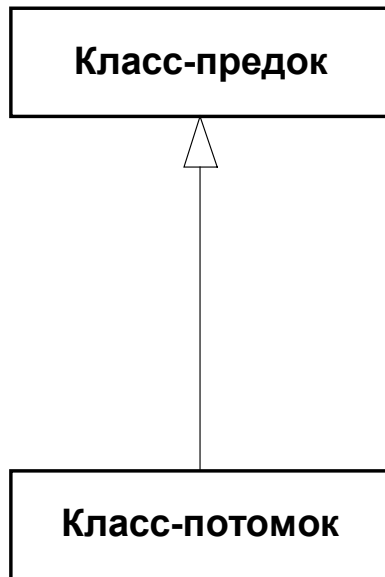


Примеры ассоциации класса и рефлексивной ассоциации

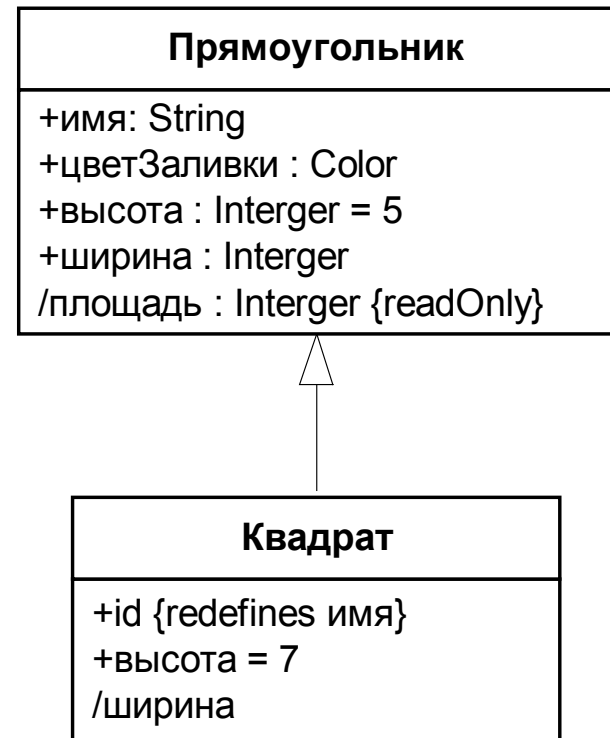


Обобщение (*generalization*)

- таксономическое отношение между более общим классификатором (родителем или предком) и более специальным классификатором (дочерним или потомком)

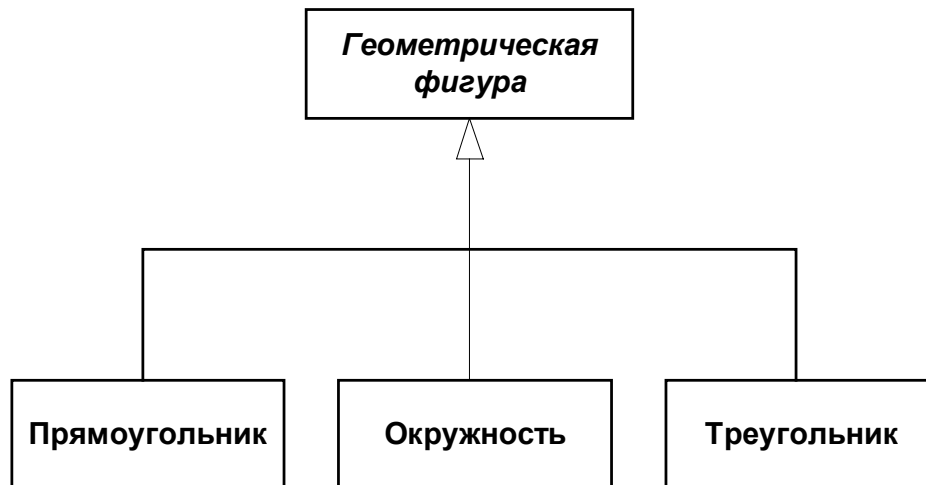
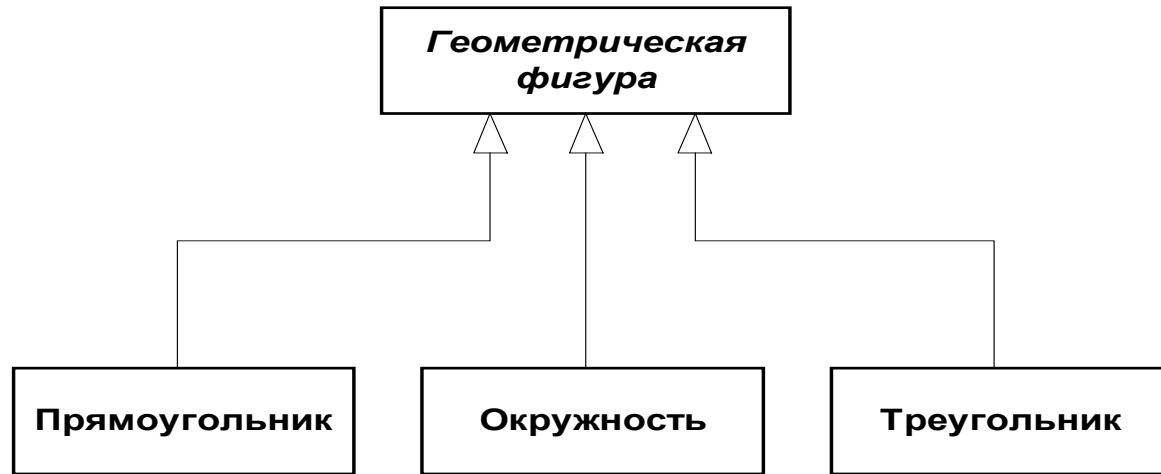


(a)

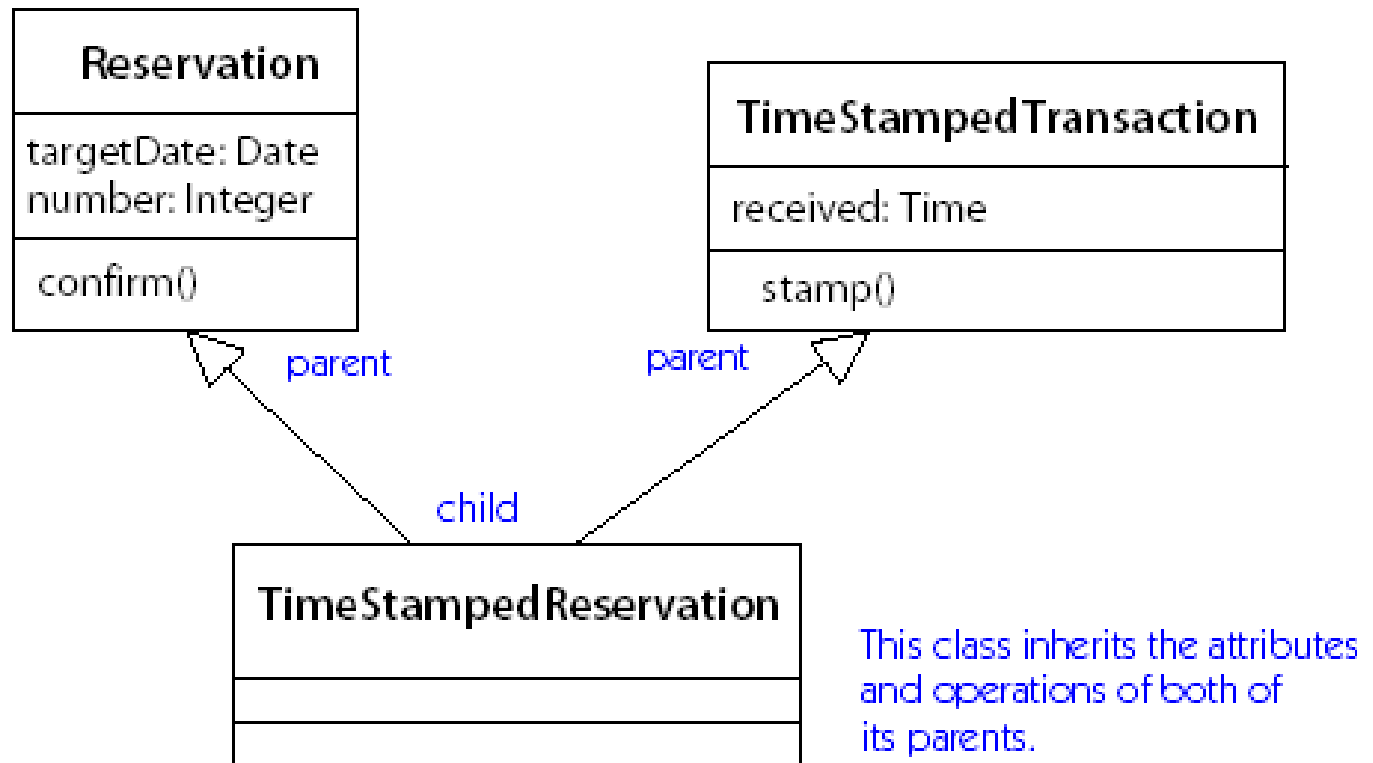


(б)

Примеры отношения обобщения



Множественное наследование – в языке UML разрешено



No new features are needed by the child.

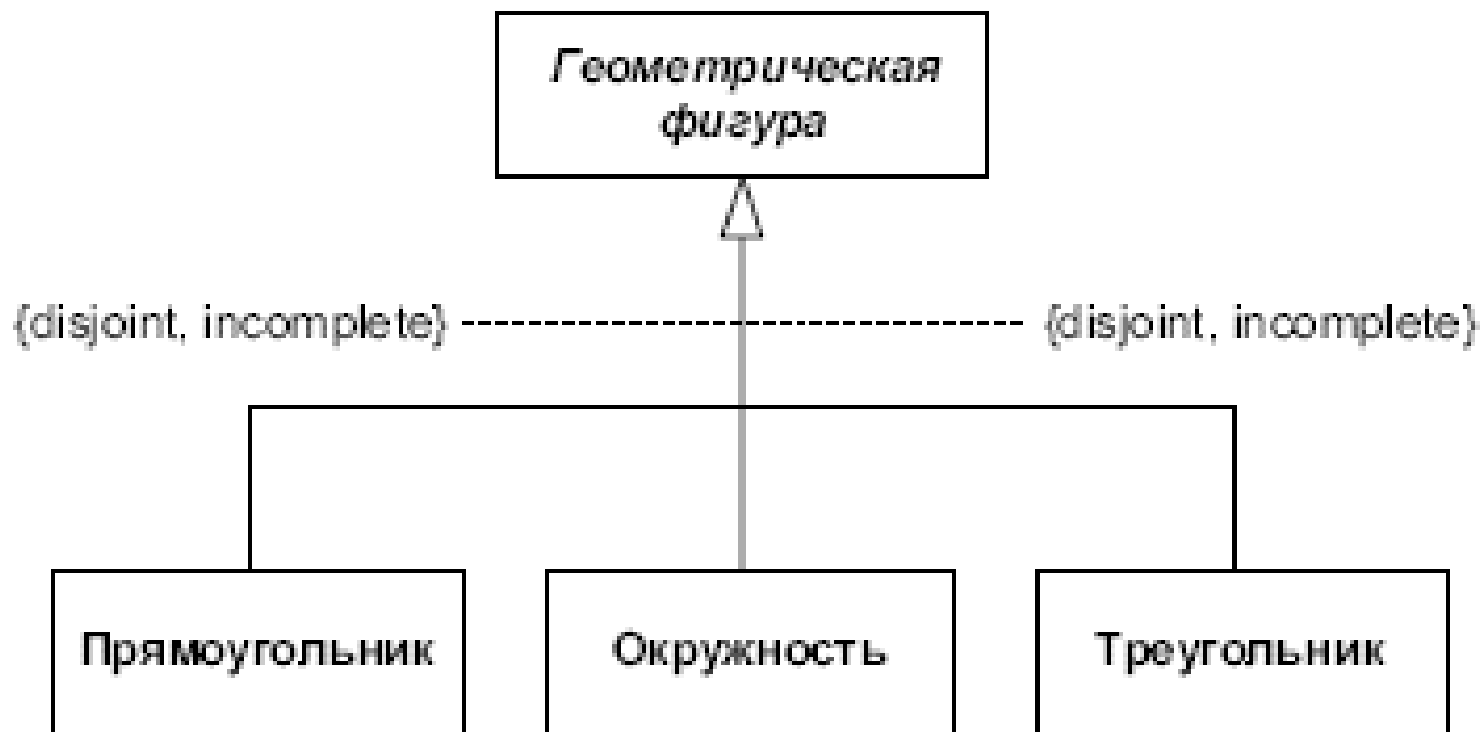
{complete}

{incomplete}

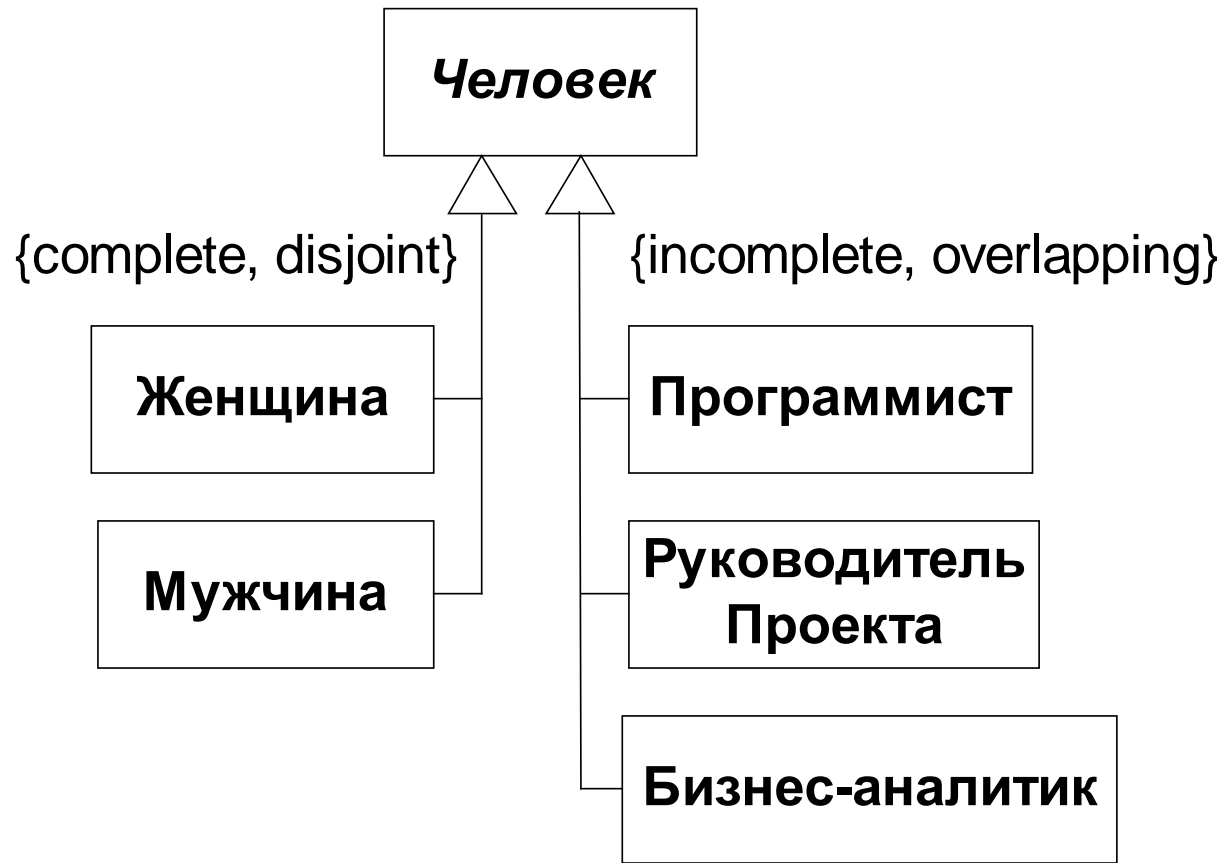
{disjoint}

{overlapping}

По умолчанию - {incomplete, disjoint}



Примеры ограничений на множество обобщения

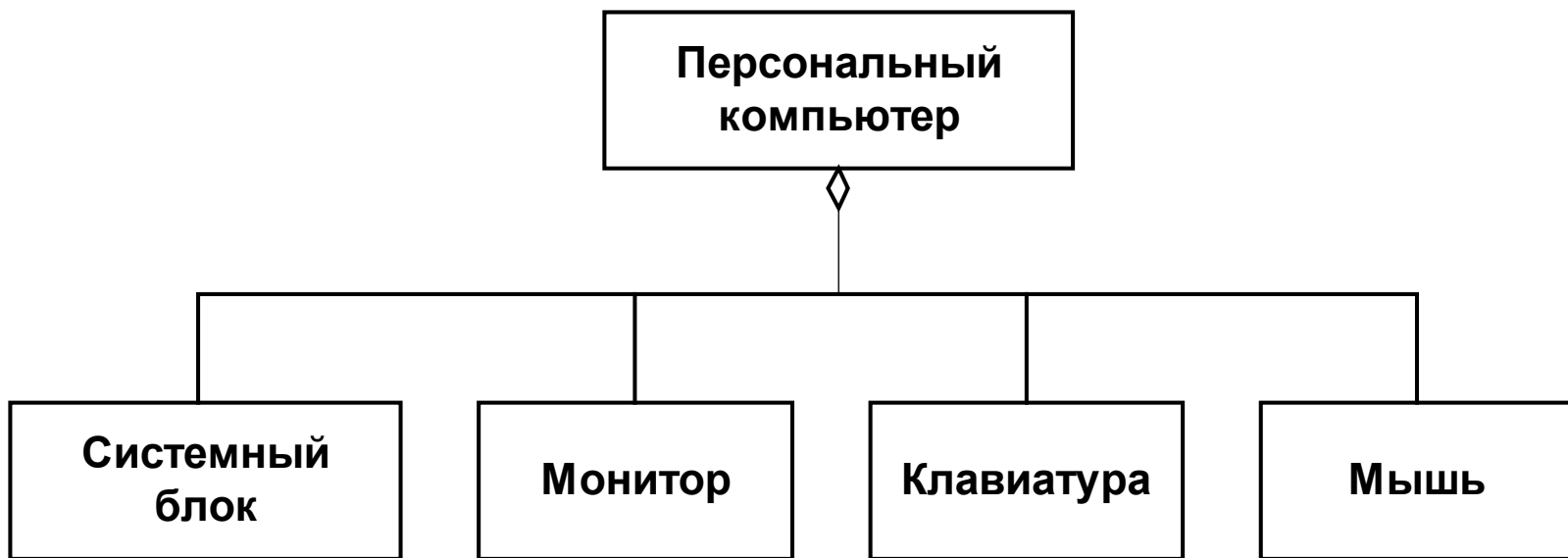


Агрегация (*aggregation*)

- – направленное отношение между двумя классами, предназначенное для представления ситуации, когда один из классов представляет собой некоторую сущность, которая включает в себя в качестве составных частей другие сущности

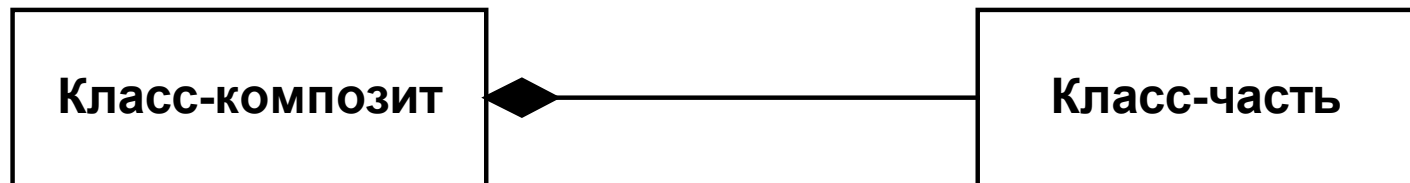


Пример отношения агрегации

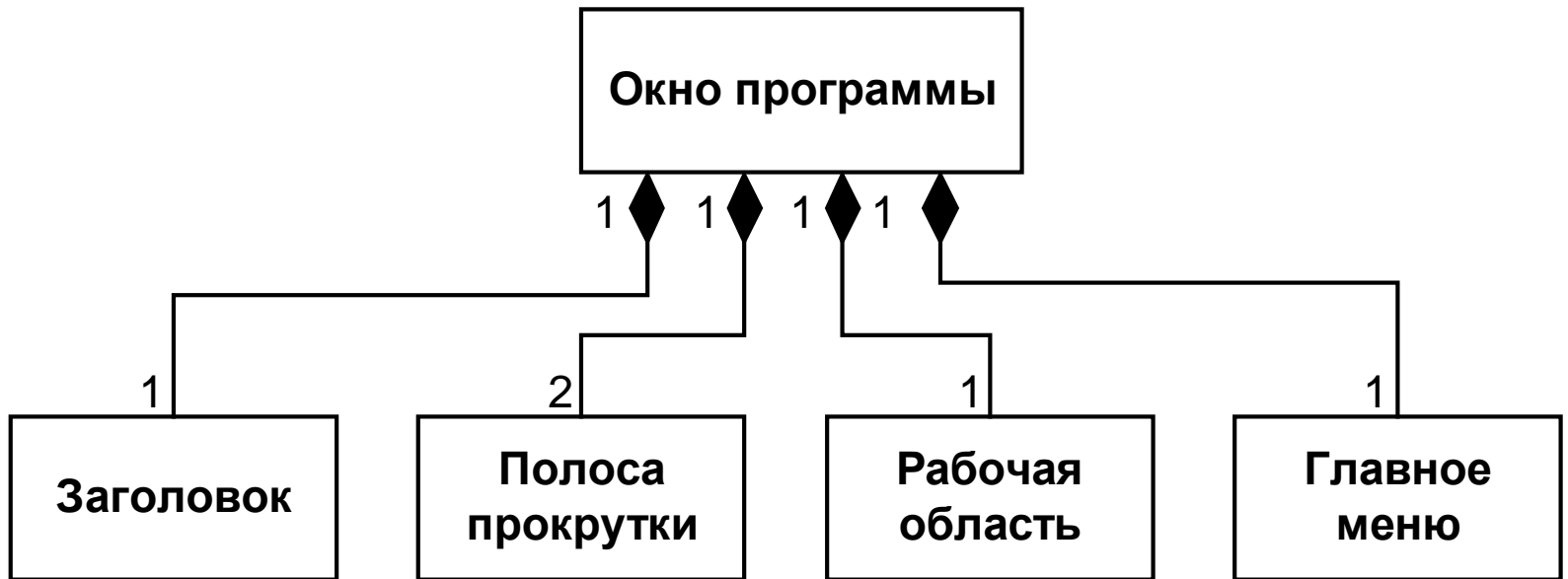


Композиция (*composition*)

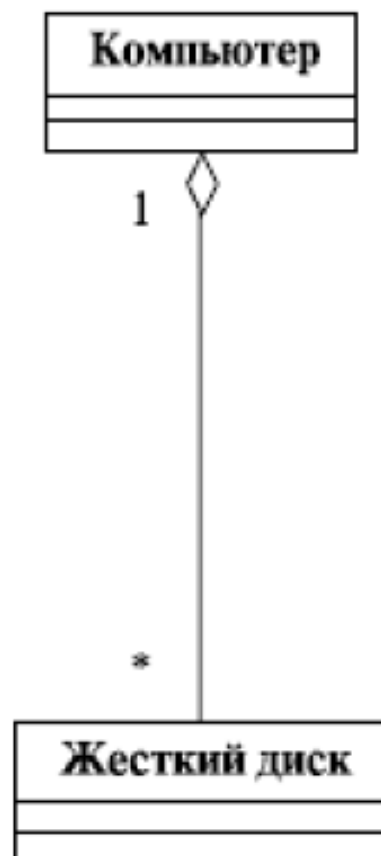
- или *комполитная агрегация* предназначена для спецификации более сильной формы отношения "часть-целое", при которой с уничтожением объекта класса-контейнера уничтожаются и все объекты, являющимися его составными частями.



Пример отношения КОМПОЗИЦИИ



простое
агрегирование



композиционное
агрегирование

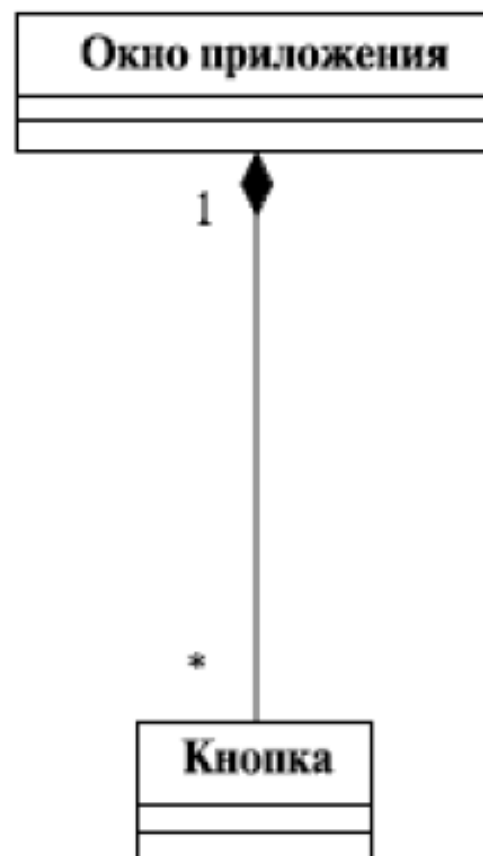
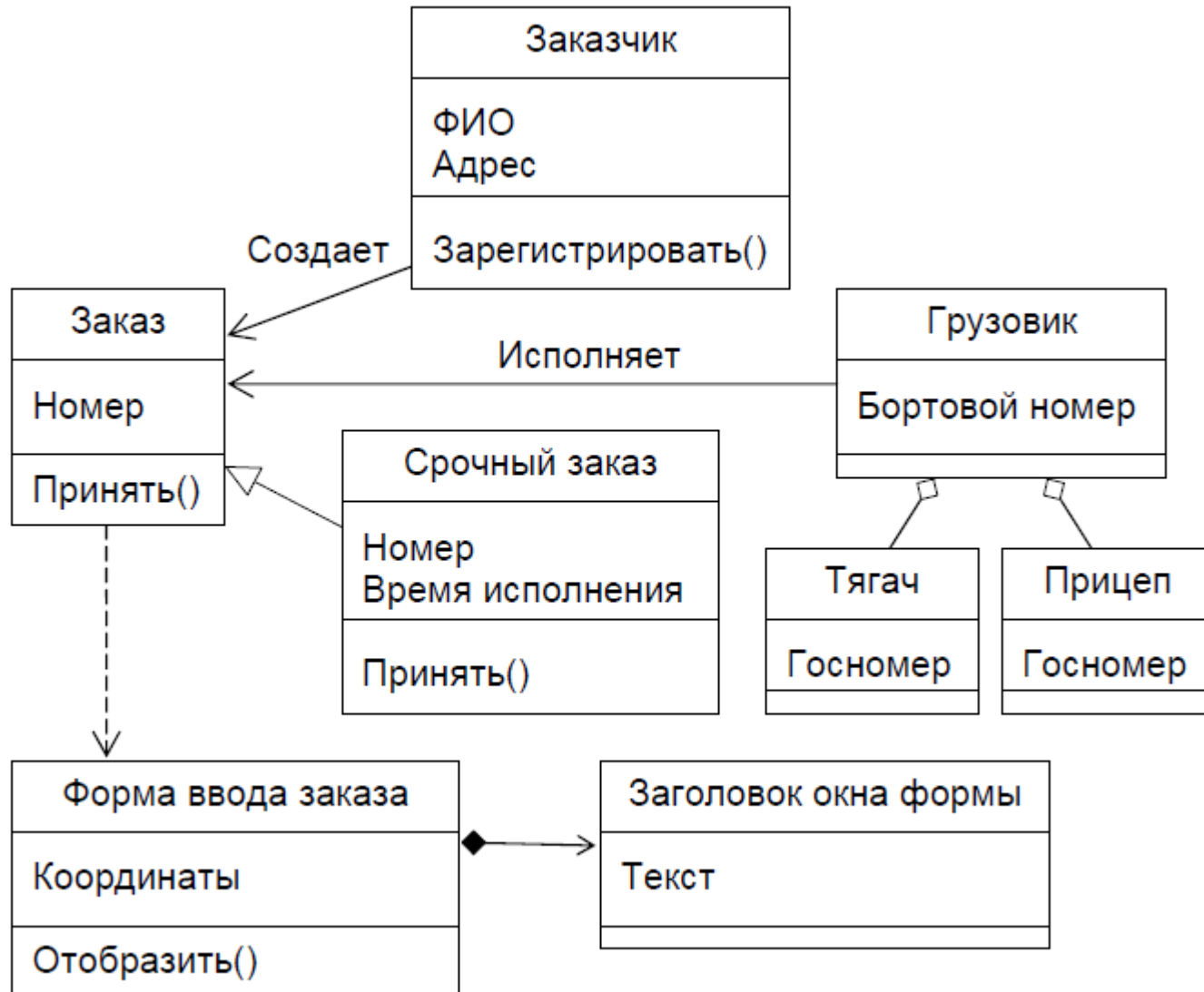
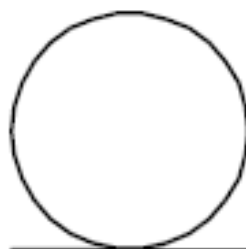


Диаграмма классов

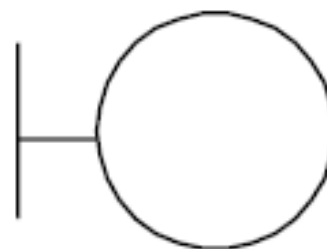




управляющий класс



класс-сущность



граничный класс

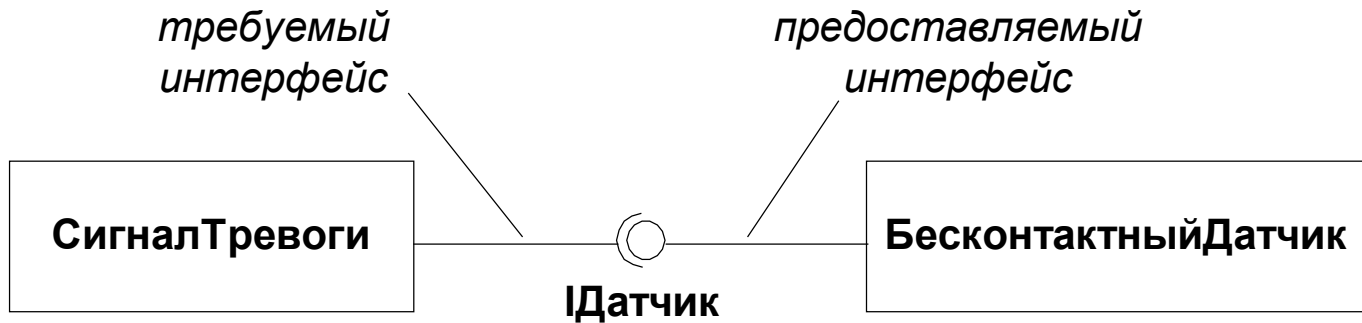
`<<control>>`
Имя класса

`<<entity>>`
Имя класса

`<<boundary>>`
Имя класса

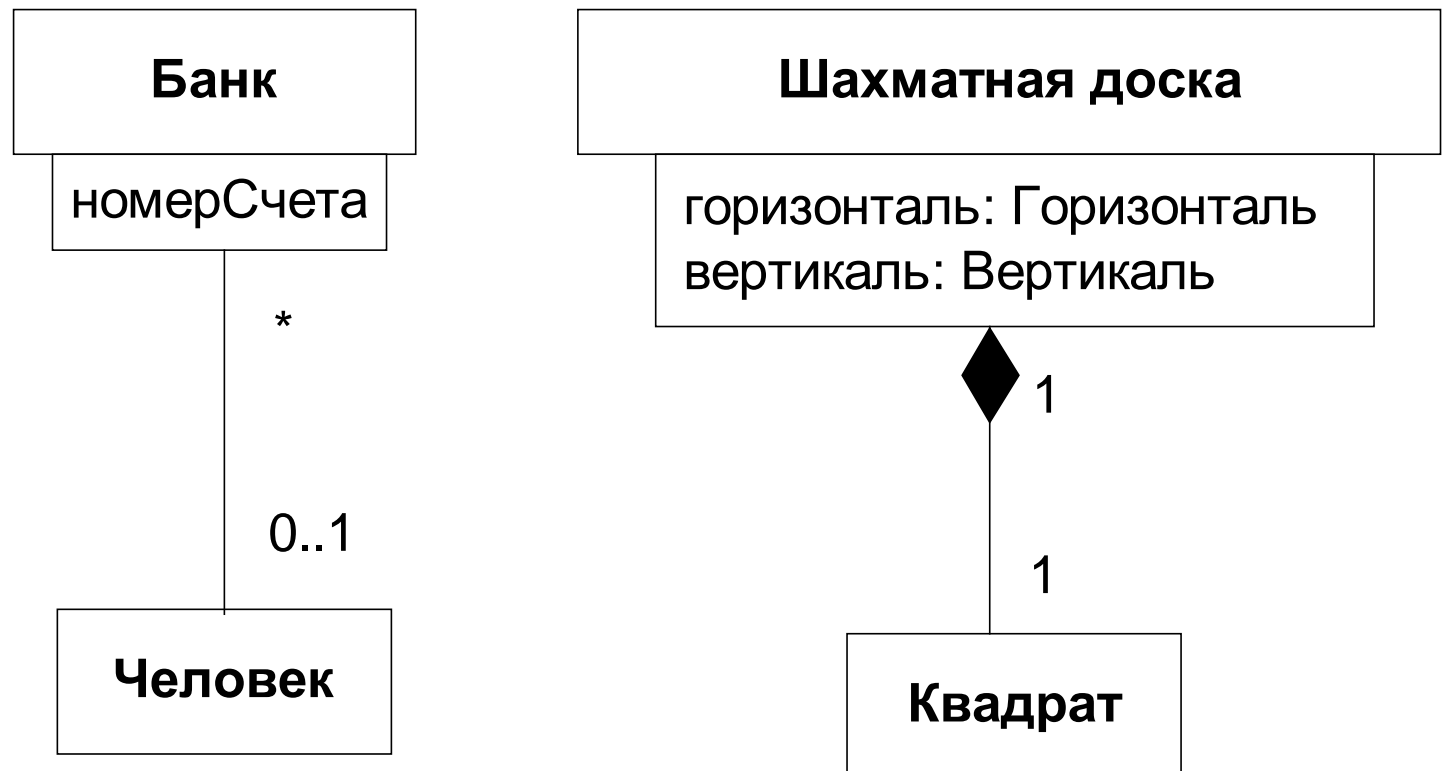
Интерфейс (*interface*)

- вид класса, который представляет собой объявление множества общедоступных характеристик и обязанностей.



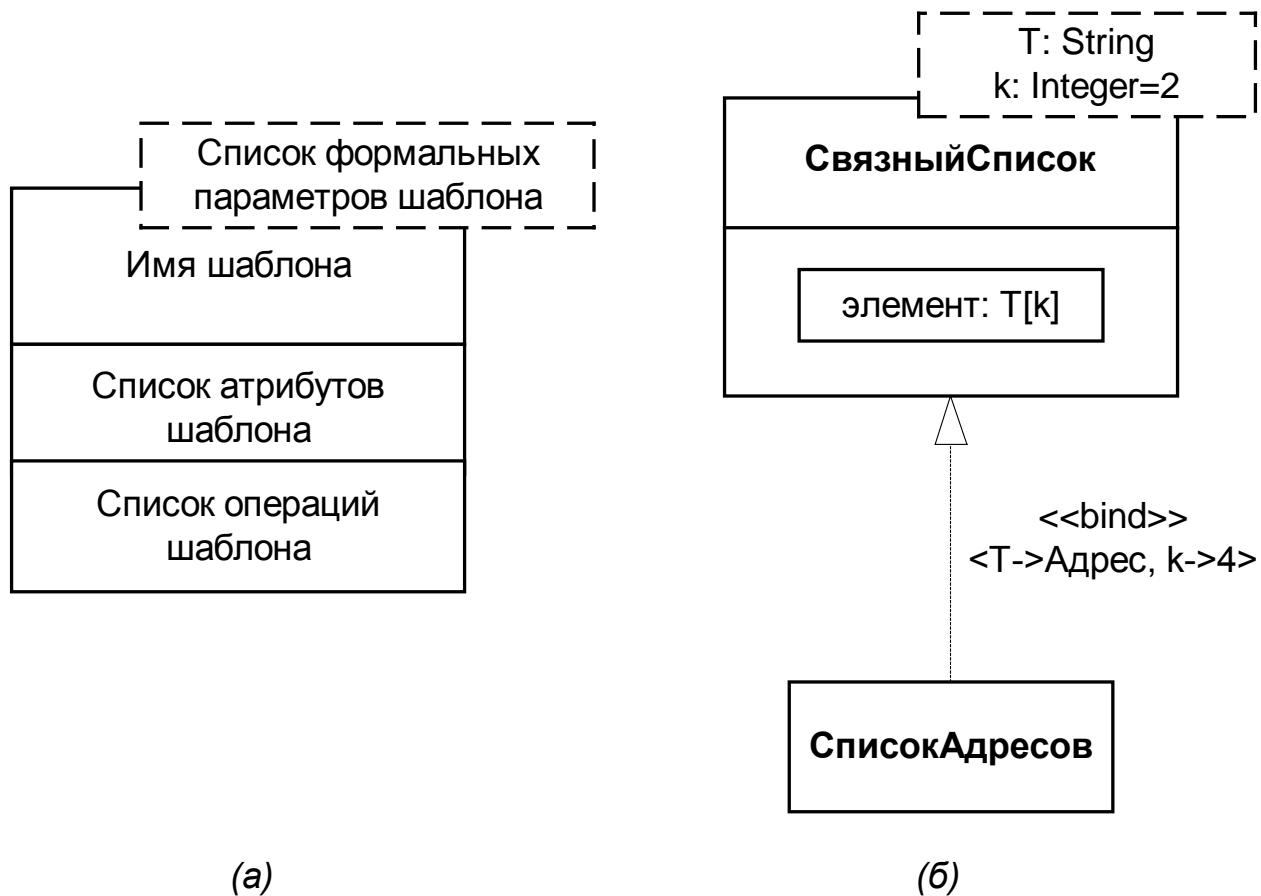
Квалификатор (*qualifier*)

- *Квалификатор* (*qualifier*) объявляет разбиение множества ассоциированных экземпляров относительно экземпляра на квалифицированном конце ассоциации

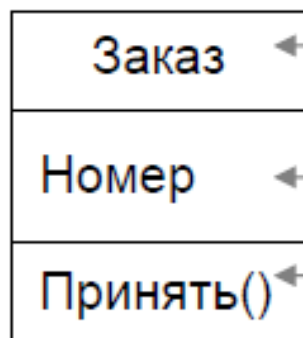


Шаблон (*template*)

- классификатор, который в своем описании имеет несколько формальных параметров



Графические примитивы диаграммы классов UML



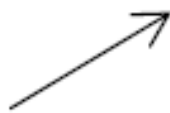
← Название

← Атрибуты

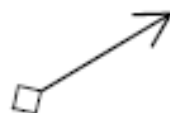
← Методы

Класс

○ Интерфейс



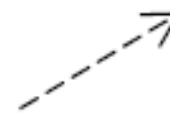
Ассо-
циация



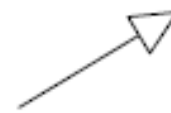
Агрега-
ция



Компо-
зиция



Зависи-
мость



Обоб-
щение