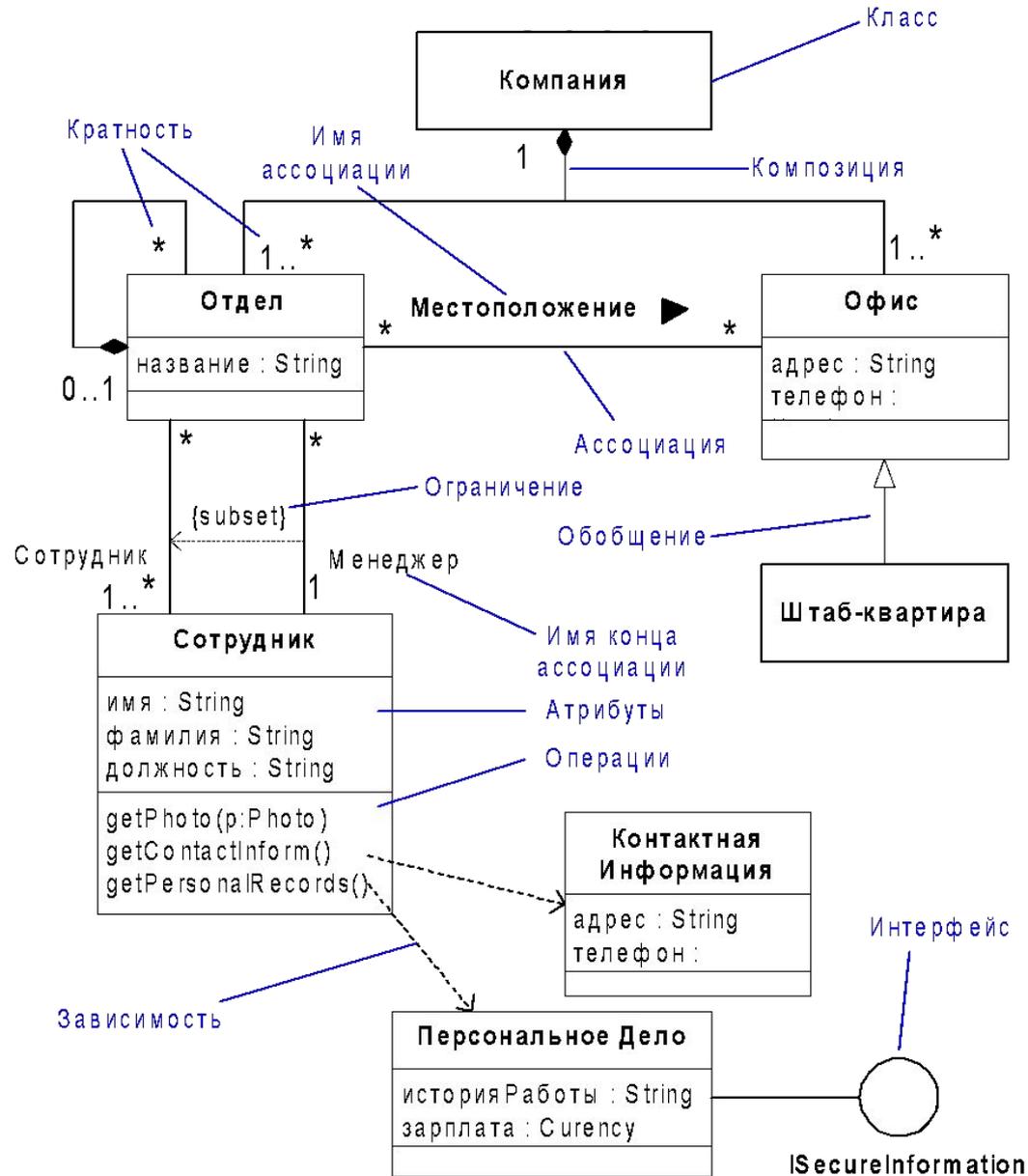


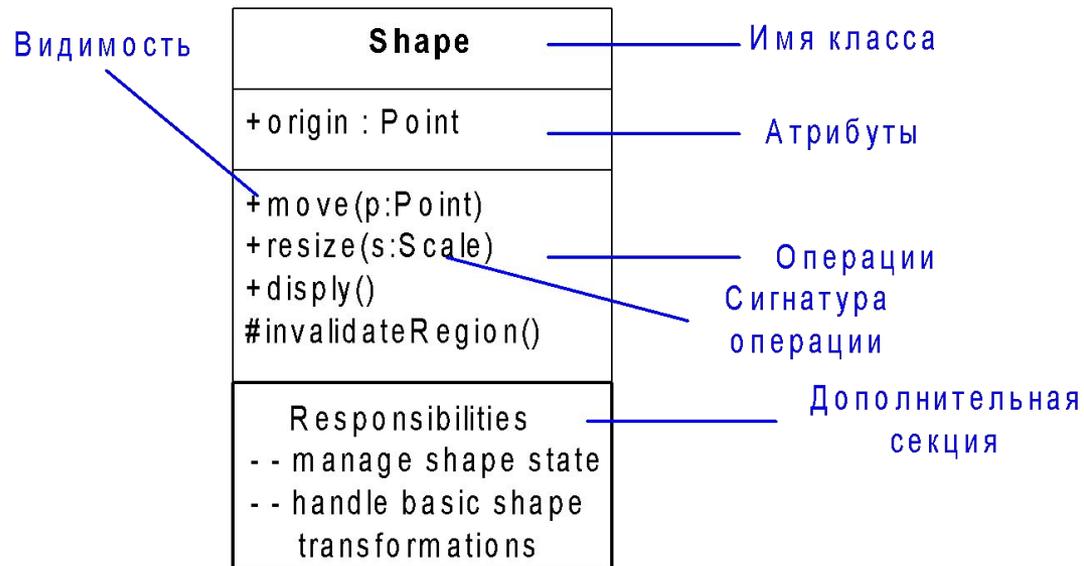
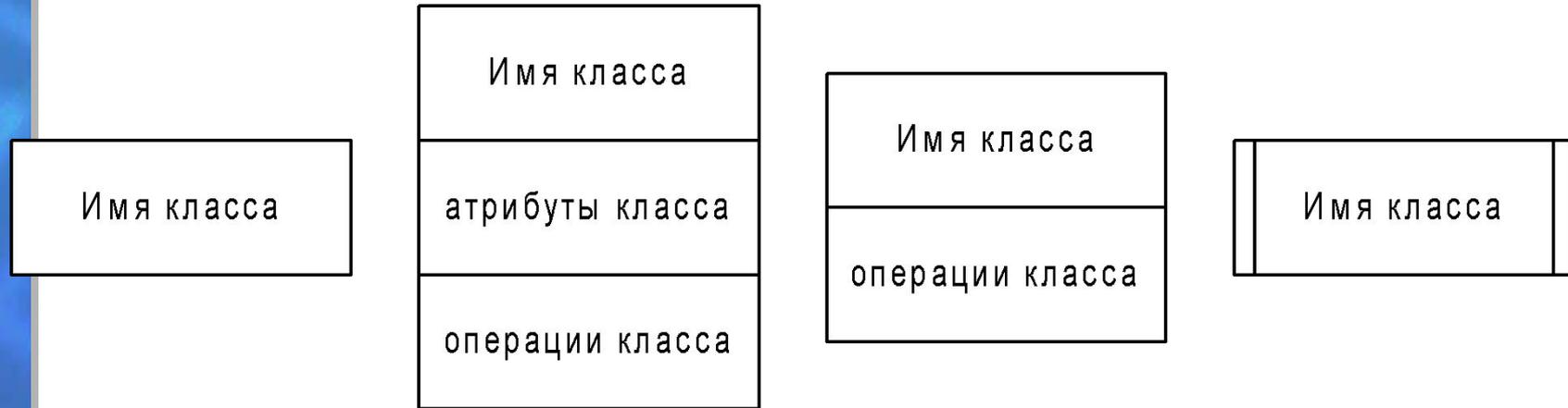
Диаграмма классов — основная логическая модель проектируемой системы

- *Диаграмма классов (class diagram)* — диаграмма, предназначенная для представления модели статической структуры программной системы в терминологии классов объектно-ориентированного программирования
- Диаграмма классов представляет собой граф, вершинами или узлами которого являются элементы типа “классификатор”, которые связаны различными типами структурных отношений
- *Классификатор (classifier)* – специальное понятие, предназначенное для классификации экземпляров, которые имеют общие характеристики

Основные обозначения на диаграмме



Варианты графического изображения класса на диаграмме классов



Атрибут (attribute) класса

- – служит для представления отдельной структурной характеристики или свойства, которое является общим для всех объектов данного класса
- $\langle \text{атрибут} \rangle ::= [\langle \text{видимость} \rangle] ['/'] \langle \text{имя} \rangle [':'] \langle \text{тип атрибута} \rangle [['\langle \text{кратность} \rangle']] ['='] \langle \text{значение по умолчанию} \rangle [['\{ \langle \text{модификатор атрибута} \rangle [',' \langle \text{модификатор атрибута} \rangle]^* \}']]]$
 - Где:
- $\langle \text{видимость} \rangle ::= '+' \mid '-' \mid '#' \mid '~'$.
- *видимость* (visibility) может принимать одно из 4-х возможных значений и отображаться либо посредством специального символа, либо соответствующего ключевого слова

Вид видимости

- + public (общедоступный). Общедоступный элемент является видимым всеми элементами, который имеют доступ к содержимому пространства имен, который им владеет.
- - private (закрытый). Закрытый элемент является видимым только внутри пространства имен, который им владеет.
- # protected (защищенный). Защищенный элемент является видимым для элементов, которые имеют отношение обобщения с пространством имен, который им владеет.
- ~ package (пакет). Элемент, помеченный как имеющий пакетную видимость, является видимым всеми элементами в ближайшем охватывающем пакете в предположении. За пределами ближайшего охватывающего пакета элемент, помеченный как имеющий пакетную видимость, не является видимым.

Элементы записи атрибута

- “/” означает, что атрибут является *производным* (derive). Значение производного атрибута может быть вычислено на основе значений других атрибутов этого или других классов. Поэтому данный атрибут называют иногда *вычислимым*. При использовании производных атрибутов разработчик должен явно указать процедуру или операцию для вычисления их значений.
- <имя> (name) представляет собой строку текста, которая используется в качестве идентификатора соответствующего атрибута и поэтому должна быть уникальной в пределах данного класса. Имя атрибута является единственным обязательным элементом в обозначении атрибута, должно начинаться со срочной (малой) буквы и, как правило, не должно содержать пробелов.

Элементы записи атрибута

- *<тип атрибута>* (attribute type) есть имя классификатора, который является типом данного атрибута. Тип атрибута представляет собой имя некоторого типа данных, определенного или в пакете Типу атрибута должно предшествовать двоеточие
- *<кратность>* (multiplicity) атрибута характеризует общее количество конкретных значений для атрибута, которые могут быть заданы для объектов данного класса
- *<значение по умолчанию>* (default) – некоторое выражение, которое служит для задания начального значения или значений данного атрибута в момент создания отдельного экземпляра соответствующего класса. Конкретное значение по умолчанию должно соответствовать типу данного атрибута. Если этот терм не указан, то значение атрибута на момент создания нового экземпляра класса не определено.

Кратность

- *Кратность (multiplicity)* является спецификацией допустимой мощности множества при инстанцировании соответствующего элемента модели
- Спецификация кратности в нотации БНФ имеет следующий формат:
- $\langle \text{кратность} \rangle ::= \langle \text{диапазон-кратности} \rangle [\{ ' \langle \text{указатель-упорядоченности} \rangle [', ' \langle \text{указатель-уникальности} \rangle] ' \}]$
- $\langle \text{диапазон-кратности} \rangle ::= [\langle \text{нижняя-граница} \rangle '..'] \langle \text{верхняя-граница} \rangle$
- $\langle \text{нижняя-граница} \rangle ::= \langle \text{целое число} \rangle | \langle \text{спецификация значения} \rangle$
- $\langle \text{верхняя-граница} \rangle ::= '*' | \langle \text{спецификация значения} \rangle$
- $\langle \text{указатель-упорядоченности} \rangle ::= 'ordered' | 'unordered'$
- $\langle \text{указатель-уникальности} \rangle ::= 'unique' | 'nonunique'$

Примеры записи атрибутов

- + имяСотрудника : String {readOnly}
- ~ датаРождения : Data {readOnly}
- # /возрастСотрудника : Integer
- + номерТелефона : Integer [1..*] {unique}
- – заработнаяПлата : Currency = 500.00

Операции класса

- *Операция (operation)* класса служит для представления отдельной характеристики поведения, которая является общей для всех объектов данного класса
- Общий формат записи отдельной операции класса следующий (БНФ):
- $\langle \text{операция} \rangle ::= [\langle \text{видимость} \rangle] \langle \text{имя операции} \rangle \text{'('} [\langle \text{список параметров} \rangle] \text{' ')} [\text{'.'} [\langle \text{тип возвращаемого результата} \rangle] \text{'{'}} \langle \text{свойство операции} \rangle [\text{','} \langle \text{свойство операции} \rangle]^* \text{'}'}]$
- Где:
- $\langle \text{видимость} \rangle ::= \text{'+'} \mid \text{'-' } \mid \text{'\#'} \mid \text{'\sim'}$
- $\langle \text{имя операции} \rangle$ (operation name) представляет собой строку текста, которая используется в качестве идентификатора соответствующей операции и поэтому должна быть уникальной для каждой операции данного класса

Формат записи операции класса

- *<список параметров>* (parameter list) представляет собой перечень разделенных запятыми формальных параметров операции и имеет следующий общий формат записи (БНФ):
- *<список параметров> ::= <параметр> [',' <параметр>]**.
- *<параметр> ::= [<направление>] <имя параметра> ':' <выражение типа> ['[' <кратность> ']'] ['=' <значение по умолчанию>] ['{' <свойство параметра> [',' <свойство параметра>]* '}']*
- *<тип возвращаемого результата>* (return type) специфицирует тип значения, возвращаемого данной операцией

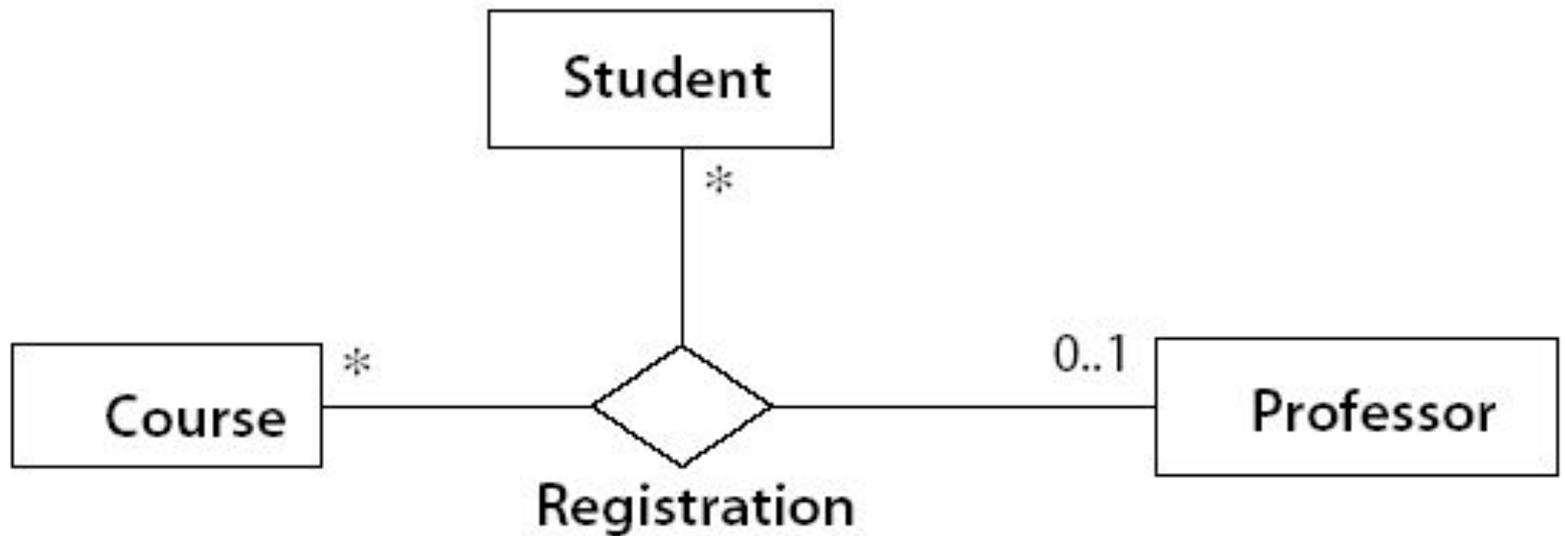
Примеры записи операций:

- +добавить(in номерТелефона : Integer [*] {unique})
- –изменить(in заработнаяПлата : Currency)
- +создать() : Boolean
- toString(return : String)
- toString() : String

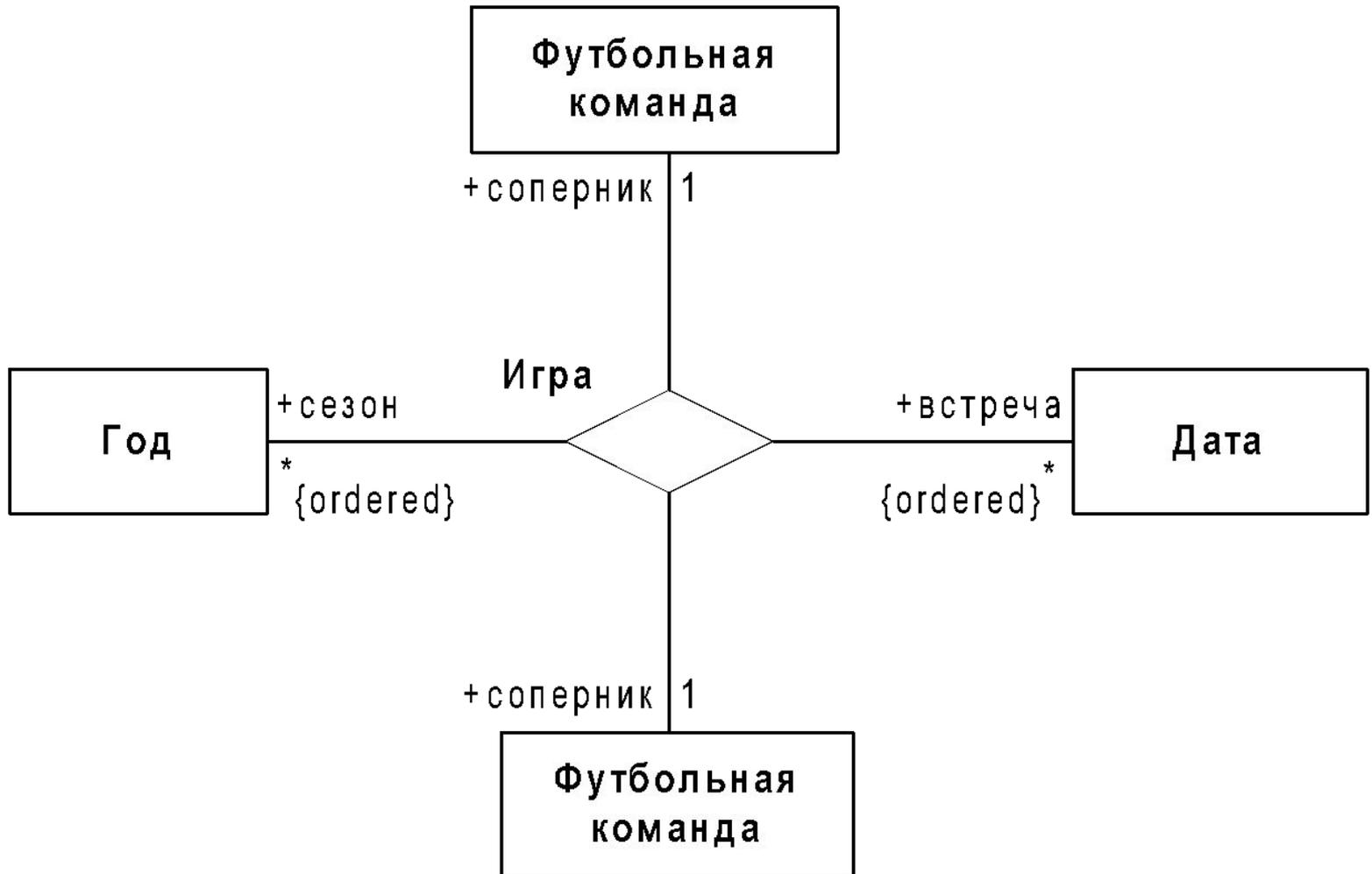
Ассоциация с навигацией и эквивалентное ему представление класса с атрибутом



Пример тернарной ассоциации



Пример 4-арной ассоциации



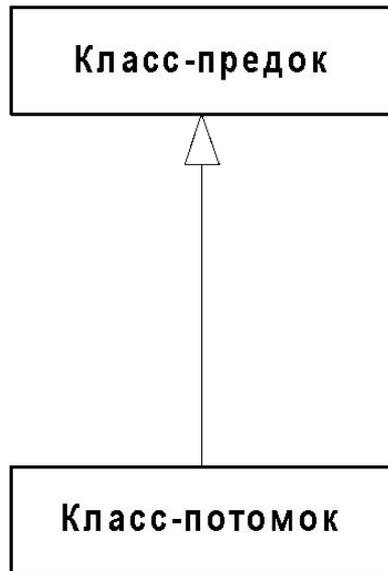
Ассоциация класс (*association class*)

- элемент модели, который имеет свойства как ассоциации, так и класса, и предназначенный для спецификации дополнительных свойств ассоциации в форме атрибутов и, возможно, операций класса.

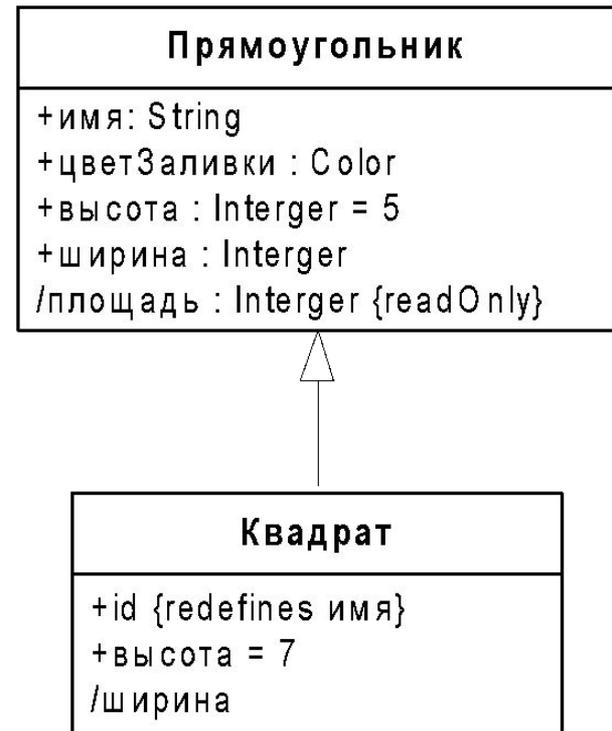


Обобщение (*generalization*)

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

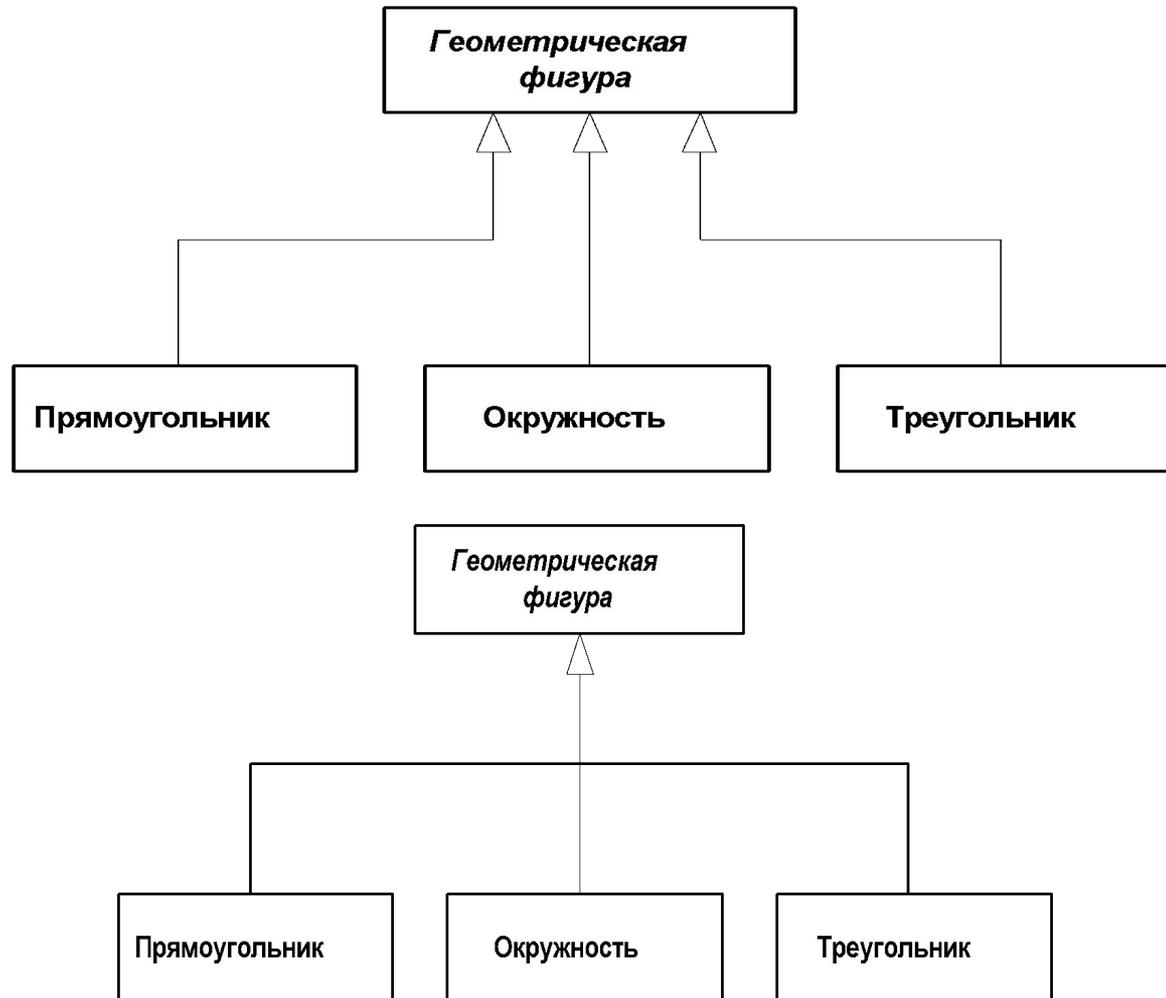


(a)

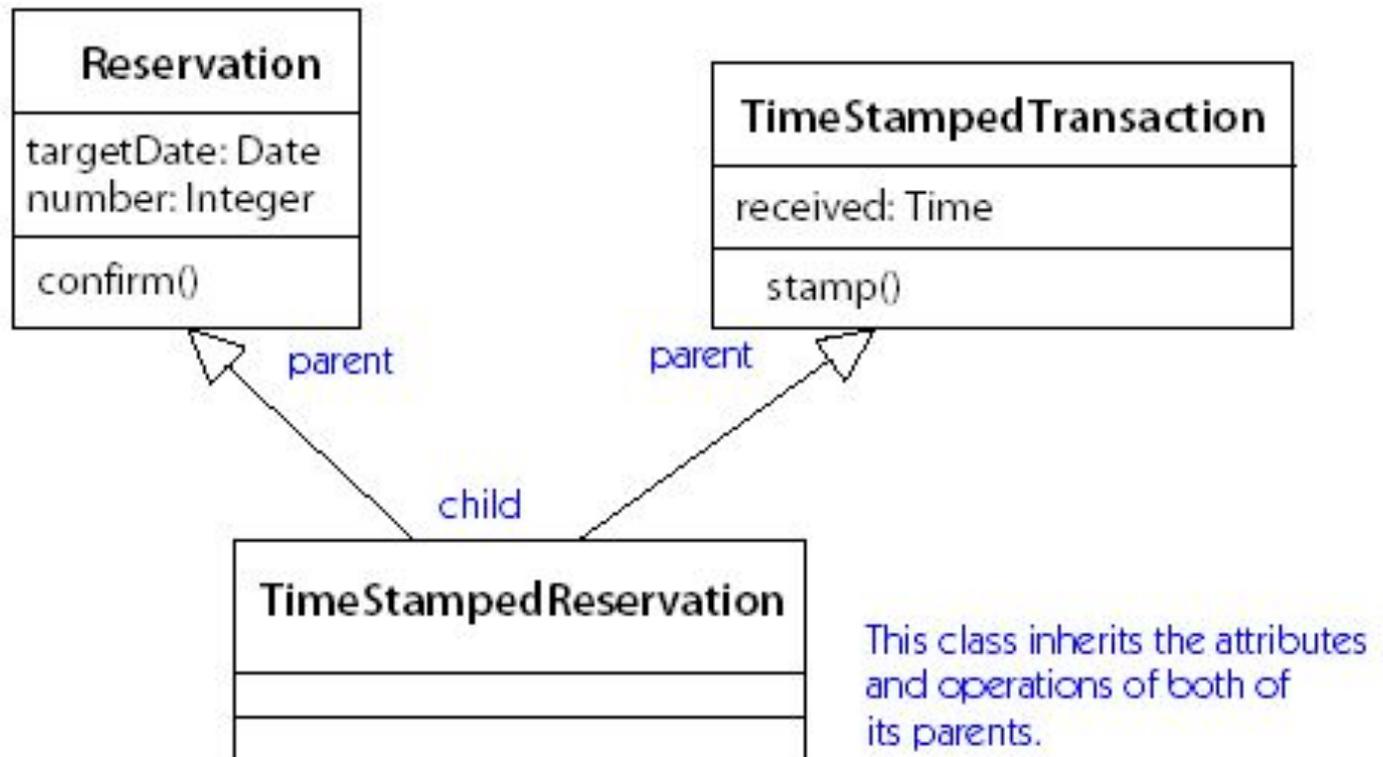


(б)

Примеры отношения обобщения



Множественное наследование – в языке UML разрешено



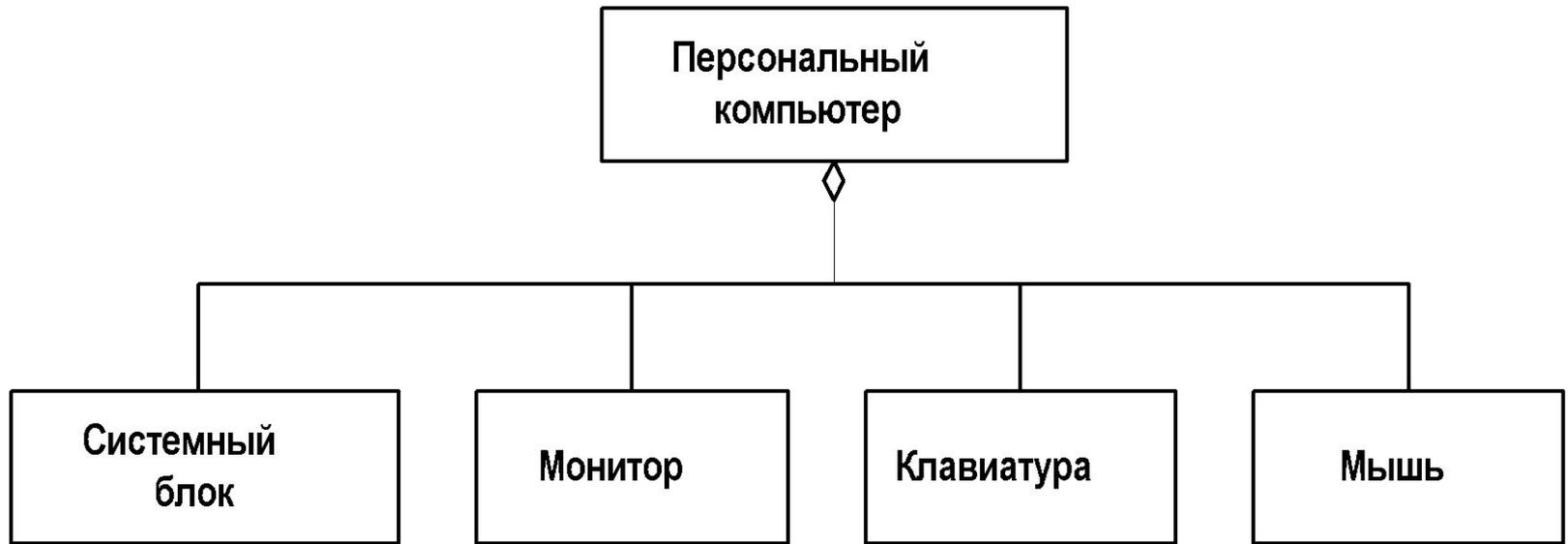
No new features are needed by the child.

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

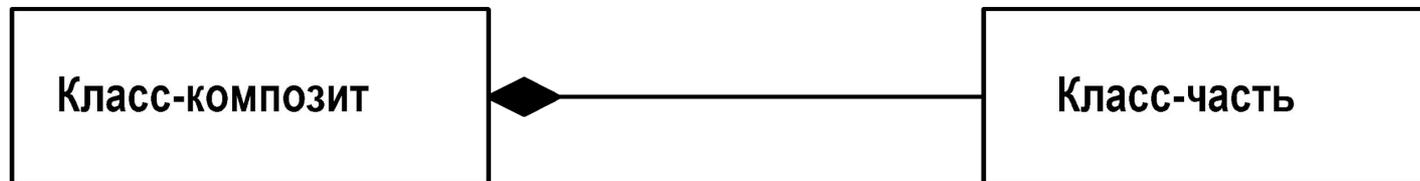


Пример отношения агрегации

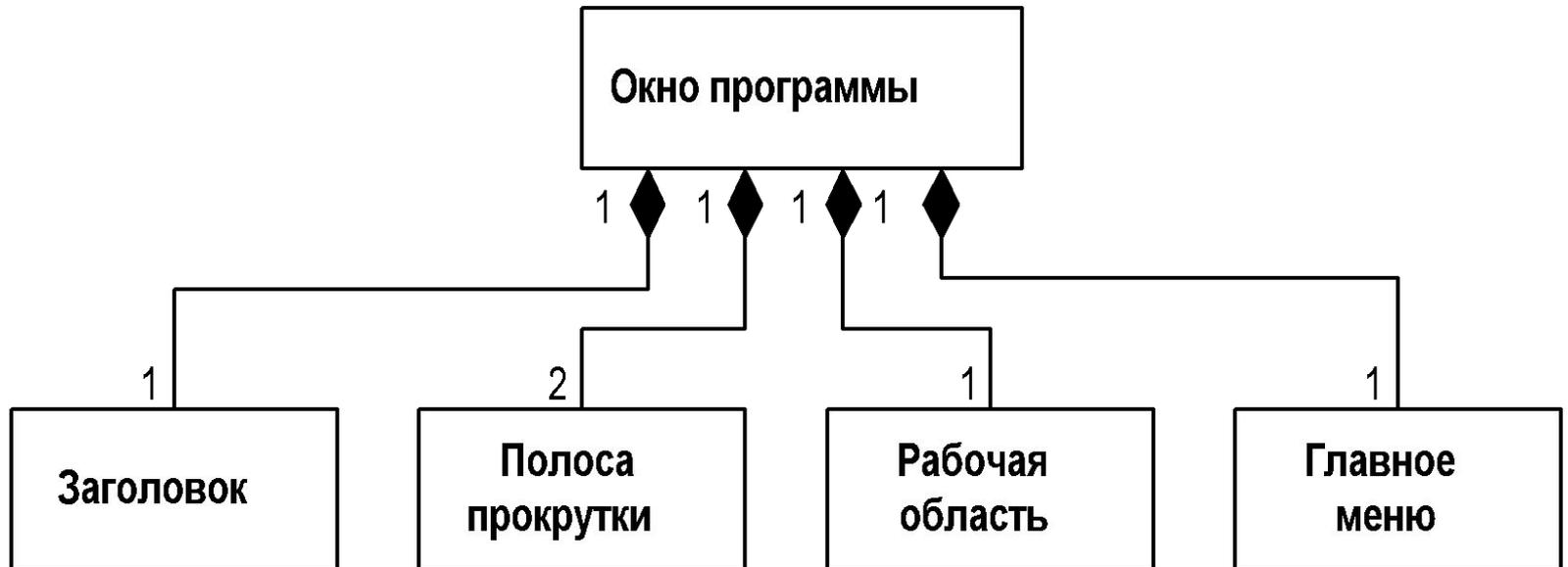


Композиция (*composition*)

- или *композиционная 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



Варианты обозначения композиции

