**Роль полюса ассоциации**

**Роль** (role) — это интерфейс, который предоставляет классификатор в данной ассоциации.

Полюс ассоциации — это точка соприкосновения линии ассоциации с прямоугольником класса. Именно вблизи этой точки располагаются многочисленные дополнения полюсов ассоциации.

**Роль полюса ассоциации** (association end role), или **спецификатором интерфейса**— это способ указать, как именно участвует классификатор (присоединенный к данному полюсу ассоциации) в ассоциации.

Нотация этого дополнения — текст, указанный на полюсе

ассоциации.

В общем случае роль полюса ассоциации имеет следующий синтаксис:

видимость ИМЯ : тип

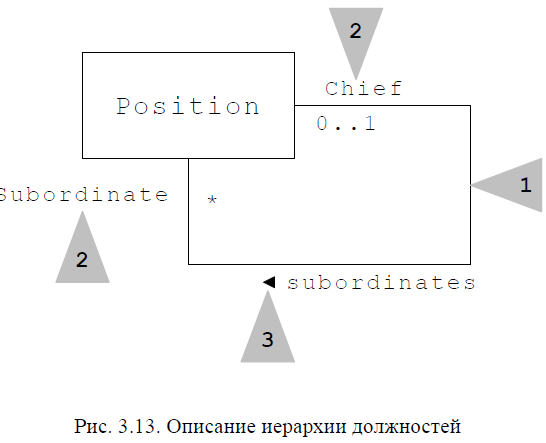
Имя является обязательным, оно называется *именем роли* и фактически является собственным именем полюса ассоциации, позволяющим различать полюса.

Если рассматривается одна ассоциация, соединяющая два различных класса, то в именах ролей нет нужды: полюса ассоциации легко можно различить по именам классов, к которым они присоединены.

Однако, если это не так, т. е. если два класса соединены несколькими ассоциациями, или же если ассоциация соединяет класс с самим собой, то указание ролей полюсов ассоциации является необходимым.

На рисунке изображена ассоциация класса Position с самим собой (1 рис. 3.13). На полюсах ассоциации указаны роли (2 рис. 3.13). Значок, показывающий направление чтения (3 рис. 3.13) позволяет прочесть данную ассоциацию как “Chief subordinates Subordinate”.

Эта ассоциация призвана отразить наличие иерархии подчиненности должностей в организации. Однако из рис. 3.13 видно только, что объекты класса Person образуют некоторую иерархию (каждый объект связан с некоторым количеством нижележащих в иерархии объектов и не более чем с одним вышележащим объектом), но не более того.



Используя роли и, заодно, отношения реализации, можно описать

субординацию в информационной системе отдела кадров достаточно

лаконично и точно.

Например, на рис. 3.14 указано, что в иерархии субординации каждая должность может играть две роли.

С одной стороны, должность может рассматриваться как начальственная (1) (chief), и в этом случае она предоставляет интерфейс IChief (2) имеющий операцию petition (начальнику можно подать служебную записку). С другой стороны, должность может рассматриваться как подчиненная (3) (subordinate), и в этом случае

она предоставляет интерфейс ISubordinate (4), имеющий операцию

report (от подчиненного можно потребовать отчет). У начальника

может быть произвольное количество подчиненных (5), у подчиненного может быть не более одного начальника (6).

