**Семь наиболее важных свойств классификаторов, которые понадобятся, прежде всего:**

**1. Классификаторы имеют имена**. Имя служит для идентификации элемента модели, поэтому имя должно быть уникально в данном пространстве имен.

**2. Классификатор может иметь экземпляры.** Экземпляры бывают прямые и косвенные.

Если некоторый объект непосредственно порожден с помощью конструктора классификатора А, то этот объект называется

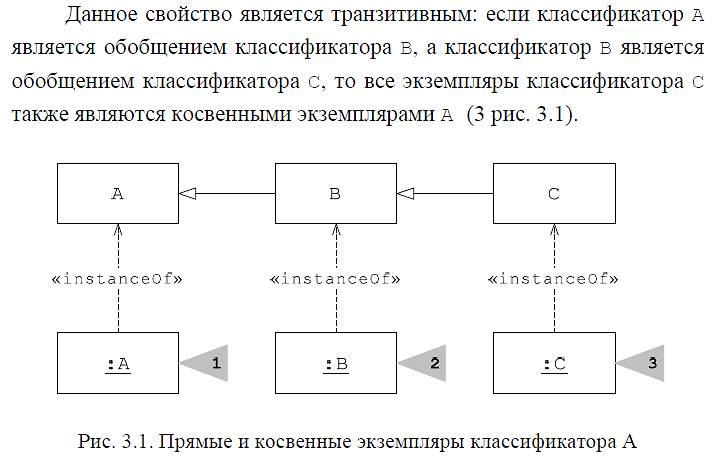
**прямым экземпляром** (direct instance) классификатора А (1 рис. 3.1). **Употребляют также термин "непосредственный экземпляр".**

Если классификатор А является обобщением классификатора В

или, что то же самое, классификатор В является ***специализацией***

классификатора А, то все экземпляры классификатора В являются

**косвенными экземплярами** классификатора А (2 рис. 3.1).



**3. Классификатор может быть абстрактным или конкретным.** Абстрактный (abstract) классификатор не может иметь прямых экземпляров и в этом случае *его имя выделяется курсивом****.***

**Из данного правила есть исключение**, если классификатор имеет стереотип «interface», то его имя не нужно выделять курсивом.

Конкретный (concrete) классификаторможет иметь прямые экземпляры и в этом случае его имя записывается прямым шрифтом.

**4. Классификатор имеет видимость.**

Видимость (visibility) определяет, может ли составляющая одного классификатора (в том числе имя) использоваться в другом классификаторе.

Другими словами, если в определенном контексте нечто доступно и может быть как-то использовано, то оно является видимым в этом контексте. Если же оно не видимо, то и не может быть использовано. Видимость является свойством всех элементов модели (хотя не для всех элементов это свойство является существенным).

**Видимость может принимать одно из четырех**

**значений:**

- открытый (обозначается знаком + или ключевым словом

public);

- защищенный (обозначается знаком # или ключевым словом

protected);

- закрытый (обозначается знаком – или ключевым словом

private).

- пакетный (обозначается знаком ~ или ключевым словом

package).

Элемент модели со значением видимости пакетный, виден элементам только того пакета, в котором он сам определен.

Остальные значения видимости работают также как в ООП.

**5. Все составляющие классификатора имеют область действия.**

Область действия (scope) определяет, как проявляет себя составляющая классификатора в экземплярах, т. е. имеют экземпляры свои значения составляющей (обычная структура данных) или совместно используют одно значение (static структура данных/конструктор/деструктор)

**Область действия имеет два возможных значения:**

- экземпляр (instance) — никак специально не обозначается, поскольку подразумевается по умолчанию;

- классификатор (classifier) — описание составляющей классификатора подчеркивается.

Если областью действия составляющей является экземпляр, то каждый экземпляр классификатора имеет свое значение составляющей

(обычная структура данных).

Например, областью действия атрибута по умолчанию является экземпляр. Это означает, что каждый объект имеет свое собственное значение атрибута, которое может меняться независимо от значений данного атрибута других объектов, экземпляров этого же класса.

Если областью действия составляющей является классификатор, то все экземпляры классификатора совместно используют одно значение составляющей (static структура данных/конструктор/деструктор).

Например, конструктор обычно имеет область действия классификатора (класс), поскольку является процедурой, общей для всех экземпляров данного класса.

**6. Классификатор имеет кратность**, т. е. ограничение на количество экземпляров классификатора, как множества. Кратность может быть не только у классификаторов, но также у атрибутов и полюсов ассоциаций (см. параграф 3.2.2).

Не следует путать кратность с количеством экземпляров. Множество, указанное в модели, во время выполнения может иметь различное количество элементов, и количество элементов может динамически меняться. Кратность определяет пределы изменения количества элементов.

Кратность (multiplicity) множества *— это множество чисел,*

*которые задают все допустимые значения мощности для данного*

*множества.*

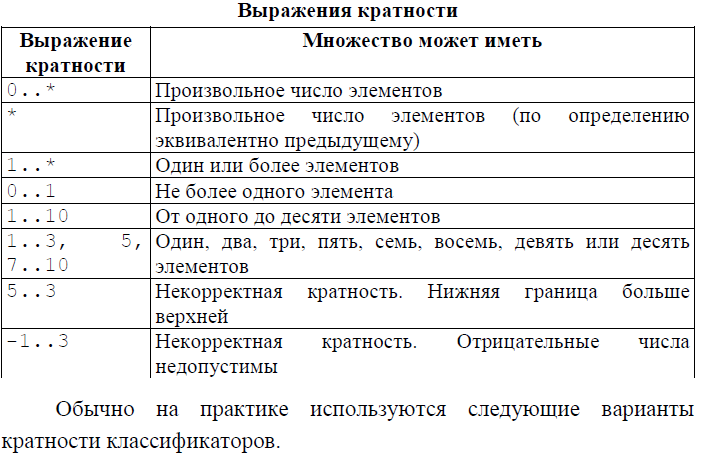
Синтаксически кратность задается выражением, которое является непустой последовательностью элементов (разделенных запятыми), каждый из которых имеет следующий формат:

Нижняя граница .. ВЕРХНЯЯ ГРАНИЦА

В качестве верхней и нижней границы используются натуральные числа (ноль тоже).

Кроме того, в качестве верхней границы может использоваться символ \*.

Если нижняя граница не задана, то она опускается вместе с символом .. (две точки).



Как видно из таблицы выражения кратности записываются почти также, как множества в алгебре (значения указанные в выражении кратности как бы записаны в [], то есть включены в множество.

Для записи отсутствующих (выколотых) значений используется пробел (смотри 6 столбец).

В таблице также даны неправильные записи выражения кратности (последние два столбца).

**7.** **Только классификаторы могут участвовать в отношении обобщения.**