类与对象

- 定义与表示
- 类名
- 属性
- 操作
- 操作的多态性

6.2.1 结构化地表达客观事物的术语

6.2.1.1 类与对象

——体现数据抽象

(1) 定义与表示

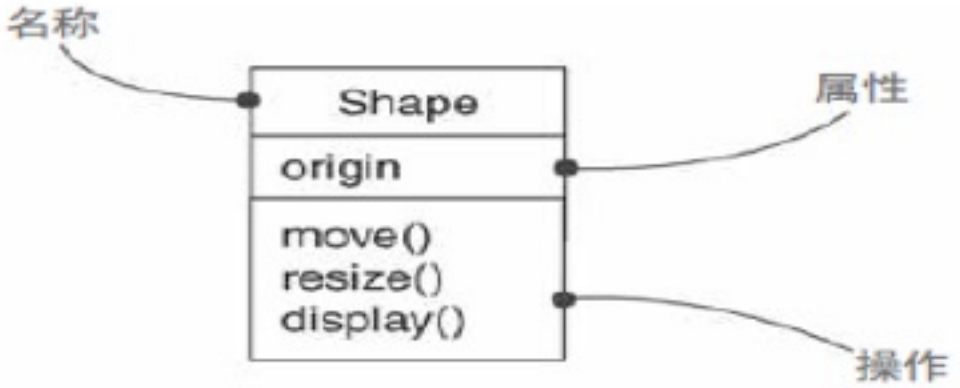
类(Class)：是一组具有相同属性、操作、关系和语义的对象的描述。

对象(object)：对象是类的一个实例。

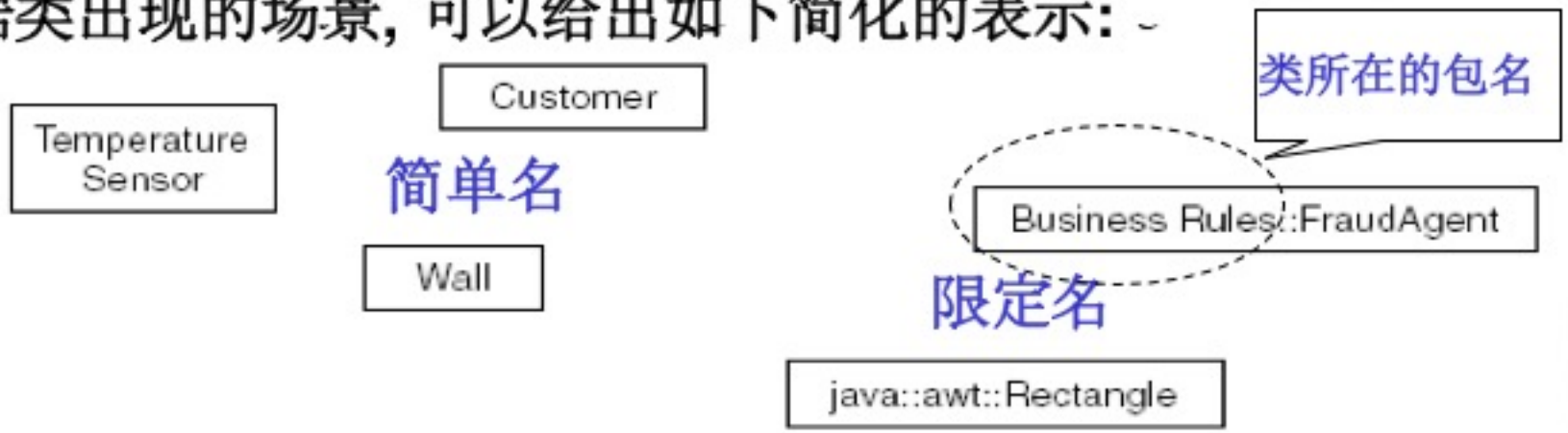
类与对象

- 定义与表示
- 类名
- 属性
- 操作
- 操作的多态性

类表示为具有三个栏目的矩形，如下所示：



依据类出现的场景，可以给出如下简化的表示：

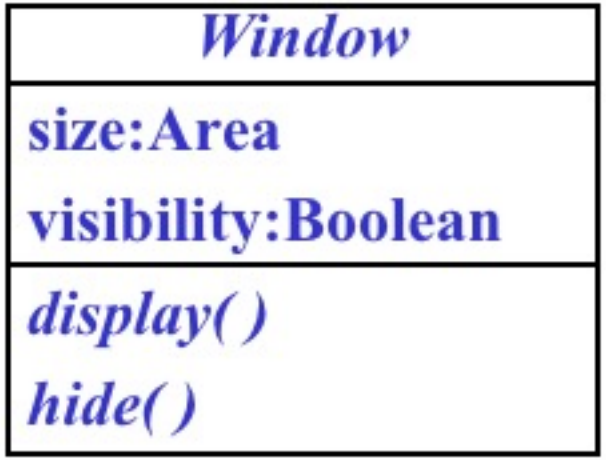


类与对象

- 定义与表示
- 类名
- 属性
- 操作
- 操作的多态性

类可以是抽象类，即没有实例的类，此时类名采用斜体字.

例如：



类与对象

- 定义与表示
- 类名
- 属性
- 操作
- 操作的多态性

(2)类名(类的标识)

❶类名使用黑体字，第一个字母通常要大写，并位于第一栏的中央.

❷类名往往是从正被建模系统的词汇表中提取的简单名词或名词短语.

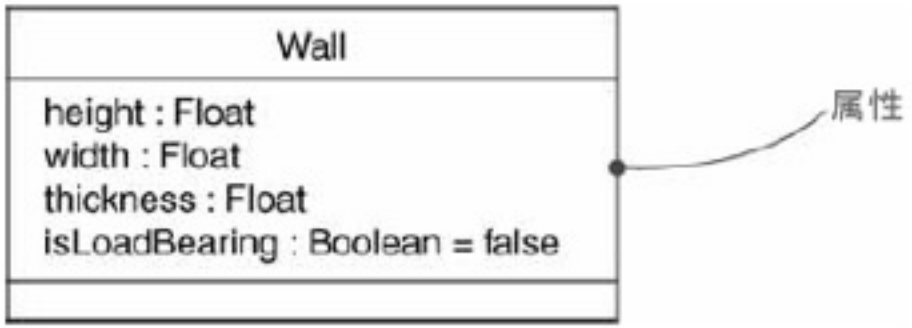
类与对象

- 定义与表示
- 类名
- 属性
- 操作
- 操作的多态性

(3)属性（attribute）

属性是类的一个命名特性，由该类的所有对象所共享，用于表达对象状态的数据。

表示：



❶ 一个属性往往具有所属的类型，用于描述该特性的实例可以取值的范围。

❷ 类的一个对象对每一个属性应有特定的值。

❸ 一个类可以有多个属性，也可以没有属性。

类与对象

- 定义与表示
- 类名
- 属性
- 操作
- 操作的多态性

属性的作用范围：

❶ 实例范围的属性：一个类的所有对象具有相同的属性即属性的个数、名称、数据类型相同，但属性值可不同，并随程序的执行而变化.

❷类范围的属性：描述类的所有对象共同特征的一个数据项，对于任何对象实例，它的属性值都是相同的，通常对属性加下划线来表示该属性为实例范围的属性.

注：如C++中冠以static的成员变量和smalltalk中的class attribute都是类属性。

类与对象

- 定义与表示
- 类名
- 属性
- 操作
- 操作的多态性

定义属性的格式为：

[可见性]属性名[:类型][多重性][=初始值][{特性串}]

注:加了方括号的内容是可选的.

❶可见性

表明该属性是否可以被其它类所使用。

其可见性的值可以为：

+ 公有的：可供其它类使用之；

受保护的：其子类可以使用之；

- 私有的：只有本类的操作才能使用之；

~ 包内的：只有在同一包中声名的类才能使用之。

也可以使用关键字 **public**、**protected**、**private**和**package**，分别表示公有的、受保护的、私有的和包内的。

类与对象

- 定义与表示
- 类名
- 属性
- 操作
- 操作的多态性

❷属性名

属性名是一个表示属性名字的标识串。通常以小写字母开头，左对齐。

❸类型

类型是对属性实现类型的规约，与具体实现语言有关。

例如：name:String

其中，“name”是属性名，而“String”是该属性的类型。

❹多重性

多重性用于表达属性值的数目。即该类实例的这一特性可以具有的值的范围。

例如：points[2..*]:Point

多重性是可以省略的。在这种情况下，多重性是1..1。即属性只含一个值。如果多重性是0..1，就有可能出现空值。

例如，name[0..1]:String

这样的声明就允许属性“name”为空值或空串。

类与对象

- 定义与表示
- 类名
- 属性
- 操作
- 操作的多态性

❺初始值

初始值是与语言相关的表达式，用于为新建立的对象赋予初始值。例如：origin:Point=(0,0)

初始值是可选的。如果不声明对象这一属性的初始值，那么就要省略语法中的等号。对象的构造函数可以参数化或修改默认的初始值。

❻性质串

如果说“类型”、“多重性”以及“初始值”都是围绕一个属性的可取值而给出的，那么“性质串”是为了表达该属性所具有的性质而给出的。例如：

a:integer=1{frozen}

其中，“frozen”是一个性质串，表示属性是不可以改变的。如果没有对一个属性给出这一性质串，那么就认为该属性是可以改变的。

类与对象

- 定义与表示
- 类名
- 属性
- 操作
- 操作的多态性

属性的声明举例：

origin	只有属性名
＋origin	可见性和属性名
origin：Point	属性名和类型
name ：String[0..1]	属性名、类型和多重性
origin ： Point=(0,0)	属性名，类型和初始值
id: Integer{readonly}	属性名，类型和性质串

类与对象

- 定义与表示
- 类名
- 属性
- 操作
- 操作的多态性

(4)操作（operation）

操作是对一个类中所有对象要做的事情的抽象.

表示：

❶一个类可以有多个操作，也可以没有操作。

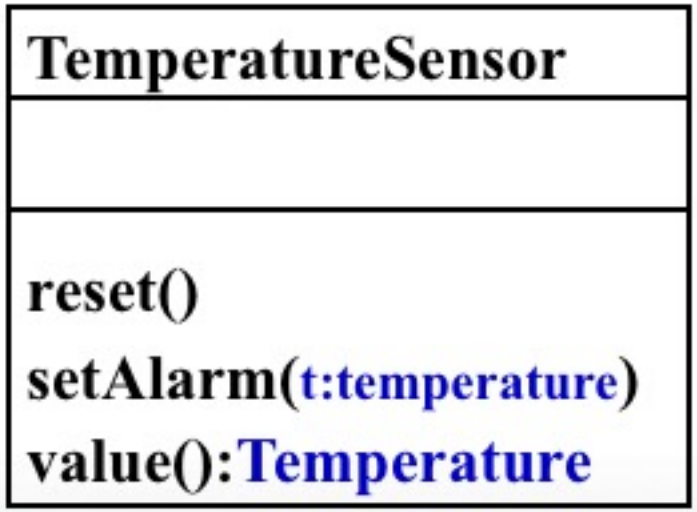
❷操作名除第一个词之外，其他每个词的第一个字母要大写

类与对象

- 定义与表示
- 类名
- 属性
- 操作
- 操作的多态性

❸操作名往往是描述其所在类的行为的动词或动词短语。

❹可以通过给出操作的特征标记进一步描述之，特征标记通常包括参数名、类型和默认值；如果该操作是一个函数，那么其特征标记还包括返回类型。如下所示：



类与对象

- 定义与表示
- 类名
- 属性
- 操作
- 操作的多态性

❺操作可以是抽象操作，即没有给出实现的操作。此时的操作名采用斜体。例如：

Window
size:Area visibility:Boolean
display( ) *hide( )*

注：抽象操作映射到C++称为纯虚操作

❻调用一个对象上的操作可能会改变该对象的数据或状态。

类与对象

- 定义与表示
- 类名
- 属性
- 操作
- 操作的多态性

操作的作用范围：

类范围的操作和实例范围的操作的唯一区别是：
类范围的操作名和类型表达式要加下划线

类范围的操作和实例范围的操作的表示

类与对象

- 定义与表示
- 类名
- 属性
- 操作
- 操作的多态性

表达操作的完整语法格式为：

[可见性] 操作名[（参数表）][:返回类型][{性质串}]

其中：

❶可见性　如同属性的可见性一样，其值可以为：

+ 公有的	可供其它类访问之；
# 受保护的	其子类能访问之；
- 私有的	只有本类的操作才能访问之；
~ 包内的	只有在同一包中声名的类才能访问之。

❷操作名

- 操作名一般是一动词或动词短语，通常以小写字母开头，左对齐；
- 如果操作名是动词短语，除第一个词外，其余每个词的第一个字母为大写，例如isEmpty()；
- 若操作是一个抽象操作，则以斜体字表示之。

类与对象

- 定义与表示
- 类名
- 属性
- 操作
- 操作的多态性

❸参数表

给出该操作的参数。一个操作可以有参数表，也可以没有。如果有参数表的话，其语法为：

［方向］参数名：类型［=默认值］

- 方向是对输入/输出的规约，其取值可以为：
  - in　输入参数，不能修改之
  - inout　输入参数，为了与调用者进行信息通讯，可能要对之进行修改.
  - out　输出参数，为了与调用者进行信息通讯，可能要对之进行修改.
- 类型是实现类型的（与语言有关）规约；
- 默认值是一个值表达式，用最终的目标语言表示。该项是可选的。

注释:out或inout参数等价于返回参数和in参数。提供out和inout参数是为了与较老的编程语言相兼容. 可用显式的返回参数来代替.

❹返回类型

返回类型是对操作的实现类型或操作的返回值类型的规约，它与具体的实现语言有关。

- 如果操作没有返回值（例如C++ 中的void），就省略冒号和返回类型。
- 当需要表示多个返回值时，可以使用表达式列表。
- 根据实际问题的需要，可以省略全部的参数表和返回类型，但不能只省略其中的一部分。

类与对象

- 定义与表示
- 类名
- 属性
- 操作
- 操作的多态性

(5)操作的多态性