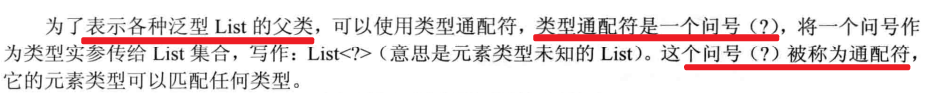
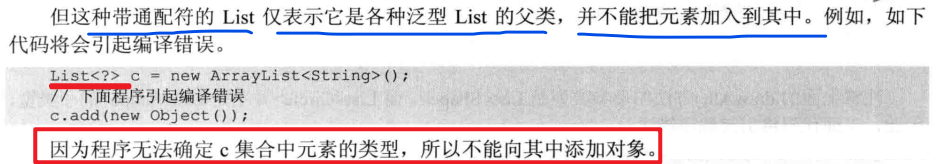
## 泛型类定义

**泛型 既参数化类型 也可叫 类型参数 或 类型实参。**

## 通配符-形参



### 通配符的局限



**<T>** 表示一个具体类型的泛型，**[只能在定义类时指定]**。（T表示一个具体的类型可以是任意大写字母。）

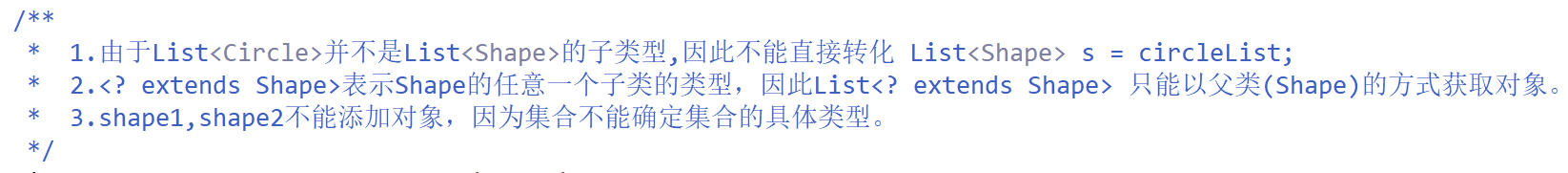
**<T extends Shape>**表示Shape的任意一个子类的类型，**[只能在定义类时指定]**。此种指定方式指定一个范围内的一个具体类型**。**

**<?>** 表示的是任意一个泛型的父类，因此就无法确定具体的类型。不能添加数据，只能以父类（Object）的方式获取数据。[**只能在引用类型参数中使用，**]

## 泛型范围-上下限

**设定上限-表示确定了父类**

**<? extends Shape>** 表示Shape的任意一个子类泛型的父类，因此List<? extends Shape >只能以父类(Shape)的方式获取数据。[**只能在引用类型参数中使用**]



**设定下限-表示确定了子类**

<? supper Shape> 用法同上。

### 小结

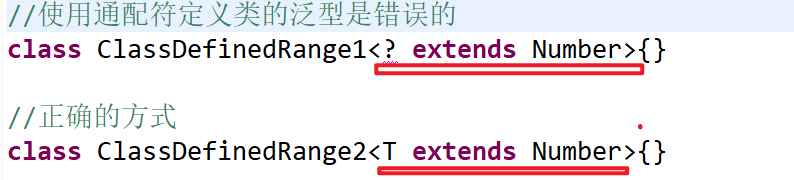
1.T表示一个具体的类型。?表示多个类型(也就是任意一个类型)，是T的父类。

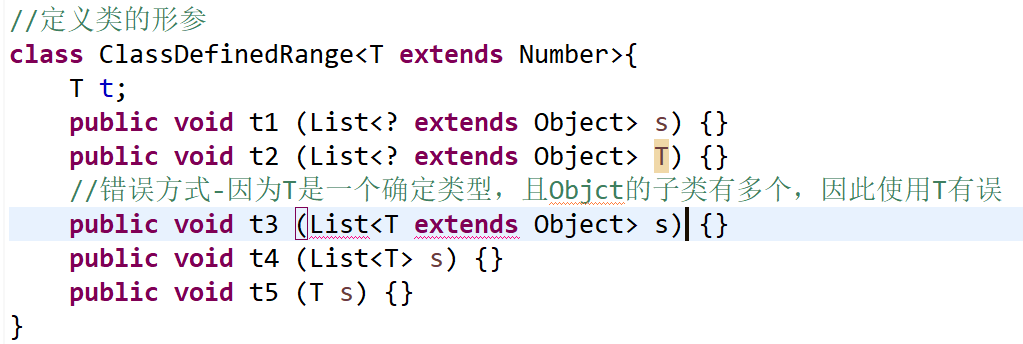
2.泛型中通配符(?)表示的是数据范围，因此不能在泛型类的定义中。【**因为泛型在类中定义需要指定确定的类型信息**】。只能用于形参中。

形参定义->是指在定义函数时使用的参数.即方法的参数都是形参。

3.而且通配符定义的类型只能查询数据，不能添加数据（**通配符指的就是 ?**）。

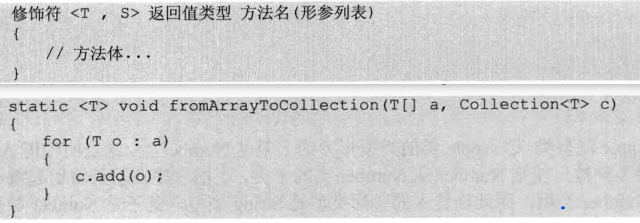
错误案例





## 泛型方法

### 定义



泛型方法定义： 返回值类型前边

泛型类定义： 类名后边 public class A<T>{}

