

## task7. Generic

### 为什么要使用泛型程序设计：

泛型程序设计 (Generic programming) 意味着编写的代码可以被很多不同类型的对象所重用。例如，我们并不希望为聚集 `String` 和 `File` 对象分别设计不同的类。实际上，也不需要这样做，因为一个 `ArrayList` 类可以聚集任何类型的对象。这是一个泛型程序设计的实例。

类型参数的魅力在于：使得程序具有更好的可读性和安全性。

#### 1. 泛型：

泛型，即“参数化类型”。一提到参数，最熟悉的就定义方法时有形参，然后调用此方法时传递实参。那么**参数化类型**怎么理解呢？顾名思义，就是将类型由原来的具体的类型参数化，类似于方法中的变量参数，此时类型也定义成参数形式（可以称之为类型形参），然后在使用/调用时传入具体的类型（类型实参）。

泛型的**本质**是为了参数化类型（在不创建新的类型的情况下，通过泛型指定的不同类型来控制形参具体限制的类型）。也就是说在泛型使用过程中，操作的数据类型被指定为一个参数，这种参数类型可以用在类、接口和方法中，分别被称为泛型类、泛型接口、泛型方法。

泛型使得代码可以被不同类型的对象重用。

#### 2. 泛型类

一个泛型类就是拥有一个或者多个类型变量的类。需要注意的是，泛型类只能用在成员变量上，并且只能使用引用类型。

#### 3. 泛型方法

下面是定义泛型方法的规则：

- 所有泛型方法声明都有一个类型参数声明部分（由尖括号分隔），该类型参数声明部分在方法返回类型之前
- 每一个类型参数声明部分包含一个或多个类型参数，参数间用逗号隔开。一个泛型参数，也被称为一个类型变量，是用于指定一个泛型类型名称的标识符。
- 类型参数能被用来声明返回值类型，并且能作为泛型方法得到的实际参数类型的占位符。
- 泛型方法体的声明和其他方法一样。注意类型参数只能代表引用型类型，不能是原始类型（像 `int`, `double`, `char` 的等）。

#### 4. 类型通配符：

类型通配符一般是在不确定具体参数时候，使用 `?` 代替。`?` 表示未知类型，类型参数赋予不确定值，只能用在声明类型、方法参数上，不能用在定义泛型类上