**JAVA泛型**

**出现的原因：**

为了编写通用性代码，当要使用的数据类型不确定时，在没有泛型机制的时，一般把数据类型声明为Object类型，因为Object类型是所有类型的父类。但是这样会出现类型转换的问题，比如声明一个Object类型的数组，那么它就既可以放String类型，double,自己定义的类，当提取出来使用的时候，就需要强制类型转换，很可能就会发生错误，发生错误是运行时错误，不方面调试。

在使用泛型后，编译时期就可以保证类型安全，消除强制类型转换

**使用场景：**

在编写通用性代码时，当要使用的数据类型不确定，这个时候就可以使用泛型，比如容器类

**定义：**

泛型的实质就是类型参数化，即把类型当成参数传递。

泛型通配符：指定类型的参数范围

**分类：**

泛型类

使用场景：容器类

泛型接口

使用场景：策略模式的公共策略（Comparator）

泛型方法