

一、泛型的设计背景

二、泛型的概念

三、泛型的引入解决的问题

一、泛型的设计背景

集合容器类在设计阶段声明阶段不能确定这个容器到底实际存的是什么类型的对象，所以在 JDK 1 5 之前只能把元素类型设计为 Object，JDK 1 5 之后使用泛型来解决。因为这个时候除了元素的类型不确定，其他的部分是确定的，例如关于这个元素如何保存，如何管理等是确定的。因此此时，**把元素的类型设计成一个参数，这个类型参数叫做泛型**。Collection<E> List<E> ArrayList<E> 这个 <E> 就是类型参数,即泛型。

二、泛型的概念

1.所谓泛型，就是允许在定义类、接口时通过一个标识通过一个标识表示类中某个属性的类型或者是某个方法的返回值及参数类型。这个类型参数将在使用时（例如，继承或实现这个接口，用这个类型声明变量、创建对象时）确定（即传入实际的类型参数，也成为类型实参）

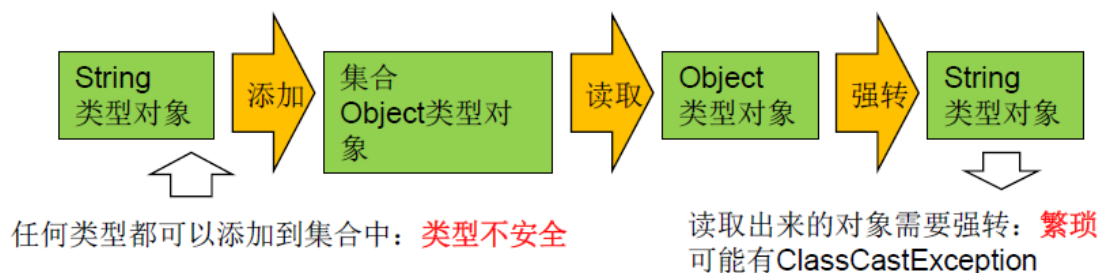
2.jdk 5.0新增的特性

三、泛型的引入解决的问题

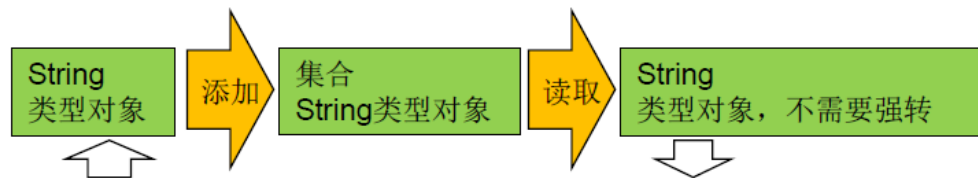
1.解决元素存储的安全性问题

2.解决获取数据元素时 需要类型强制转换的问题

在集合中没有泛型时



在集合中有泛型时



只有指定类型才可以添加到集合中：类型安全 读取出来的对象不需要强转：便捷

Java泛型可以保证如果程序在编译时没有发出警告，运行时就不会产生 `ClassCastException` 异常。同时，代码更加简洁、健壮。

