**泛型的概念:** 广泛的类型.定义一个类型,类中方法参数或者返回值,不能确定类型,那么可以泛型表示

1. **泛型使用的注意事项**
2. 泛型的前后,需要保持一致
3. 泛型推断 : 前面泛型定义好之后,后面泛型不写,只写一对<>,后面就默认与前面泛型保持一致,从JDK1.7版本开始
4. 可以理解为前面写了后面不用写，光后面写了前面不写是没用的

**带有泛型的方法**

定义方法时,方法上带有泛型

定义格式:

修饰符<泛型类型1,泛型类型2...> 返回值类型 方法名(参数列表){

return 数值;

}

说明:

1. 如果方法不是静态的,那么方法上面可以自己定义泛型,也可以使用类上的泛型
2. 静态方法,如果要使用泛型,只能在方法上,自己定义泛型

原因 : 静态属于类,静态优先于对象存在的,静态方法不能使用类的泛型, 类上的泛型,只有创建类对象时,才能确定具体类型

1. 方法上定义的泛型,只能在方法中使用

带有泛型的接口

定义一个接口,接口上带有泛型

定义格式:

修饰符 interface 接口名<泛型类型1,泛型类型2...>{

// 抽象方法

}

说明:

带有泛型的接口,实现类情况

1. 实现类不带有泛型,要求实现类在实现接口时,接口上的泛型需要制定成具体类型

class MyInterfaceImpl1 implements MyInterface<String>

1. 实现类带有泛型,那么实现的接口就不需要给出具体的泛型类型

class MyInterfaceImpl2<T> implements MyInterface<T>

1. 接口上的泛型,可以在整个接口中,当做已知类型使用