**泛型（Generics）：**

1. 泛型指的是在是声明类的时候不指定类中属性的具体类型，而是由在创建该类对象 和实例化该类对象时指定具体的类型。
2. 泛型的原理是在声明类的时候通过标识表示类中某一属性的类型或者某个方法的返 回值及参数类型。在声明对象和实例化对象时，指定其标识所代表的的类型。
3. 指定泛型标识时，只能指定为类，不能指定为基本数据类型。

**泛型的安全警告：**

在声明和实例化泛型类对象时，如果不指定泛型的类型，则在编译时会出现安全警告。 但是并不影响程序的运行，因为所有的类型同一使用了Object类型进行接收。

**泛型通配符：**

1）涉及到泛型对象的引用传递时。将实例化了泛型对象作为调用方法时的实参传递给 一个方法，此时接收该泛型对象的方法必须使用相同泛型类型的对象进行接收。否 则编译出现错误。除非该泛型对象没有指定泛型；

2）但这样做有些不妥，所以在Java中引入了泛型通配符“？”指定泛型的类型，表示 可以接收任意泛型类型的对象；

3）但时需要注意的是，如果使用“？”接收泛型对象，则不能设置被该对象指定的内 容，如属性。

也就是说，如果涉及到了泛型对象的引用传递，则接收时要么使用相同泛型类型的对象 接收，要么使用泛型通配符接收。使用相同泛型类型的对象接收其对象的内容可以修改， 而使用泛型通配符进行接收其对象的内容不允许被修改。

**受限泛型：**

受限泛型指的是设置泛型对象在声明泛型类型时的上限或下限。

1. 上限：上限使用extends关键字声明，表示泛型的类型只能是所指定的类型或者是 此类型的子类。

**·声明类：类名称<泛型标识 extends 类>{ }**

**·声明对象：类名称<? extends 类> 对象名称**

1. 下限：下限使用super关键字声明，表示泛型的类型只能是所指定的类型或者是此 类型的父类。

**·声明对象：类名称<? super类> 对象名称**

**泛型在子类继承中的限制：**

在泛型操作中，子类的泛型类型无法使用父类泛型类型进行接收。例如存在泛型类Info，

则“Info<Object> i=new Info<String>( )”是错误的。

**定义泛型接口：**

在声明接口时也可以定义泛型。其定义格式为：

**interface 接口名称<泛型标识>{ }**

**子类实现泛型接口的两种方式：** 定义泛型接口后，子类可以通过两种方式实现泛型接口。

1. 子类也定义了与泛型接口相同标识的泛型。(该类也将变成了泛型类)
2. 子类在实现此接口时指定具体类型。(该类不会变成泛型类)

也可以将其理解为泛型类实现接口和普通实现接口两种。

**定义泛型方法：** 泛型方法的定义与其所在的类是否是泛型类没有任何关系。其定义格式为：  
 **访问权限 <泛型标识> 泛型标识 方法名称（[泛型标识] 参数名称）{ }**

以方法public <T> T fun(T t)为例，<T> 表示该方法参数只能接受T标识的类型的参数，T 表示返回类型，如果此时传入一个字符串，则返回字符串。

**通过泛型方法返回泛型类实例：**

通过泛型方法返回一个泛型类的实例，则必须在方法的返回类型声明处明确地指定泛型 标识和泛型类。如public static <T> Info<T> fun(T gram)，此方法将返回一个Info泛型类 实例。

**使用泛型同一传入的参数：**

如果有一个方法要求传入的泛型对象类型是一致的，也可以通过泛型方法指定。

**泛型数组：**

使用泛型方法时，也可以接收和返回一个泛型数组。

**泛型的嵌套：**

在一个泛型类对象中可以指定另个泛型类类型。