**泛型基本概念：**

泛型（Generics）是指创建一个类时不指定类中数据的类型，而是在外部声明声明及实 例化对象时指定其数据类型。

**泛型的与原理：**

泛型的原理是在声明类时通过一个或多个标识符来表示类中属性、方法返回值和参数的 类型。然后在声明该类或实例化对象时，指定好具体的数据类型。

**指定泛型时的规则：**

1）指定泛型的类型时，是无法指定基本数据类型的，必须设置成一个类，所以指定为 基本数据类型时需要指定基本数据类型所对应的包装类。

2）在泛型类中，如果覆写了toString( )方法，则返回时必须得调用toString( )方法将其 变为字符型返回。

**泛型的安全警告：**

在泛型应用中，声明泛型类对象时必须指定好具体的数据类型。如果没有指定，则在编 译时会出现安全警告；但是并不是影响程序的运行，因为如果没有指定泛型，则会默认 为Object类型。

**泛型通配符：**

1. 在开发中，如果涉及到了泛型类对象的引用传递，则接收该泛型类对象的泛型类对 象必须与原泛型类对象所指定的泛型类型一致（不管是否转型），否则编译会报错。
2. 但在Java中引入了通配符“？”用来指定泛型类型，表示可以接收任何类型的泛型 类对象。但如果用“？”来指定泛型，则该对象内容不可被修改，但可以将其改 为默认值。
3. 当然也可以不指定泛型类型直接接收，并且还可以修改内容，但这样编译时会出现 安全警告，并且不妥。

**受限泛型：**

受限泛型指的是指定泛型时需要将会受到限制，只能指定某个范围内的类型，受限泛型 分为泛型上限和泛型下限。

1. 泛型上限使用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 类型的子类类型。格式为：

**声明对象：<? extends 类> 对象名称（做方法参数时）**

**定义类：class 类名<泛型标识 extends 类>{}（用于限定该类对象指定泛型的范围）**

1. 泛型下限使用super关键字声明，表示泛型的类型只能是该所指定的类型或者是此 类型的父类类型，或者是Object类。

**声明对象: <? super 类> 对象名称（做方法参数）**

**定义类: class 类名<泛型标识 extends 类>{}（用于限定该类对象指定泛型的范 围）**

注：将受限泛型应用在对象上，一般表示该类对象的通配符“？”不再能接受任意泛型 类对象了，而是只能接受所受限的泛型类型的对象，一般在引用传递时作为方法的 参数，表示该参数只能接受指定范围内的泛型类型对象。

将受限泛型应用在类的定义上，主要是限制该类对象指定泛型类型时的范围。

**指定泛型类型时转型的限制：**

一个类的子类可以通过对象的多态性为其父类实例化，但是在泛型操作中，父类泛型类 型无法使用子类泛型类型进行接收。也就是说声明泛型类对象所指定的类型如果为 String，则在实例化该对象时所指定的泛型类型也必须是String，而不能是其它类型。

**定义泛型接口：**

声明泛型接口与声明泛型类的语法类似，格式为：

**interface 接口名称<泛型标识>{  
 }**

**泛型接口的两种实现方式：**

子类要想实现泛型接口有两种方式：

1. 一是直接实现泛型接口，但子类也需要定义为泛型类，并声明了与接口同样的泛型 标识。
2. 二是直接实现接口，并指定接口中的泛型类型，子类无需定义为泛型类。

**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，所在的类可以是泛型类，也 可以不是泛型类，即泛型方法可以定义在泛型类中，也可以定义在非泛型类中。泛型方 法的定义格式为：

**[访问权限] <泛型标识符> 泛型标识 方法名称(泛型标识 参数名称)**

其中，泛型标识表示该方法返回值类型，泛型标识符标识表示此方法所能接收的泛型标 识，可以是一个也可以是多个。

**通过泛型方法返回泛型类实例：**

可以通过泛型方法返回一个泛型类的实例，但必须泛型方法的返回值类型必须明确指明 返回实例的类型。

**利用泛型同一方法的参数：**

如果有一个方法要求传入的泛型对象的泛型类型是一致的，则可以通过泛型方法指定， 同一方法参数的泛型类型。

**泛型数组：**

使用泛型方法时，也可以传递或返回一个泛型数组。

**泛型的嵌套：**

可以在一个类的泛型中指定另外一个类的泛型。

**标识接口：**

在程序中定义没有方法的接口，这样的接口一般称为标识接口。