## 2.6.1　泛型标记和泛型限定：E、T、K、V、N、?

在使用泛型前首先要了解有哪些泛型标记，如表 2-2 所示。

表 2-2

| 序　号 | 泛型标记 | 说　　明 |
|---|---|---|
| 1 | E-Element | 在集合中使用，表示在集合中存放的元素 |
| 2 | T-Type | 表示 Java 类，包括基本的类和我们自定义的类 |
| 3 | K-Key | 表示键，比如 Map 中的 key |
| 4 | V-Value | 表示值 |
| 5 | N-Number | 表示数值类型 |
| 6 | ? | 表示不确定的 Java 类型 |

类型通配符使用“?”表示所有具体的参数类型，例如 List<?>在逻辑上是 List<String>、List<Integer>等所有 List<具体类型实参>的父类。

在使用泛型的时候，若希望将类的继承关系加入泛型应用中，就需要对泛型做限定，具体的泛型限定有对泛型上线的限定和对泛型下线的限定。

### 1. 对泛型上限的限定：<? extends T>

在 Java 中使用通配符“？”和“extends”关键字指定泛型的上限，具体用法为<? extends T>，它表示该通配符所代表的类型是 T 类的子类或者接口 T 的子接口。

### 2. 对泛型下限的限定：<? super T>

在 Java 中使用通配符“？”和“super”关键字指定泛型的下限，具体用法为<? super T>，它表示该通配符所代表的类型是 T 类型的父类或者父接口。

## 2.6.2　泛型方法

泛型方法指将方法的参数类型定义为泛型，以便在调用时接收不同类型的参数。在方法的内部根据传递给泛型方法的不同参数类型执行不同的处理方法，具体用法如下：

```
public static void main(String[] args) {
    generalMethod("1",2,new Wroker());
}
```