

HenCoder Plus 讲义

泛型中的 T、<>、?、extends、super：情景再归纳

Type Parameter 和 Type Argument

- Type Parameter：
 - `public class Shop<T>` 里面的那个 `<T>`；
 - 表示「我要创建一个 `Shop` 类，它的内部会用到一个统一的类型，这个类型姑且称他为 `T`」。
- Type Argument：
 - 其它地方尖括号里全是 Type Argument，比如 `Shop<Apple>` `appleShop`；的 `Apple`；
 - 表示「那个统一代号，在这里的类型我决定是这个」。
- `Type Parameter` 和 `Type Argument`；泛型的创建和泛型的实例化

T 情景归纳

- 写在类名（接口名）右边的括号里，表示 Type Parameter 的声明，「我要创建一个代号」；
- 写在类或接口里的其他地方，表示「这个类型就是我那个代号的类型」；
 - 很容易搞错的场景：当继承或实现一个泛型类型的时候，如果父类（或父接口）名的右边也写了 `<T>`：

```
interface RepairableShop<T> extends Shop<T> {  
}
```

这个右边父类（接口）右边的 `<T>` 表示：

- 我要对父类（接口）进行实例化，即确定它类型参数的实际值

- 实例化的具体类型是我的这个类型参数

所以它从父类和自己这两个不同角度来看，同时具有「已确定的实例化」和「未确定的声明」这两个属性。

- 使用自己的类型参数 (Type Parameter) 来作为
- 只在这个类或者接口里有用，出了类或接口就没用了（对于泛型方法是，只在这个方法里有用，出了这个方法就没用了）。
- 所以不能这么写：

```
ArrayList<T> list;
```

- 也不能这么写：

```
new ArrayList<T>();
```

? 号情景归纳

- 只能写在泛型实例化的地方：

```
Shop<? super Fruit> fruitShop;
```

表示「这个类型是什么都行，只要不超出 `? extends` 或者 `? super` 的限制」。

- 虽然用于实例化，但因为它表示「类型还有待进一步确定」，它不能用在类型参数最终确定的时候：

```
new Shop<? extends Fruit>(); // 不能这么写
```

- 「类型还有待进一步确定」的特殊场景：嵌套泛型实例化：

```
List<? extends Shop<? extends Fruit>> shops  
    = new ArrayList<Shop<? extends Fruit>>();
```

包在多层尖括号里面的实例化，因为全都属于「类型有待进一步确定」，所以全都可以加问号。

extends 情景归纳

- 用在泛型类型或泛型方法声明的时候，对类型参数设置上界：

```
interface Shop<T extends Fruit & Runnable> {  
    }
```

- 用在泛型实例化时 `?` 号可以出现的地方，用来设置上界，缩小 `?` 号的限制范围。

super 情景归纳

- 只有一个地方：用在泛型实例化时 `?` 号可以出现的地方，用来设置下界，缩小 `?` 号的限制范围。

问题和建议？

课上技术相关的问题，都可以去群里和大家讨论，对于比较通用的、有价值的问题，可以去我们的知识星球提问。

具体技术之外的问题和建议，都可以找丢物线（微信：diuwuxian），丢丢会为你解答技术以外的一切。



觉得好？

如果你觉得课程很棒，欢迎给我们好评呀！<https://ke.qq.com/comment/index.html?cid=381952>

一定要是你真的觉得好，再给我们好评。不要仅仅因为对扔物线的支持而好评（报名课程已经是你最大的支持了，再不够的话 B 站多来点三连我也很开心），另外我们也坚决不做好评返现等任何的交易。我们只希望，在课程对你有帮助的前提下，可以看到你温暖的评价。

更多内容：

- 网站：<https://hencoder.com>；<https://kaixue.io>
- 各大搜索引擎、微信公众号、微博、知乎、掘金、哔哩哔哩、YouTube、西瓜视频、抖音、快手、微视：统一账号「扔物线」，我会持续输出优质的技术内容，欢迎大家关注。
- 哔哩哔哩快捷传送门：<https://space.bilibili.com/27559447>

大家如果喜欢我们的课程，还请去扔物线的哔哩哔哩，帮我素质三连，感谢大家！