

# HenCoder Plus 讲义

---

## 泛型类型的创建

---

### 泛型类的创建

```
public class Wrapper<T> {  
    T instance;  
  
    public T get() {  
        return instance;  
    }  
  
    public void set(T newInstance) {  
        instance = newInstance;  
    }  
}
```

### 泛型接口的创建

```
public interface Shop<T> {  
    T buy();  
    float refund(T item);  
}
```

### 泛型作用小结

1. 帮助检查代码中的类型，提前报错；
2. 自动强制转型。

「创建一个泛型类型」到底是为了什么？

- 本质目标或原因：这个类型的不同实例的具体类型可能会有不同，针对的是实例
- 因此，静态字段和静态方法不能使用泛型类型的类型参数（也就是那个 **T**）

## 继承

```
public class AppleShop extends Shop<Apple> {  
    @Override  
    Apple buy();  
  
    @Override  
    float refund(Apple item);  
}
```

```
public interface RealShop<T> extends Shop<T> {  
    @Override  
    public T buy() {  
        return null;  
    }  
  
    @Override  
    public float refund(Object item) {  
        return 0;  
    }  
}
```

## 类型参数 **<T>** 到底是什么

- 不是一个类，也不是一个接口。只是一个标记符号，一个代号
- 代什么？代表这个类型内部某个通用的类型

## 多个类型参数

```
public interface HenCoderMap<K, V> {
    public void put(K key, V value);

    public V get(K key);
}
```

```
public interface SimShop<T, C> extends Shop<T> {
    T buy(float money);
    float refund(T item);
    C getSim(String name, String id);
}
```

## 类型参数的上界 **<T extends Xxx>**

```
public interface Simhop<T, C extends Sim> extends
Shop<T> {
}
```

```
public interface Simhop<T, C extends Sim & Cloneable &
Runnable> extends Shop<T> {
}
```

## 泛型类型的实例化

- 其实就是「确定这个 T 的实际值」的意思

```
// 左右两边的尖括号都是 ArrayList 的类型参数的实例化
ArrayList<Apple> apples = new ArrayList<>();
```

```
// 左边的 E 是 Repairable 的类型参数的声明；右边的 E 是 Shop
的类型参数的实例化
interface RepairableShop<E> extends Shop<E>{
}
```

## 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。不要仅仅因为对丢物线的支持而好评（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！