



链滴

瞎扯淡：Java 泛型编程 - 泛型类型

作者：[flhuoshan](#)

原文链接：<https://ld246.com/article/1486264929669>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

定义：Java中，泛型类型就是对Java中普通类和接口进行基于类型的参数化。以用来表示TypeName of T的概念。

书写格式:

```
class TypeName<T>{  
/*内容*/  
}''''
```

比如，我们来表示一瓶液体，可以是纯净水，也可以是其他

```
public class Bottle {  
  
private T t;  
  
public void fillWith(T t) {  
this.t = t;  
}  
  
public void showType() {  
System.out.println("瓶里装的是: " + t.getClass().getName());  
}  
}''''
```

接口Interface写法相同。

我们来编写一个测试例子,主程序:

```
public class TestGenericTypes {  
  
public static void main(String[] args) {  
    Bottle ba = new Bottle<>();  
    ba.fillWith(new OrangeJuice());  
    ba.showType();  
    Bottle bb = new Bottle<>();  
    bb.fillWith(new PureWater());  
    bb.showType();  
}  
}
```

OrangeJuice:

```
public class OrangeJuice {  
  
private String name = "OrangeJuice";  
  
public String getName() {  
return name;  
}  
}''''
```

PureWater类似，就不赘述。运行主程序，得到：

瓶里装的是：OrangeJuice

瓶里装的是：PureWater

注意：从Java7开始，泛型在使用时，新建对象时可以不在<>中书写具体的对象名，因为编译会进行自动类型推断。

复合类型参数

如果你以为，泛型类型不过如此，那就错了，如何编写复合类型的泛型类型，请看下面的例子。

先定义一个泛型接口：

```
public interface Carrer {  
    public K getKey();  
    public V getValue();  
}
```

然后定义一个类MyCarrer,实现该接口：

```
public class MyCarrer implements Carrer {  
  
    private K k;  
    private V v;  
  
    public MyCarrer(K k, V v) {  
        this.k = k;  
        this.v = v;  
    }  
  
    @Override  
    public K getKey() {  
        return k;  
    }  
  
    @Override  
    public V getValue() {  
        return v;  
    }  
}
```

在之前的TestGenericType测试主程序：

```
public class TestGenericTypes {  
  
    public static void main(String[] args) {
```

// JDK7之后, 由于编译器会进行类型推断, 新建对象时<>内的内容可写可不写, 但写出来是一个好习惯

```
Carrer c1 = new MyCarrer("Student",10);
Carrer c2 = new MyCarrer("ITer","six or seven years");
// 泛型中可以嵌套泛型使用
Carrer<> c3 = new MyCarrer<>("ITer",new Bottle());
}
}
```

rawTypes

rawtypes, 从字义上讲叫"半成品类型", 在使用泛型类的时候, 如果不在<>中指明泛型类的具体参类型, 那么这就是rawtypes。比如这样:

```
Bottle bb = new Bottle();
```

这样

```
Bottle<PureWater> bb = new Bottle();
```

甚至这样

```
Bottle bb = new Bottle<PureWater>();
```

上述的写法, 会产生编译警告, 需要@SuppressWarnings("rawtypes")进行警告去除, 但不建议这样做, 因为编译器不会对基本类型Bottle进行类型安全检查。建议这样写:

```
Bottle<PureWater> bb = new Bottle<PureWater> ();
```

类型参数的命名规范

- + E - Element (广泛用于Java集合框架)
- + K - Key
- + N - Number
- + T - Type
- + V - Value
- + S,U,V etc. - 2nd, 3rd, 4th 类型

翻看JDK源代码, 可以发现很多类的实现都采用了泛型技术, 比如集合类、容器类等。泛型是Java中其重要的技术, 用好泛型非常有利于写出优雅的代码, 下一篇写写泛型方法。