



链滴

java 的泛型

作者: [sologxl](#)

原文链接: <https://ld246.com/article/1543222092404>

来源网站: [链滴](#)

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

闲来无事，说一个简单的泛型知识吧

泛型，别称为：参数化类型

开发过程中，我们可能都用过泛型，但是可能很少写过泛型。

很普遍的用法就是List< T >,还有dao的增删改查操作。

从而我们能够总结出来，其实泛型就是为了将一系列相同的操作作为一个模板去应用，减少不必要的码。

比如我们要对string数组操作增加，删除，要对int数组操作增加，删除，这个时候直接定义数组list，后把数组类型传入进去，即可对该类型数组进行操作。

为了更好的理解泛型的好处，我们先来假设如果没有泛型，那么我们的代码应该怎么写：

```
StringList list = new StringList();
IntegerList list = new IntegerList();
```

从以上代码可以看出，有多少种类型，就要有多少种对应的数组类型，这样便显得代码特别糟乱和冗。

下面祭出简单的泛型定义，简单的使用示例：

```
package com.gxl.butt;
/**
 *Created by guoxiaolin on 2018/11/26.
 */
public class Students< T > {
    private T score;
    public void setScore(T data){
        this.score = data;
    }
    public T getScore(){
        return this.score;
    }
    public static void main(String[] args){
        Students< MathScore > scoreStudents1 = new Students<>();
        MathScore mathScore = new MathScore();
        mathScore.setId(1L);
        mathScore.setNum(89L);
        Students< EnglishScroe > scoreStudents2 = new Students<>();
        EnglishScroe englishScroe = new EnglishScroe();
        englishScroe.setId(2L);
        englishScroe.setNum(90L);
        scoreStudents1.setScore(mathScore);
        scoreStudents2.setScore(englishScroe);
        System.out.println(scoreStudents1.getScore().getName() + " scroe :" + scoreStudents1.getScore().getNum());
        System.out.println(scoreStudents2.getScore().getName() + " scroe :" + scoreStudents2.getScore().getNum());
    }
}
```

}

接着，我做个简单的小备注，那就是，泛型是在编译期就完成任务，即在运行期，其实已经不存在泛了，所有泛型在编译期就已经固定好具体类型了，就我的理解，泛型只是便于码农理解和编码而已。哈，是不是，很骚气的操作呢？

其实吧，写代码有时候你会发现你一旦理解了，很多东西就变得特别容易。

而且说实话，本猿，真的很少写泛型代码，不知道是我接触的项目太low，还是水平太low，反正就很少用到。