

SpringSide推荐的JDK5.0feature PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022_SpringSide_c104_144688.htm 随着光阴推移，Annotation 慢慢在开源框架中推广，泛型渐渐被程序员们用熟，加上AutoBoxing的小糖，SpringSide终于离不开JDK5.0。

1.AutoBoxing 与 For Each 循环 本来int的非Object性就很无聊，在JDK5.0终于提供了autoboxing功能。这个语法简化糖，被用在了每一个地方。for each 循环也改善了原本总要愣一下的collection遍历。不过对于非JDK基本类型，collection必须用泛型声明，如List。

2. 泛型 泛型大量用于SpringSide Core中的基类，使子类更简洁，基类更强大。当然，基类是难读了，所以才需要社区花上这么长的时间来把看到顺眼。泛型使用的有两个定式：2.1 避免强制类型转换 如果函数输入参数里含Class类型，而返回值又是该Class的实体，应该将该函数设为泛型函数。最典型的例子是HibernateGenericDao的get() 函数

```
public T> T get(ClassT> entityClass, Serializable id) { return (T) getHibernateTemplate().get(entityClass, id).}
```

其中眼花缭乱的第一个声明这是一个泛型函数，第2个T声明返回值为T，第三个Class代表 T.class。基类写的辛苦，但子类用得爽快

```
Book book = (Book)manager.get(Book.class,1) 简化成了 Book book = manager.get(Book.class,1).
```

2. 2 泛型配合反射API从T获得 T.class。最典型的例子HibernateEntityDao，子类只需以下定义，即获得要管理的Entity的Class。BookManager extends HibernateEntityDaoBook> 此时子类只要声明一次T，上面的

```
Book book = (Book)manager(Book.class,1) 就能简化成 Book =
```

manager.get(1). 一举两得地既避免了强制类型转换，又声明了T.class 供框架使用，无须再在Manager的构造函数或getEntityClass()函数定义entityClass，。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com