

Java之泛型进阶——泛型代码转化为普通代码

原创 2016年03月09日 21:25:43

2049

0

7

Java之泛型进阶——泛型代码转化为普通代码

简介

本篇是java中泛型深入一点的东西，主要是泛型代码如何转换成普通的Java代码，类型擦除以及一个通过反射绕开编译器类型检测的例子来证明类型在运行时是真正被擦除的。

泛型的转化

用泛型编写的Java程序和普通的Java程序基本相同，只是多了一些参数化的类型同时少了一些类型转换。实际上泛型程序也是首先被转化成一般的不带泛型的Java程序后再进行处理的，编译器自动完成了从Generic Java到普通Java的翻译。转化主要是两个关键点，擦除和桥方法。

擦除的概念

将泛型代码转换成普通的Java代码的过程是由编译器（如javac）来完成的，虚拟机并不负责这一任务。当编译器对带有泛型的Java代码进行编译时，它会去执行类型检查和类型推断，然后生成普通的不带泛型的字节码，这种普通的字节码可以被一般的Java虚拟机接收并执行，这中技术就叫做擦除（erasure）。可见，编译器可以在对源程序（带有泛型的Java代码）进行编译时使用泛型类型信息保证类型安全，对大量如果没有泛型就不会去验证的类型安全约束进行验证，同时在生成的字节码当中，将这些类型信息清除掉。

简单示例

```
List<String> strings = new ArrayList<>();  
List<Integer> integers = new ArrayList<>();  
System.out.println(strings.getClass() == integers.getClass());//true
```

上面代码输出结果并不是预期的false，而是true。其原因就是泛型的擦除。

泛型转化大致步骤

具体的转化过程大致分为以下几个部分：

将参数化类型中的类型参数“擦除”掉；

```
//此处的String就是类型参数  
List<String> list = new ArrayList<>();  
//转换为  
list = new ArrayList();
```

将类型变量用“上限（upper bound）”取代，通常情况类型变量的上限是Object，但是也有一些情况是



Oscar Chen (<http://blog....>)

+ 关注

(<http://blog.csdn.net/chenghuaying>)

原创

粉丝

喜欢

182

14

4

- > CentOS 集群机器之间ssh免密
([/crave_shy/article/details/72964997](http://crave_shy/article/details/72964997))
- > JVM-内存管理-运行时数据区域
([/crave_shy/article/details/56675052](http://crave_shy/article/details/56675052))
- > JVM-Blog目录
([/crave_shy/article/details/56675032](http://crave_shy/article/details/56675032))
- > JVM-为什么要学JVM
([/crave_shy/article/details/56673439](http://crave_shy/article/details/56673439))

更多文章

(<http://blog.csdn.net/chenghuaying>)

在线课程



(http://edu.csdn.net/huiyiCourse/series_detail?utm_source=blog7)

【直播】机器学习&数据挖掘7周实训--韦玮

(http://edu.csdn.net/huiyiCourse/series_detail/54?utm_source=blog7)



(http://edu.csdn.net/combo/detail/471?utm_source=blog7)

【套餐】系统集成项目管理工程师顺利通关--徐朋

(http://edu.csdn.net/combo/detail/471?utm_source=blog7)

限定类型参数的边界的。这里的类型变量是实例域、本地方法域方法参数以及方法返回值中用来标记类型信息的“变量”，如我们在集合中常见的“E”。

```
void add(int index, E element);
//转换为
void add(int index, Object element);

class<T> {private T t;}
//转换为
class {private Object t;}
```

添加类型转换并插入“桥方法”（bridge method），以便Java方法的重写（overridden）可以正常的工作，当一个类实现了一个参数化的接口或是继承了一个参数化的类时，需要引入桥方法。。

源码：

```
public class Person implements Comparable<Person>{
    protected int age;
    private Person person;

    public Person(int age) {
        this.age = age;
    }

    public Person getPerson() {
        return person;
    }

    @Override
    public boolean compareTo(Person that) {
        return this.age > that.age;
    }
}

interface Comparable<T> {
    boolean compareTo(T that);
}
```

编译器编译之后的代码：

```
public class Person implements Comparable<Person>{
    ...
    /**
     * 这个方法是重写了接口Comparable中的compareTo方法，但是我们注意到接口中compareTo方法的
     * 转换成 compareTo(Object that)这种，也就是说当实现类实现此方法时，方法名称相同并且方法
     * 法中的参数不是，其原因是编译器在编译的时候加入了桥方法。
     */
    @Override
    public boolean compareTo(Person that) {
        return this.age > that.age;
    }

    /**
     * 编译器加入的桥方法
     */
    public boolean compareTo(Object that) {
        return this.compareTo((Person) that);
    }

    ...
}
```

在某些情况下，擦除技术需要引入类型转换（cast），这些情况主要包括：

情况 1. 方法的返回类型是类型参数；

```
E elementData(int index) {
    return (E) elementData[index];
}
```

情况 2. 在访问数据域时，域的类型是一个类型参数（也就是上面使用桥方法的情况）。

具体擦除示例

源码：

```
package org.andy.items.thkinjava.generics.generic2016.brige;

/**
 * Author: oscar
 * Create Data: 10/03/16
 */
public class Person implements Comparable<Person>{
    protected int age;

    public Person(int age) { this.age = age; }

    @Override
    public boolean compareTo(Person that) { return this.age > that.age; }

}

interface Comparable<T> { boolean compareTo(T that); }
```

编译后：

```
public class Person implements Comparable{
    protected int age;

    public Person(int age) { this.age = age; }

    @Override
    public boolean compareTo(Person that) { return this.age > that.age; }

    public boolean compareTo(Object that) {
        return this.compareTo((Person) that);
    }
}

interface Comparable{ boolean compareTo(Object that); }
```

使用反射来避开泛型的验证

```
public class HowToAvoidGenericRestrict {

    /**
     * 简单演示一下如何使用反射向一个拥有泛型的List中添加泛型指定类型之外的元素
     * @param args ignore
     */
    public static void main(String[] args) throws NoSuchMethodException, InvocationTargetException, IllegalAccessException {
        //声明一个只能存放String类型的list
        List<String> list = new ArrayList<>();
        //向上面一个list中加入一个Integer类型的元素
        //list.add(1); //compile error
        System.out.println("Before add, the size of list is : " + list.size());
        Class<?> clazz = list.getClass();
        Method method = clazz.getMethod("add", Object.class);
        method.invoke(list, 1);
        System.out.println("After add, the size of list is : " + list.size());

        System.out.println(list.get(0));
    }
}
```

输出：


```
Before add, the size of list is : 0
Exception in thread "main" java.lang.ClassCastException: java.lang.Integer cannot be cast to java.lang.String
After add, the size of list is : 1
    at org.andys.items.thkinjava.generics.generic2016.avoid.generic.restrict.HowToAvoidGenerics.main(HowToAvoidGenerics.java:10)
```

版权声明：本文为博主原创文章，未经博主允许不得转载。



标签：[java \(http://so.csdn.net/so/search/s.do?q=java&t=blog\)](http://so.csdn.net/so/search/s.do?q=java&t=blog) /
[泛型 \(http://so.csdn.net/so/search/s.do?q=泛型&t=blog\)](http://so.csdn.net/so/search/s.do?q=泛型&t=blog) /
[泛型转化 \(http://so.csdn.net/so/search/s.do?q=泛型转化&t=blog\)](http://so.csdn.net/so/search/s.do?q=泛型转化&t=blog) /
[类型擦除 \(http://so.csdn.net/so/search/s.do?q=类型擦除&t=blog\)](http://so.csdn.net/so/search/s.do?q=类型擦除&t=blog) /
[反射避开泛型 \(http://so.csdn.net/so/search/s.do?q=反射避开泛型&t=blog\)](http://so.csdn.net/so/search/s.do?q=反射避开泛型&t=blog) /

0 条评论

 qq_36596145 (http://my.csdn.net/qq_36596145)
(http://my.csdn.net/qq_36596145)



发表评论

暂无评论

相关文章推荐

Java返回类型泛型的用法小结 ([b1412/article/details/7075791](/b1412/article/details/7075791))

关于Java泛型的基本用法就不多说了，主要是一个编译期的检查，也避免了我们代码中的强制转换，比较经典的用法有泛型DAO,泛型Page的DTO。现在我要说的是一个挺有趣但是貌似还不是太多的人...



b1412 2011-12-15 23:10 27424

java泛型T.class的获取 ([z69183787/article/details/42142977](/z69183787/article/details/42142977))

很早之前写过利用泛型和反射机制抽象DAO，对其中获取子类泛型的class一直不是很理解。关键的地方是HibernateBaseDao的构造方法中的 [java] view ...



z69183787 2014-12-25 10:24 13517

Java获取泛型的Class对象 ([weinianjie1/article/details/6722247](/weinianjie1/article/details/6722247))

工作中瞄到的，在此收集了public class RawDao { protected Class clazz; @SuppressWarnings("unchecked")
...



weinianjie1 2012-02-28 17:43 15966

AlarmManager研究 ([thinkinwm/article/details/40119033](/thinkinwm/article/details/40119033))

<http://my.oschina.net/youranhongcha/blog/149564> 目录[-] 1.概述 2.AlarmManager 2.1 ...



thinkinwm 2014-10-15 21:41 657

C语言的基本文件操作 (/lan74_/article/details/53981296)

C语言的基本文件操作序列1：基本的txt文件读入读出先来段代码看看#include int main() { char a[] = "hellow\n"; char buf[6]; ...



LAN74_ 2017-01-02 18:56 104

MBProgressHUD的简单封装 (/anywhereios/article/details/50813485)

下面是依据MBProgressHUD提供的分类方法: /** * 显示一般信息 */ + (void)show:(NSString *)text icon:(NSString *)icon vi...



anywhereIOS 2016-03-06 15:09 1414

《深入理解java虚拟机》学习笔记9——并发编程（一） (/chjttony/article/details/7982231)

随着多核CPU的高速发展，为了充分利用硬件的计算资源，操作系统的并发多任务功能正变得越来越重要，但是CPU在进行计算时，还需要从内存读取输出，并将计算结果存放到内存中，然而由于CPU的运算速度比内存高...



chjttony 2012-09-15 15:11 3251

think in java interview-高级开发人员面试宝典(二) (/lifetragedy/article/details/9751079)

从现在开始，以样题的方式——例出各种面试题以及点评，考虑到我在前文中说的，对于一些大型的外资型公司，你将会面临全程英语面试，因此我在文章中也会出现许多全英语样题。这些题目来自于各个真实的公司，公司名我...



lifetragedy 2013-08-05 00:43 38487

就 3 点，提升工作效率 (/foruok/article/details/76950020)

要想提高工作效率，不论你看什么书，看什么文章，用什么工具，只有下面这三点最重要



foruok 2017-08-09 09:10 102005

计算机视觉、机器学习相关领域论文和源代码大集合--持续更新..... (/yingchunhua365/article/details/14043609)

计算机视觉、机器学习相关领域论文和源代码大集合--持续更新..... zouxy09@qq.com
http://blog.csdn.net/zouxy09 注：下面有project网站的大部分都有pap...



diaoguangqiang 2013-12-03 10:41 608

计算机科学精彩帖子收集--JAVA和分布式专栏 (/zhang_2017/article/details/54926625)

之前收集了一个计算机科学精彩帖子收集，渐渐发现帖子越来越大，所以现在干脆把Java和分布式的部分单独一贴。Java jdk下载 http://openjdk.jav...



zhang_2017 2017-02-08 14:37 2250



编程新手导论（转载） (http://weiyinchao88.iteye.com/blog/1414243)

第二部分 导论，这一部分主要是关于编程的导论，（要懂得一点思想具备一点常识）《设计，编码，，与软工》（编程与思想）这一章解释了三种思想，原语，抽象，组合，，和软件开发的二个重要过程，，软件工程的相关概念，是编程入门的关键（要懂得一点领域内的数学）《数学与算法》（编程与数学）计算机整个就是架构在数学上的，跟计算机平台实现，算法设计，，架构密切相关，，真正要深入编程，，对数学的学习是必须的，，千万不要相

 weiyinchao88 2012-01-22 10:26  1803



GitHub 优秀的 Android 开源项目第二篇——转自多篇网络文章 (/lhx19930428/article/details/56960703)

原文地址为<http://www.trinea.cn/Android/android-open-source-projects-view/>，作者Trinea 主要介绍那些不错个性化的V...

 lhx19930428 2017-02-24 23:18  713



Java高质量代码之 — 泛型与反射 (<http://ray-yui.iteye.com/blog/1933127>)

[size=large] 前言:由于上一个星期工作繁忙,利用上下班和晚上睡前空余的时间拜读了秦小波老师的《改善Java程序的151建议》,感觉豁然开朗,注意到了很多平时在编写代码中并不会注意的问题,甚至感觉自己对Java只是略懂皮毛,不足以登大雅之堂,特此与读者分享读书笔记.[color=red]以下内容摘自《改善Java程序的151建议》一书和笔者的理解[/color] [/size] [size=large] Java高质量代码系列文章 面向对象篇:[url]<http://ray-yui.iteye.com/blog/1926984>[/url] 数据类

 ray_yui 2013-08-29 09:36  14420



编程新手导论（转载） (<http://mmdev.iteye.com/blog/1451514>)

第二部分 导论，这一部分主要是关于编程的导论，（要懂得一点思想具备一点常识）《设计，编码，，与软工》（编程与思想）这一章解释了三种思想，原语，抽象，组合，，和软件开发的二个重要过程，，软件工程的相关概念，是编程入门的关键（要懂得一点领域内的数学）《数学与算法》（编程与数学）计算机整个就是架构在数学上的，跟计算机平台实现，算法设计，，架构密切相关，，真正要深入编程，，对数学的学习是必须的，，千万不要相

 mmdev 2012-01-22 10:26  771

编程新手导论（转载） (/china_video_expert/article/details/7213519)



第二部分 导论，这一部分主要是关于编程的导论，（要懂得一点思想具备一点常识）《设计，编码，，与软工》（编程与思想）这一章解释了三种思想，原语，抽象，组合，，和软件开发的二个重要过程，，软件工程的相...

 szu030606 2012-01-22 10:26  9554

JAVA学习代码——泛型 (/reserved_person/article/details/52195458)



泛型的定义 泛型的格式 package 泛型; import java.util.ArrayList; import java.util.List; /** * 如何用泛型规定集合的类型：泛型...

目录

 Reserved_person 2016-08-12 21:19  77
喜欢

GitHub 优秀的 Android 开源项目 淘宝技术牛p博客整理开发中最常用的GitHub上 优秀的 Android 开源项目整理（精品） (<http://f059074251.iteye.com/blog/2200969>)

原文地址为<http://www.trinea.cn/android/android-open-s>

 f059074251 2014-01-09 17:35  2566

C++之路 (/wlywly1987/article/details/14001437)

【C++概述】 C++这个词在中国大陆的程序员圈子中通常被读做“C加加”，而西方的程序员通常读做“C plus plus”，“CPP”。它是一种使用非常广泛的计算机编程语言。C++是一种静态数...



wlywly1987 2013-11-01 15:01 657

诊断 Java 代码: 轻松掌握 Java 泛型 (<http://ericyou.iteye.com/blog/298356>)

Eric E. Allen (eallen@cs.rice.edu), 博士研究生, Rice 大学 Java 编程语言团队 2003 年 5 月 14



nbkangta 2008-06-26 02:21 478

Effective Java 第五章 泛型 (含代码) (/qq_34203494/article/details/56847186)

Effective Java 第五章 泛型 (含代码)



qq_34203494 2017-02-24 17:57 3

关于java中的泛型以及用面向对象的继承性来简化DAO的代码量 (<http://fengyanzhang.iteye.com/blog/1939855>)

今天看到这段代码,加深了对java中的泛型编程的理解: public class BaseDao<T>{ private Class<T> entityClass; @Autowired private HibernateTemplate hibernateTemplate; /** * 通过反



_Yggd 2013-09-09 22:53 568

windows类书的学习心得 (/thanklife/article/details/38298871)

windows类书的学习心得 这篇文章应该是凑的,不够很长,还是值得读的,转发来。下满原网址: <http://www.blogjava.net/sound/archive/2008/08/2...>



thanklife 2014-07-30 13:11 8727

JCodeModel定义生成代码中的泛型变量 (<http://chinaestone.iteye.com/blog/1745169>)

JClass temp = new JCodeModel().ref(Class.class).narrow(cm.ref(Serializable.class).wildcard()); 生成内容: Class<T> extends Serializable>; JClass str = cm.ref("org.isunday.tools.cg.codemodel.core.Generator"); JClass rlist = cm.ref(List.class).narrow(str);



ChinaEstone 2012-12-10 17:30 700

test (/xiang_freedom/article/details/54949602)

《Android开发艺术探索》笔记本笔记整理自: https://www.gitbook.com/book/tom510230/android_ka_fa_yi_shu_tan_suo/detai...



xiang_freedom 2017-02-09 17:34 0

java泛型详解下篇之纯代码讲解 (<http://jgsj.iteye.com/blog/2050419>)

这篇我将在上篇的基础上进行实例编程讲解,也就不多说废话了,直接进入编程,剖析java的多种泛型类型。 By-何泽江 (KJ Internationl company) package demoTest; /** * 普通泛型类 * @author <ahref="mailto:hezj@china.kjlink.com" href="mailto:hezj@china.kjlink.com">hezj



jgsj 2013-07-09 10:38 86

新手编程导论 (/zhenweilee3471/article/details/52787931)

第一部分 前言 By Chenyi 眼前这本书充分体现了作者的所思、所想、所感，他用自己独特的眼光审视着计算机技术的世界，也用自己独特的思维逻辑对技术进行解读，并用自己特有的，呵呵，偶...



zhenweilee3471 2016-10-11 14:37 1229

C# 冒泡排序法示例代码(包含泛型方法) (http://txf2004.iteye.com/blog/472732)

```
using System;using System.Collections.Generic;using System.Text; namespace bleb_sort{ class Program { public class SwapObj { public static void swap<T>(ref T a, ref T b) { T c; c=a; a=b; b=c; } } static void Main(s
```



txf2004 2007-05-31 14:55 842

Java基础进阶_day07_(泛型,Collection集合,迭代器,增强for循环) (/l631106040120/article/details/69055820)

泛型 Collection集合



L631106040120 2017-04-04 10:45 448

Java下的Framework编写(反射,泛型,元数据,CGLib,代码动态生成,AOP,动态语言嵌入)(1) --序 (http://txf2004.iteye.com/blog/499521)

作者：江南白衣序 以Ruby为助力的Ruby on Rails 横空出世，又刷新了一次人们对Framework的认识：Closures、Maxin、Continuations、Metaprogramming、Reflection，又一堆名词砸得人猝不及防。Ja



txf2004 2005-10-17 11:03 307

Animation动画详解(五)——ValueAnimator高级进阶（一） (/ni357103403/article/details/50724569)

转载自http://blog.csdn.net/harvic880925/article/details/50546884 相关文章：1、《Animation 动画详解（一）——al...



ni357103403 2016-02-23 17:08 165

诊断 Java 代码: 轻松掌握 Java 泛型 (http://gongstring.iteye.com/blog/321667)

本月的 诊断 Java 代码介绍泛型类型 (generic type) 和支持它们的特性，计划在 2003 年末发布的 Tiger，也就是 Java V1.5 中打算包含这些泛型和特性。Eric Allen 提供了代码样本，这些样本通过重点描述诸如基本类型的限制、受限泛型和多态方法之类的 Tiger 特性来说明泛型类型的优缺点（即将发表的专栏文章将讨论其它特性，比如 Tiger 中泛型类型的特定表现以及可能扩展为 Tiger 之外的泛型类型）。请通过单击文章顶部或底部的 讨论进入论坛，与作者和其他读者分享您对本文的心得体会。J2SE 1.5 - 代号为 Tiger - 计划在 2003 年



gongstring 2009-02-01 19:34 463

关于c++的专业描述 (/guanyudatou/article/details/6464206)



guanyudatou 2009-05-06 20:00 1109

Java下的Framework编写(反射,泛型,元数据,CGLib,代码动态生成,AOP,动态语言嵌入)(1) --序 (http://webzixue.iteye.com/blog/953247)

作者：江南白衣序 以Ruby为助力的Ruby on Rails 横空出世，又刷新了一次人们对Framework的认识：Closures、Maxin、Continuations、Metaprogramming、Reflection，又一堆名词砸得人猝不及防。Ja



webcode 2005-10-17 11:03 338

编程及C/C++初学者 FAQ (/qcx1221/article/details/51506196)

编程及C/C++初学者 FAQ 本文原为本人在论坛所发若干帖，意在集中解决新手学习C/C++语言时将遭遇到的各类问题，网友反馈情况良好，集合修订后作为个人作品贴于此处。 本贴主要分析概念原理和解...



qcx1221 2016-05-26 12:31 1169

编程新手导论（转载） (http://weimou66.iteye.com/blog/1417194)

第二部分 导论，这一部分主要是关于编程的导论，（要懂得一点思想具备一点常识）《设计，编码，，与软工》（编程与思想）这一章解释了三种思想，原语，抽象，组合，，和软件开发的二个重要过程，，软件工程的相关概念，是编程入门的关键（要懂得一点领域内的数学）《数学与算法》（编程与数学）计算机整个就是架构在数学上的，跟计算机平台实现，算法设计，，架构密切相关，，真正要深入编程，，，对数学的学习是必须的，，千万不要相



weimou66 2012-01-22 10:26 532

编程及C/C++初学者 FAQ (/star267/article/details/8151081)

编程及C/C++初学者 FAQ 本文原为本人在论坛所发若干帖，意在集中解决新手学习C/C++语言时将遭遇到的各类问题，网友反馈情况良好，集合修订后作为个人作品贴于此处。 本贴主要分析概念原理和解...



star267 2012-11-06 00:05 3808

读编写高质量代码整理（九）-警惕泛型不能协变和逆变 (http://ljh143.iteye.com/blog/1943881)

先引进两个名词“协变”、“逆变” 协变——用一个窄类型替换宽类型 逆变——用宽类型替换窄类型 协变代码如下：
public class Base { public Number doStuff() { return 0; } }
class Sub extends Base { @Override
public



elite20130514 2013-09-18 16:24 248

windows类书的学习心得 (/g272165123/article/details/8913094)

原文地址：http://www.blogjava.net/sound/archive/2008/08/21/40499.html 内容如下： 创建人： paul 现在的计算机图书发展的可真快,很久...



g272165123 2013-05-10 23:47 5837