场景式教学

改错学Java—程序设计初学者的秘密武器

Java 改错学习法

朱福喜 编著

- 颠覆传统的教学方法
- 以老带幼的对话教学模式
- 克服学习中的畏难心理
- 纠正编程中易发生的错误

在改错中
学Java
用Java

清华大学出版社

Java 改错学习法

朱福喜 编著

清华大学出版社
北京

内 容 简 介

本书作者虚构一个教学场景，采用一老一少的对话形式，将自己多年教学经验融入其中，颠覆传统教学模式，创新一种改错学 Java 的方法，帮助学习者树立正确的学习观念，使初学者明白，改错也是一种学习方式，在改正错误的过程中也能够学到和巩固很多基础知识，只要有了足够的基础，就可以编写出非常复杂和漂亮的程序。

本书从基本概念入手，对开始学习 Java 编程时会发生的错误进行纠正和引导，从而克服学习过程中的畏难心理，在改错中逐步成长。本书从简单的例子开始，循序渐进，帮助初学者提高查错、排错和改错的能力。相信通过足够多的练习，读者定能熟悉 Java 程序设计的精华，进而提高 Java 编程能力。

图书在版编目（CIP）数据

Java 改错学习法/朱福喜编著.—北京：清华大学出版社，2013.1

ISBN 978-7-302-30346-6

Ⅰ. ①J… Ⅱ. ①朱… Ⅲ. ①JAVA 语言—程序设计 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2012）第 240839 号

责任编辑：夏非彼
封面设计：王　翔
责任校对：闫秀华
责任印制：李红英

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：185mm×230mm　　**印　　张**：20.5　　**字　　数**：500 千字
版　　次：2013 年 1 月第 1 版　　**印　　次**：2013 年 1 月第1 次印刷
印　　数：1～4000
定　　价：39.00 元

产品编号：047270-01

常言道，万事开头难，学 Java 更是如此。编程还有与许多事情不一样的地方，例如，说话时遗漏个字，一般人也听得懂，写文章时输入个错别字，别人可能也能看懂，但编程不一样，它不能出现半点差错，否则会毫不留情，不让你通过，因此使得很多人，甚至是聪明人都产生了畏难心理。如何让初学者克服这个障碍呢？这就是本书的出发点。

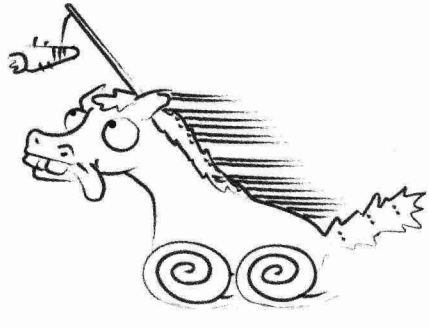
编程的人不需要太聪明，但需要培养机械和逻辑思维的能力。有时候，过于聪明反而会给自己造成更多的编程困难。本书正是基于这一点，虚构了一个场景，刻画了两个不同的形象。

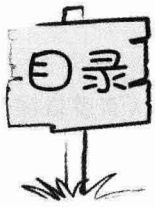
一休：日本动画片《聪明的一休》中的主人公，聪明，肯动脑筋，遇到困难的问题就冥思苦想，是很有灵气、顽皮的小精灵的象征。

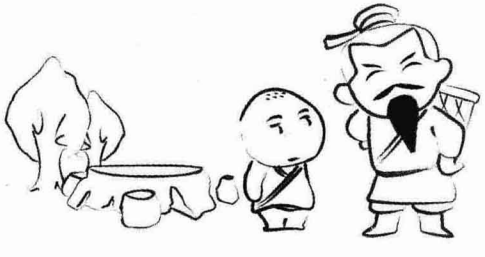
愚公：中国寓言中倔强的老头，为搬去挡在家门的大山，挖山不止，是有勇气、肯实干的象征。

在本书虚构的学习环境中，一休是想学习 Java 的小生；愚公是埋头编程的 Java 教头。愚公教一休**如何改错学 Java**，这便是本书的主题。本书就围绕他们的对话而展开。

参与本书编写的人员，除封面署名作者外，还有朱三元、刘世超、朱丽达、陈端直、王瑛、周孟、束元等，在此编者对他们表示衷心的感谢。







1 动手才是硬道理

一休：人人都夸我聪明，好像没有什么能够难倒我的问题。可是，碰到 Java 编程这玩意，我就不灵了，动手就错，我想破了脑壳就是想不出用什么招数治它，气得我摔坏机器也无济于事。

愚公：哈哈，我知道其中的缘由。编程这活不是光靠聪明就能搞定的。

一休：不靠聪明靠什么？

愚公：靠的是一股傻劲，就像我一样，每天挖山不止。

一休：我学的是“假挖”（Java），难道要真的挖吗？

愚公：我指的是要多动手，动手才是硬道理。

一休：您能具体讲解一下吗？

愚公：Java 作为一门编程语言，最好的学习方法就是动手编写代码。当学习一个概念后，就可以自己编写一个简单的程序来运行一下，看看有什么结果，然后再多调用几个类的方法，比较一下运行结果，这样就能够非常直观地学会 Java 了，而且记忆深刻。

学会以后，不能仅仅满足于把代码调通，你应该想想如果我不这样编写，换个方式行不行。记住，学习编程就是一个破坏的过程，把书中的范例、自己学习的程序在运行通过以后，不断尝试利用不同的方法实现，不断尝试破坏代码的结构，看看会有什么结果。通过这样的方式，你就能很彻底地掌握 Java。

一休：是啊，我也想这么做。但一动手就会碰到问题，我想啊想，就是想不出为什么。

愚公：Java 语言和人类语言在某些方面是类似的，只是它是讲给计算机听的，在许多方面都存在一些特殊约定，要通过计算机的运行去体会它，而不是靠你有多聪明去想象它。

2

从简单的程序开始

一休：愚公，在学习过程中，我点子虽多，但总觉得 Java 太难。

愚公：万事开头难，我们可以从简单的程序开始。举个例子，我们都编过 HelloWorld 吧？

一休：是的，我会编，我编给你看。

```
public class HelloWorld
{   public static void Main(String[] args)
    {   System.out.println("Hello World");
    }
}
```

愚公：你看看，一动手就错了！

一休：哪有错？编译都通过了。

愚公：你执行看一下。

一休：噢，的确出错了，错误信息如图 1 所示。

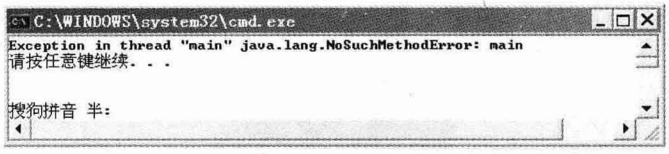


图 1　显示无 main 方法的出错信息

愚公：错在哪里，你知道吗？

一休：应该是没有 main 方法，但是我编了 Main 方法，编译也没有指出错误啊？

愚公：你编了一个 Main 方法，这在语法上没有错，但是没有 Java 虚拟机要的 main 方法，main 方法是入口，没有它，就像建造一栋房子没有预留门的位置一样，没有地方可以进入。

一休：那 main 方法前面还带了很多头衔，可以不这样写吗？

愚公：不行，一定要写 public static void main。很多初学者都不理解为什么 main 方法一定要这样来定义：

```
public static void main (String[] args)
```

是否可以修改一下呢？很简单，把 main 换个名字运行一下，看看报什么错误，然后根据出错信息进行分析；把 main 之前的 public 去掉再试试看；把 static 去掉，看看还能不能运行；是否一定要是 String[]数组呢？把 String[]改成 int[]或者 String 试试看；想知道参数名称是否必须写为 args 吗？可以把 args 改成其他名字，看看运行结果如何。

一休：我把 HelloWorld 程序反复改了七八次，不断运行，分析运行结果，终于明白原因了——Java 虚拟机就是这样约定的，难道一点也不能改动吗？

愚公：也不是，public 可以与 static 交换一下位置；String[] args 可以写成 String args[]；args 也可以写成其他名字。

一休：终于会写 main 方法了。

愚公：体会到了吗？所以，动手才是硬道理呀！

3

如何设置路径

一休：愚公，我是想学您挖山不止的精神，可是总是使不上劲啊！

愚公：为什么？

一休：每次编译和运行程序，不是找不到这个文件就是找不到那个文件，有时候把我气得想砸机器。

愚公：哦，我明白了。你应该了解关于寻找 class 文件的原理，建议开始学习 Java 的时候在命令行窗口编译和运行 Java 程序，不要借助 JCreator 或者 Eclipse 等 IDE 去做那些事情。首先要正确设置 classpath 环境变量，它就是一个路标，指出你要找的类文件可能存放在哪些路径之下。

一休：我也对照书中讲的方法设置了 classpath，但有时候就是找不到。

愚公：运行时的状态不好掌握，可能是你设置后没有重新启动计算机，也可能是运行其他程序时改变了环境，但我有个“笨”办法：

- 编译程序时使用：javac -classpath yourpath *.java。
- 执行程序时使用：java -classpath yourpath *.class。

“-classpath”的作用是指定查找用户类文件和注释处理程序的位置，所以它后面要跟一个路径，也就是你的源程序文件或类文件所放的位置。

一休：有时候又显示找不到命令文件 java 和 javac 在哪里。

愚公：那你应该先设置好 path 变量，使系统可以找到 jdk 的 bin 目录。我也有一个“笨”办法，即把编译或执行命令文件的绝对路径指出来，例如：

```
>d:\jdk16\bin\javac -classpath yourpath *.java
```

再“笨”一点的办法就是把要编译的源程序的位置也指出来，例如：

```
>d:\jdk16\bin\javac -classpath d:\myprog\Hello.java
```

这个“笨”办法一定可行。你还可以利用下面的方法指定存放生成类文件的位置：

```
javac -d <目录> *.java
```

一休：以上这些都是没有办法时的办法。

愚公：是的，当然，你还是应该从原理上弄清楚为什么要设置这些环境变量。首先，了解过计算机的基础知识后就会知道，设置 path 环境变量是为了使系统能够找到要执行的命令；而设置 classpath 的目的是告诉 Java 虚拟机去哪里寻找你的 class 文件。JVM 要依靠 AppClassLoader 加载类文件。

可以在前面编写的 HelloWorld 类中加入如下语句：

```
System.out.println(new HelloWorld().getClass().getClassLoader().toString());
```

这样就可以简单测试出 HelloWorld 的加载类为 sun.misc.Launcher$AppClassLoader。

一休：好的，我马上去试试。

愚公：这就对了，一定要多动手，万事开头难！

4

正确引用对象

一休：愚公，我又碰到问题了，还是那个 main。我把字符串"Hello World"搬到 main 之外就不行了，代码好像没有什么问题：

```
class StaticError
{   String mystring="Hello World";
    public static void main(String args[])
    {   System.out.println(mystring);
    }
}
```

但编译时错误信息为：

```
nonstatic variable mystring cannot be referenced from a static context
```

愚公：犯错误不要紧，只要每次不犯同样的错误就行了，正所谓“失败是成功之母”，如果把所有的错误都犯尽了，编写的程序也就不会再出错了。

一休：您的意思是让我使劲犯错误吗？

愚公：那也要看是什么样的错误。犯一个错误，纠正之后要弄懂一个概念。就你这个错误而言，还是过程式编程的思维：所有的过程可以访问所有的全局变量。

在 Java 中，过程和函数统称为方法，方法分为类方法和实例方法，变量也分为类变量和实例变量。所谓类变量和类方法都是对整个类而言的，类方法只能访问类变量，换句话说，类方法不能访问实例变量。实例方法可以访问实例变量和类变量。该程序中，main 方法是静态的，只能访问静态变量。

一休：有点明白，但还是觉得抽象。

愚公：比方说，“人民”是一个类，“制定宪法”是一个类方法，“人大常委会”是一个类变量，张三、李四、王麻子是“人民”这个类的实例变量，“制定宪法”可以使用“人大常委会”这个类变量，但“制定宪法”就不能专门针对张三、李四、王麻子。这有点类似于“为人民服务”，而不是针对某个人服务。

一休：怪不得总听别人说“我是为人民服务的，不是为你一个人服务的”，那么这个问题怎么解决呢？

愚公：解决的办法有两种，第一种方法是将变量 mystring 改成类变量。

一休：我来改，就在声明 mystring 字符串的前面加上 static 即可，程序写好了。

```
class StaticError
{   static String mystring="hello";
    public static void main(String args[])
    {   System.out.println(mystring);
    }
}
```

愚公：改得不错。第二种方法是先先创建一个类的实例，然后通过该对象访问该变量。

一休：这个方法我需要想一想。

愚公：再具体一点就是先创建当前类的一个对象，然后通过该对象访问 mystring 变量。

一休：哦，代码编写出来了。

```
class NoStaticError
{   String mystring="hello";
    public static void main(String args[])
    {   NoStaticError  noError;
        noError=new NoStaticError();
        System.out.println(noError.mystring);
    }
}
```

愚公：这就对了。有时候碰到一个错误你应该高兴，高兴的是又有一个机会学到新的知识了。