# java夜未眠读书心得

来源：网络 作者：逝水流年 更新时间：2024-08-08

*通过参加这次的java夜未眠读书培训，这个对于系统的扩展是很有用处的。new 已经决定了你肯定要这个类，而静态方法却可以修改，不一定肯定返回你方法所在的类，可以在必要时替换为一个子类。下面是范文网小编为大家收集整理的java夜未眠读书...*

通过参加这次的java夜未眠读书培训，这个对于系统的扩展是很有用处的。new 已经决定了你肯定要这个类，而静态方法却可以修改，不一定肯定返回你方法所在的类，可以在必要时替换为一个子类。下面是范文网小编为大家收集整理的java夜未眠读书心得，欢迎大家阅读。

最近我在慧桥上java的培训课，老师推荐了本不错的java教材《Think in java 》

看书要学会做笔记，才会有更大收获，每天读30分钟，希望能坚持下去

今天重新翻了第一章

Introduction to object 面向对象主要围绕的五个要点

1.Everything is an object

每个东西都可以看作一个对象，它储存了你所需要的必要信息，你可以通过特定的请求方式来向这个对象请求你所需要的东西

2.A Program is a bunch of objects telling each other what to do by sending messages

一个程序由很多对象组成，你可以向对象发送消息来告诉他们做什么和怎样做

3.Each object has its own memory made up other objects

你可以根据已经存在的对象来创建新的对象，这应该就是对象的继承与重载吧

4.Every object has a type

每个对象都是类的一个实例，都有自己的特定的需要其完成其功能的数据类型

5.All objects of a particular type can receive the same messages

一个特定类型的所有对象能够获取同样的消息

本书是一本与众不同的Java入门好书。作者以独特的视角，向Java初学者讲述了如何才能真正理解和掌握Java语言。本书分3篇。第1篇为Java语言基本语法，包括Java语言概述、配置开发环境、Java中的基本数据类型、Java运算符、Java流程控制语句和数组。

第2篇为Java语言高级语法，包括类、对象、方法、继承、多态、修饰符、接口、抽象类、内部类、Java异常处理和多线程编程。

第3篇为Java语言编程进阶，包括Java编程常用知识、Java文件编程、Java文件I/O编程、Java TCP编程、Java UDP编程、Java Swing编程、JDBC编程，最后还给出了一个聊天程序实例和一个数据表操作窗口实例，将Swing、UDP、Java集合类、线程同步、接口和JDBC编程有机地结合起来，展示了开发一个简单程序的典型步骤。

首选很感谢Joshua Bloch写的这本书，当然更感谢我们的司令翻译。至少目前我看到的100页的内容，基本没发现翻译很别扭的地方，包括错别字也没有发现，至少没有影响到我对书中内容的理解。再次感谢。

在后面的读书心得里，我会根据我的个人理解，将书中的各个知识点逐个和大家分享。 书中的一些观点我也不是完全赞同的，呵呵(估计是我的水平不够，呵呵呵)。

我们先看全书第一个问题：考虑用静态工厂方法代替构造器。

也就是，不再使用 new ()这种方式来获得一个类的实例，而是通过工厂方法获得。

优点：

1 有名字

这个我体会还是比较深的，在JDK里，我见过一个类的最多的构造器数量，有16个。大家可以看看 Java.math.BigDecimal 这个类就知道了。

就算一个类的构造器有4个左右，我想你很可能在使用的时候会极其注意，不要选错了到底是用哪一个。

所以，如果能提供一个很好命名的方法来实现构造类，确实是一个不错的主意，比如

BigDecimal.getInstanceFromString(...); 我想这个名字虽然有点垃圾，但要表达的意思确实异常的明确。

在一些第三方的库里面，这种情况更加突出。我许多时候不得不看看API文档，来区分我的那个int参数到底是使用三个参数的构造器，还是使用四个参数的构造器。因为他们太像了。

2 可以单例

这个就不用说了，你可以在方法里干你要做的任何事情，而new绝对会给你一个新的实例。

3 可以返回任何子类型。

这个对于系统的扩展是很有用处的。new 已经决定了你肯定要这个类，而静态方法却可以修改，不一定肯定返回你方法所在的类，可以在必要时替换为一个子类。

4 代码简便

这点我体会不深，呵呵

不过这个写法我确实经常用

List list = new ArrayList();

后面的那个ArrayList里面的 String就是一个例子吧。不过我倒是不是很在意这个，因为我很喜欢这样写了。

下面这个例子看上去确实不错。呵呵!

[java] view plain copypublic class Test extends Thread {

public static void main(String[] args) {

Test te = Test.newInstance();

}

public static Test newInstance() {

return new Test();

}

}

本文档由站牛网zhann.net收集整理，更多优质范文文档请移步zhann.net站内查找