# 高效java代码

上周作业：

package cn.gbq.example.thread.work;  
  
/\*\*  
 \* Created by gbq on 2016-11-21.  
 \*/  
  
public class Work {  
 private final static int ***COUNT*** = 10;  
  
 public static void main(String[] args) throws InterruptedException {  
 Thread t1;  
 Thread t2;  
 Thread t3;  
 for (int i = 0; i < ***COUNT***; i++) {  
 t1 = new Thread(new PrintChar(1, "A"));  
 t2 = new Thread(new PrintChar(1, "B"));  
 t3 = new Thread(new PrintChar(1, "C"));  
 t1.start();  
 t1.join();  
  
 t2.start();  
 t2.join();  
  
 t3.start();  
 t3.join();  
 }  
 }  
}

运行结果：



## 创建对象和销毁

1. 使用静态工厂创建

定义：使用静态方法，返回相应的对象。

优点：具有名称，使不可变类不必每次都创建新对象

可以返回任何子类型的对象

1. 遇到多个构造器参数要考虑使用构造器
2. 单例模式中，可以使用私有构造器或者枚举强化Singleton属性
3. 在不可实例化的类中，比如JsonUtil等工具类，使用私有构造器强化
4. 避免在循环中创建不必要的对象
5. 销毁过期的引用
6. 除非是为了作为安全网或者为了终止非关键资源的本地资源，否则不要使用finally方法

## 对于所有对象都通用的方法

1. equals

自反性，a.equals(a)

对称性，a.equals(b) = b.equals(a)