

内部文档 请勿外传



JAVA 基础

2012-09

鸿媒体
康永亮 F3-269
kangyongliang@baidu.com

Outline

- ◎ 教科书上的java
- ◎ “实战”中的java
- ◎ 衍生物

Outline

- ◎ 教科书上的java
- ◎ “实战”中的java
- ◎ 其他

教科书上的java

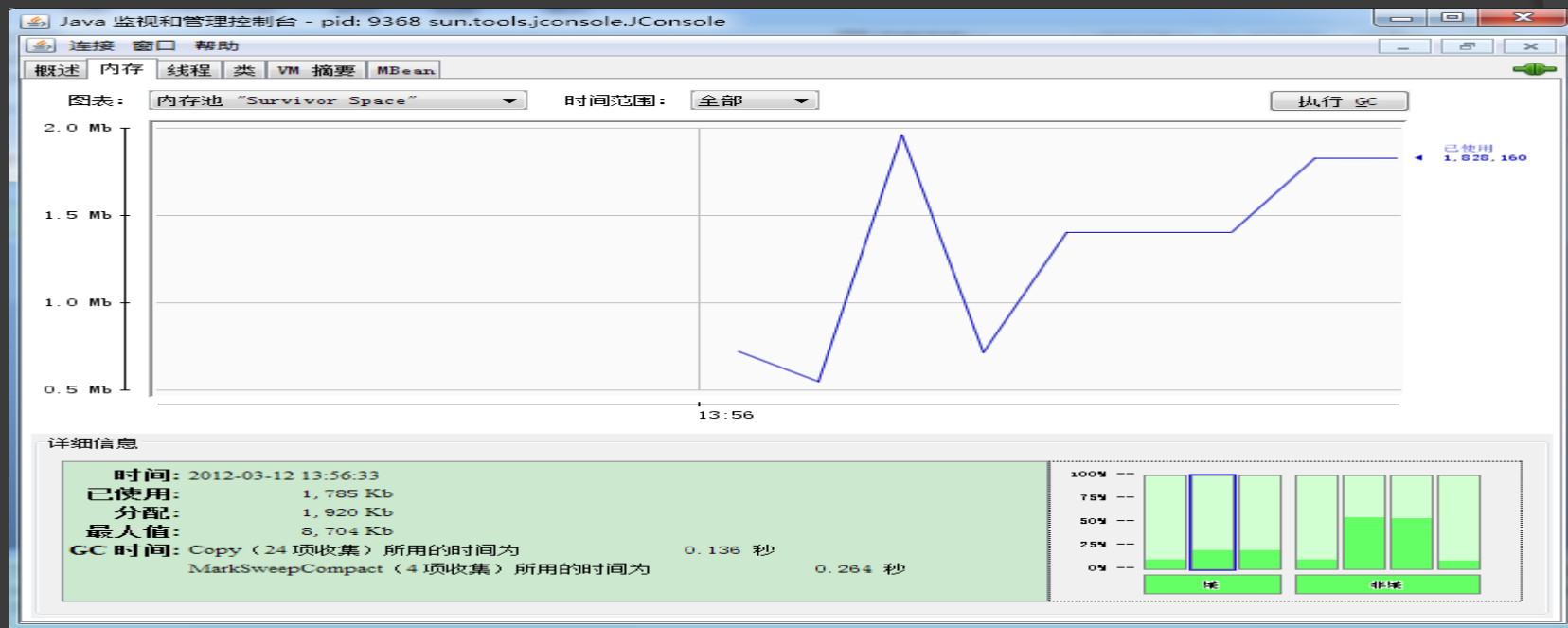
◎ JDK(jre)

- 1.4、1.5、1.6
- 1.7?
- Javac: Compile X.java
 - javac Test.java
 - java Test
 - java -jar Test.jar
 - java -cp \$classpath XXX
 - javap -c -s -l -verbose XXX
- Javadoc: generate java doc
- Appviewer: internet explorer etc. (don't use M\$'s.)

教科书上的java

◎ JDK

- Jconsole: monitor jvm etc using
-Dcom.sun.management.jmxremote.port=portNum



教科书上的java

◎ JDK

- Jconsole 用法演示
 - Monitor
 - 死锁
- Jstack 用法演示

```
"main" prio=5 tid=7fb3ba801000 nid=0x10c9b0000 waiting for monitor  
  java.lang.Thread.State: BLOCKED (on object monitor)  
    at com.baidu.bit.jsonsole.DeadLock.doIt(DeadLock.java:15)  
      - waiting to lock <7fb279200> (a java.lang.String)  
      - locked <7fb2791c8> (a java.lang.String)  
    at com.baidu.bit.jsonsole.DeadLock.main(DeadLock.java:44)  
  
Locked ownable synchronizers:  
  - None
```

- Jprofiler(\$499)

教科书上的java

◎ JDK

- 性能检测工具本身也对性能有影响，负载大的服务时该影响会非常大，**不能够忽略**
- 测试系统性能时，不要打开诸如jconsole，jprofiler等工具
- 仅在你认为程序有性能瓶颈、或者需要调优而排查的时候才使用

教科书上的java

◎ JVM

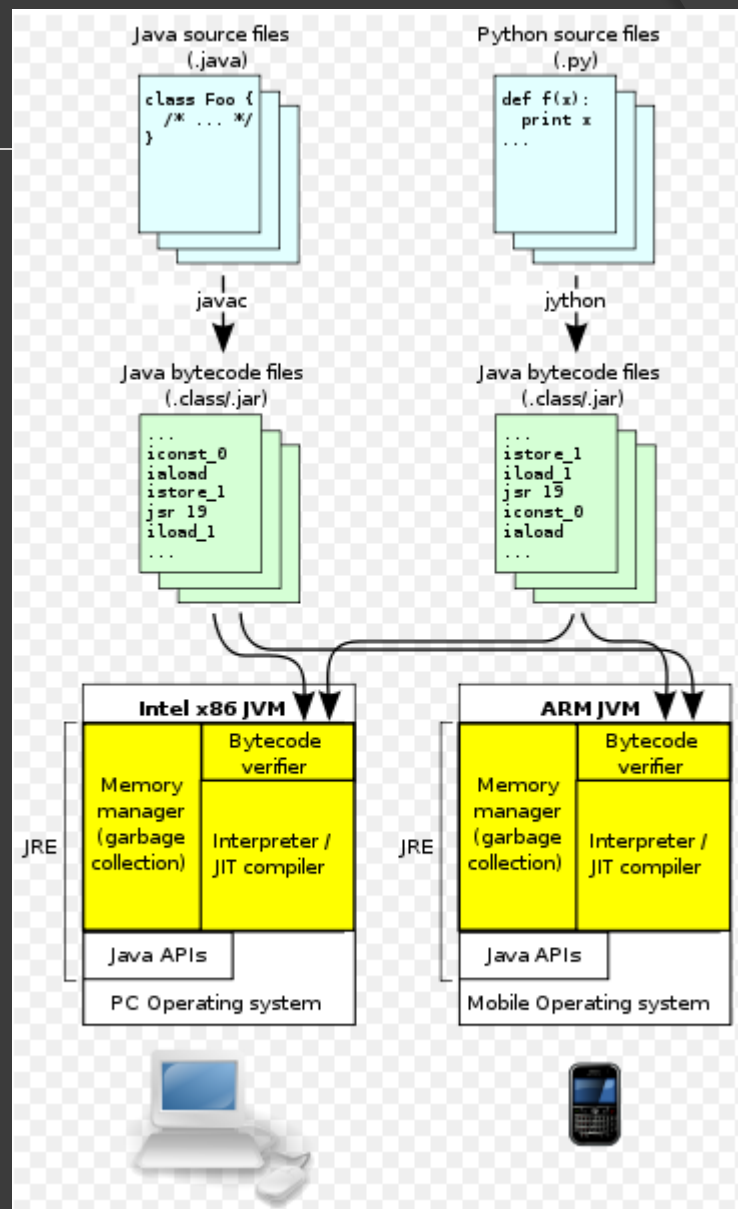
- Sun, M\$, IBM
- Dalvik (android)

Technology Compatibility Kit
Harmony

ORACLE

VS

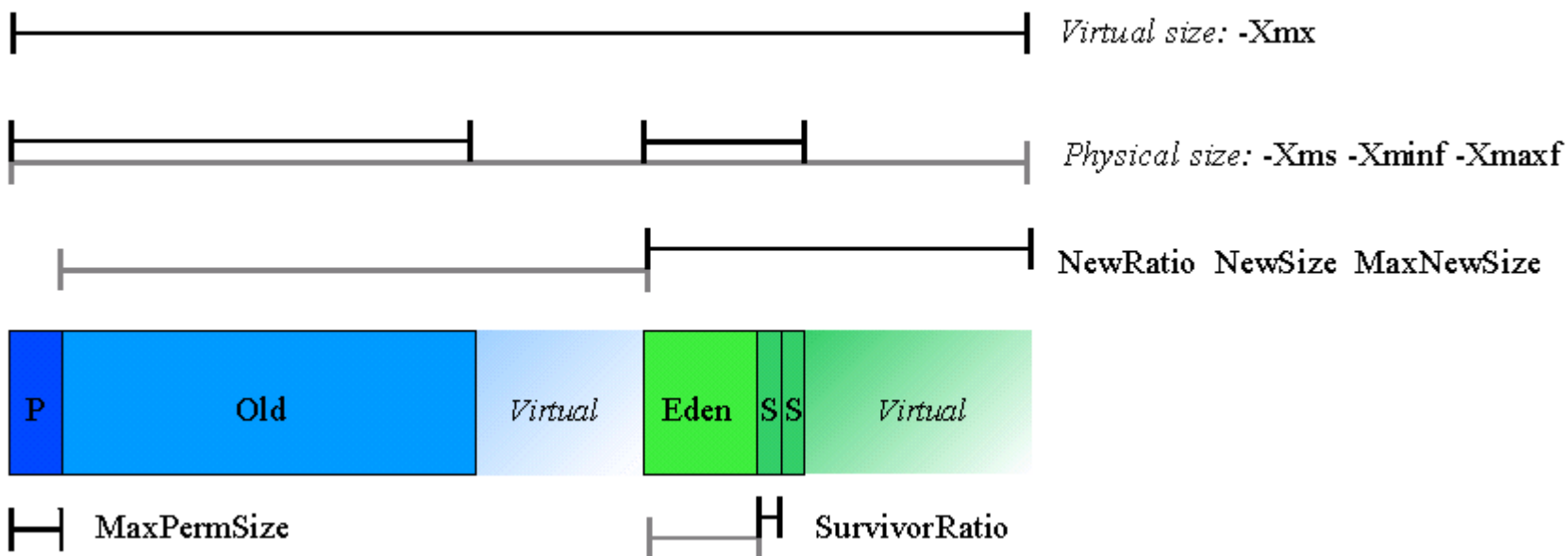
Google



教科书上的java

◎ JVM(sun/oracle hotspot)

特点：爱情不是你能买，想买就能买



Virtual是什么？

教科书上的java

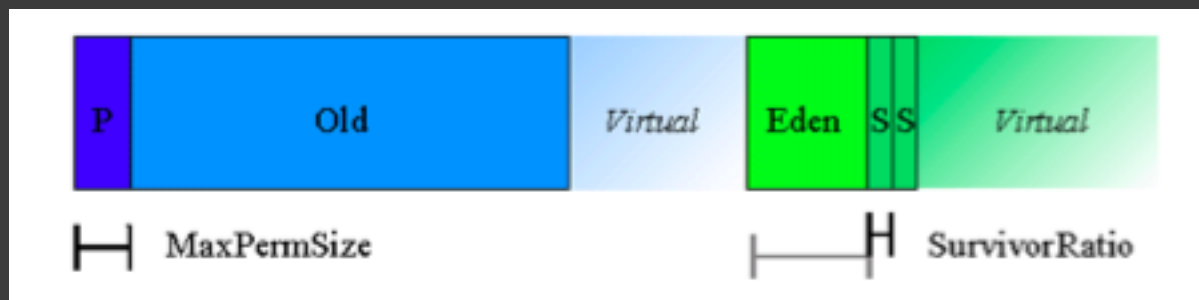
◎ JVM

permanent generation:

- Class definition
- don't need to change most of the time.
- many JSP etc
- "Caused by: [java.lang.OutOfMemoryError](#): PermGen space"

Jsp过多? 常量池溢出?

设计时没有这部分行不行?

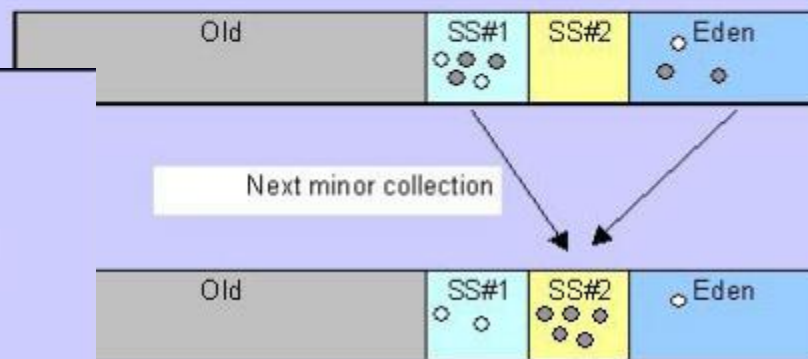
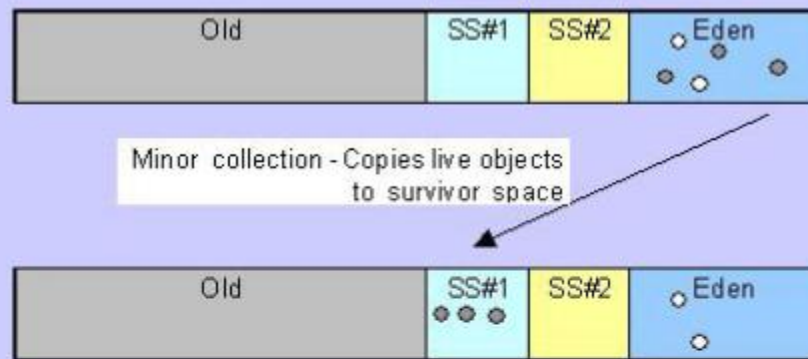
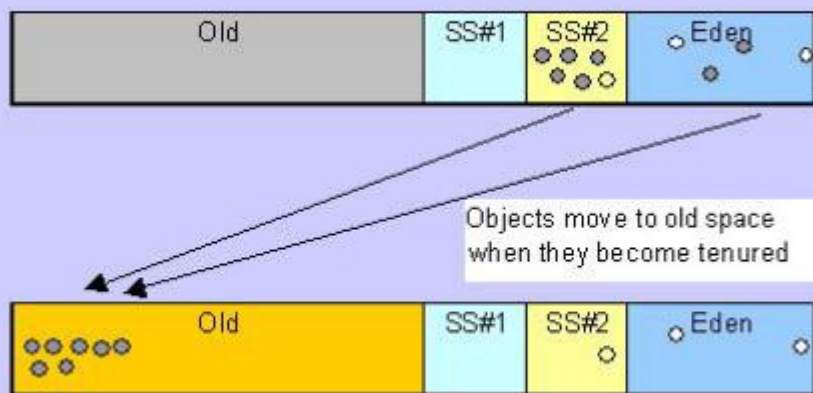


教科书上的java

◎ JVM

Young generation:

- 年轻人“好动”，挪来挪去
- New object
- Minor GC:
more memory & fast



教科书上的java

◎ JVM

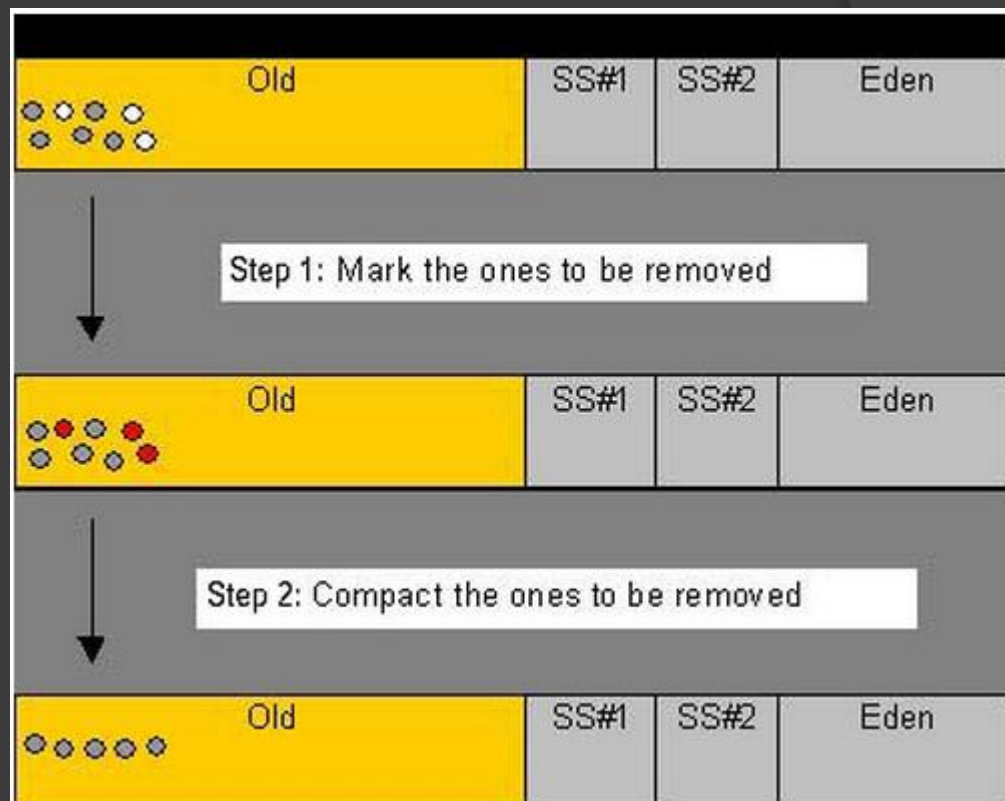
Old generation:

- 年纪大了，走不动了
- Long live object
- Full GC
- Less memory & slow
- Windows磁盘清理

Read:

<http://java.sun.com/docs/hotspot/gc/>

<http://www.javaworld.com/javaworld/jw-01-2002/jw-0111-hotspotgc.html>



教科书上的java

◎ JVM

Constant pool: perm

```
String a = "111";
```

```
String b = "111";
```

```
Assert a==b;
```

```
b = new String("111");
```

```
Assert a==b.intern();
```

Intern不是实习生是正式员工！
这东西有什么用？

教科书上的java

◎ Class file(javap)

```
CA FE BA BE 00 00 00 32 00 1B 07 00 02 01 00 21
```

Magic Number: 0xCAFEBAFE

Version of Class File Format: jdk 6 ($50=3*16+2$)

Constant Pool: Pool of constants for the class

Access Flags: for example whether the class is abstract, static, etc.

This Class: The name of the current class

Super Class: The name of the super class

Interfaces: Any interfaces in the class

Fields: Any fields in the class

Methods: Any methods in the class

Attributes: Any attributes of the class (for example the name of the sourcefile, etc.)

Outline

- ◎ 教科书上的java
- ◎ “实战”中的java
- ◎ 其他

实战中的java: 工具篇

◎ 开发工具:

普通青年:

eclipse、**myeclipse** (收费&运行卡)、**inteliJ**(收费)、**netbeans** (做界面app首选)

文艺青年:

editplus、**UE**(收费)、**notepad++**、**sublimeText**

失足青年:

jbuilder

实战中的java: 工具篇

◎ 开发工具:

eclipse常用快捷键（先改输入法设置）：

- Ctrl+1、ctrl+shift+o
- Ctrl+o、ctrl+o+o
- Ctrl+shift+G、ctrl+shift+T
- Ctrl+t
- F3、alt+左（右）
- Ctrl+shift+X（Y）
- Ctrl+shift+F（慎用）
- Debug时，ctrl+R

实战中的java: 工具篇

◎ 打包编译:

- 原始社会: ant
- 现代社会: maven
 - 依赖库管理
 - 自动下载依赖库源码
 - 全公司公用repository, 省空间、方便
- ant+ivy
 - 原始社会+现代理念

实战中的java: 工具篇

◎ 开发工具:

eclipse for JavaEE: plugin

- svn
- pydev
- easy explore
- mvn
- findbugs
- spring/hibernate/coverage

实战中的java: 工具篇

◎ hudson: CI



Hudson - Microsoft Internet Explorer

Address: http://khsuke.sfbeta.hudson/

Hudson

Hudson

ENABLE AUTO REFRESH

New Job
Configure
Reload Config

Build Queue

No.	Status
1	Idle
2	Idle
3	Building java-net-maven-mockery-jarmon #220
4	Building jackson #3183
5	Building glassfish #103
6	Idle

Build Executor Status

No.	Status
1	Idle
2	Idle
3	Building java-net-maven-mockery-jarmon #220
4	Building jackson #3183
5	Building glassfish #103
6	Idle

Hudson-working.png

Job	Last Success	Last Failure	Last Duration
Common annotations	4 days (#15)	9 months (#3)	39 seconds
bah	6 months (#11)	10 months (#2)	59 seconds
dtd-writer	5 months (#8)	N/A	1 minute
Glassfish	18 days (#388)	1 month (#367)	7 minutes
glassfish	6 days (#33)	13 days (#32)	5 minutes
glassfish	4 hours (#104)	1 day (#83)	1 hour
hudson	4 minutes (#201)	N/A	1 minute
jackson-commons	12 days (#13)	16 days (#5)	14 seconds
jackson	3 days (#22)	9 hours (#68)	1 minute
java-net-maven-mockery-jarmon	4 minutes (#10146)	10 hours (#10125)	1 minute
java-net-maven-mockery-jarmon	18 hours (#162)	N/A	0 seconds

实战中的java: 工具篇

◎ 部署:

- tomcat: 简单、方便、不要钱
不是完整的javaee容器, 不支持ejb
我们还有SSH
- Jboss: full featured javaEE server
- weblogic: bea -》 oracle
- websphere: ibm收费



实战中的java: 工具篇

◎ 测试:

- 压力测试: jmeter
- webservice测试: jmeter、soapUI (收费\社区)



实战中的java: coding篇

◎ Java coding standard

do

- 驼峰命名: baiduld, getCarName()
- 适当缩写: getSrc(), setImg()
- Map<String,String>, Set<Object>
- 使用IDE Format代码时大家统一模板
- Long型值使用1000L而不是1000I

实战中的java: coding篇

◎ Java coding standard

Don't

- 变量命名中英文结合getMyShiPin()
- magic number
- 使用和jdk中已有类同名的类: String

```
Map map = new HashMap();  
map.put("12", "强生");  
map.put("12", "强生");  
map.put(12, "强生");
```

实战中的java: coding篇

◎ 资源记得释放

- 文件

```
try{  
    打开文件流  
}catch(Exception){  
    //log  
} finally{  
    //关闭文件流  
}
```

实战中的java: coding篇

◎ 资源记得释放

● 数据库连接

```
Connection conn;
Statement stmt;
try{
    conn = DbUtil.getConnection();
    foreach(String sql : sqlList){
        stmt = conn.prepareStatement(sql); //FIXME
    }
}catch(Exception e){
}finally{
    cleanUp(conn,stmt);
}
```

实战中的java: coding篇

◎ 资源记得释放

- 其他类似资源归还到pool中
- Threadlocal线程池环境记得清空

```
*/  
public class ManagedSessionContext implements CurrentSessionContext {  
  
    private static final ThreadLocal context = new ThreadLocal();  
    private final SessionFactoryImplementor factory;  
  
    public ManagedSessionContext(SessionFactoryImplementor factory) {  
        this.factory = factory;  
    }  
}
```

实战中的java: coding篇

◎ singleton

- 单例作用？
- 怎么写？

实战中的java: coding篇

◎ singleton: 普通青年

```
public class Singleton {  
    private static final Singleton instance = new Singleton();  
  
    // Private constructor prevents instantiation from other classes  
    private Singleton() {}  
  
    public static Singleton getInstance() {  
        return instance;  
    }  
}
```

实战中的java: coding篇

◎ singleton: 普通青年 lazy-load

```
public static synchronized Singleton getInstance()
{
    if (instance == null)           //1
        instance = new Singleton(); //2
    return instance;                 //3
}
```

实战中的java: coding篇

◎ singleton: 失足青年

```
public static Singleton getInstance()
{
    if (instance == null)           //1
        instance = new Singleton(); //2
    return instance;                 //3
}
```

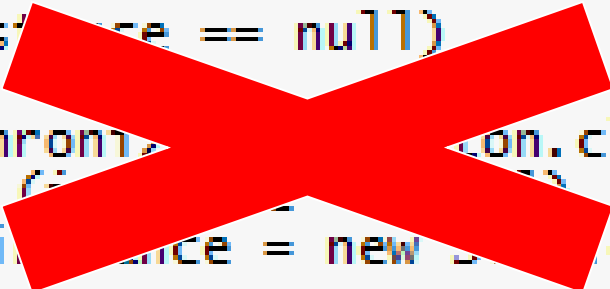
```
public static Singleton getInstance()
{
    if (instance == null)
    {
        synchronized(Singleton.class) {
            instance = new Singleton();
        }
    }
    return instance;
}
```

实战中的java: coding篇

◎ singleton: lazy-load “文艺型”

Double-checked locking

```
public static Singleton getInstance()
{
    if (instance == null)
    {
        synchronized (Singleton.class) { //1
            if (instance == null) //2
                instance = new Singleton(); //3
        }
    }
    return instance;
}
```



实战中的java: coding篇

◎ singleton: lazy-load “文艺型”

Double-checked locking

```
mem = allocate();           //Allocate memory for Singleton object.  
instance = mem;             //Note that instance is now non-null, but  
                             //has not been initialized.  
ctorSingleton(instance);    //Invoke constructor for Singleton passing  
                             //instance.
```

实战中的java: coding篇

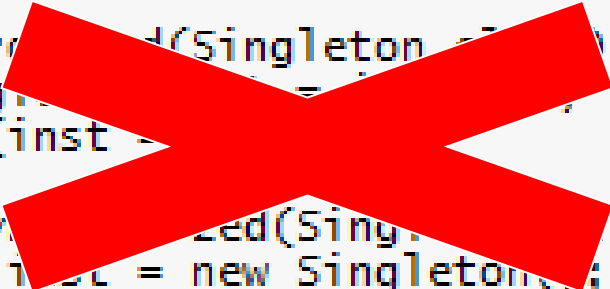
◎ singleton: lazy-load “文艺型” 改进版 Double-checked locking

```
public static Singleton getInstance()
{
    if (instance == null)
    {
        synchronized(Singleton.class) {           //1
            Singleton inst = instance;               //2
            if (inst == null)
            {
                synchronized(Singleton.class) {     //3
                    inst = new Singleton();           //4
                }
                instance = inst;                       //5
            }
        }
    }
    return instance;
}
```

实战中的java: coding篇

◎ singleton: lazy-load “文艺型” 改进版 Double-checked locking

```
public static Singleton getInstance()
{
    if (instance == null)
    {
        synchronized(Singleton.class) { //1
            Singleton inst = new Singleton(); //2
            if (inst == null) { //3
                synchronized(Singleton.class) { //3
                    inst = new Singleton(); //4
                }
            }
            instance = inst; //5
        }
    }
    return instance;
}
```



实战中的java: coding篇

◎ singleton: 文艺型

```
public class Singleton {  
    // Private constructor prevents instantiation from other classes  
    private Singleton() {}  
  
    /**  
     * SingletonHolder is loaded on the first execution of Singleton.getInstance()  
     * or the first access to SingletonHolder.INSTANCE, not before.  
     */  
    private static class SingletonHolder {  
        public static final Singleton instance = new Singleton();  
    }  
  
    public static Singleton getInstance() {  
        return SingletonHolder.instance;  
    }  
}
```

实战中的java: coding篇

◎ singleton: 文艺型2 (effective java)

```
public enum Singleton {  
    INSTANCE;  
    public void execute (String arg) {  
        //... perform operation here ...  
    }  
}
```

实战中的java: coding篇

◎ compareTo

```
public int compareTo(Integer value){  
    return this.value - value;  
}
```

◎ 二分查找

实战中的java: coding篇

◎ HashMap: 最常用的类之一

- 多线程同步?
- 基本实现?
- 多线程下文艺青年用什么?

<http://www.ibm.com/developerworks/cn/java/java-lo-concurrenthashmap/index.html>

实战中的java: coding篇

◎ Serialization

- 有什么用?
- 这个warning干什么的?

The serializable class Ad does not declare a static final serialVersionUID field of type long

实战中的java: coding篇

- ◎ 我需要看jdk源代码么？
 - 参加了apache harmony？
 - Collections.sort好用么？
 - 源代码从头看到尾？

<http://www.ibm.com/developerworks/cn/java/java-lo-concurrenthashmap/index.html>

实战中的java: coding篇

◎ 开源项目源代码？

- Nio: mina
- ORmapping: ibatis
- Struts2
- tomcat

实战中的java: coding篇

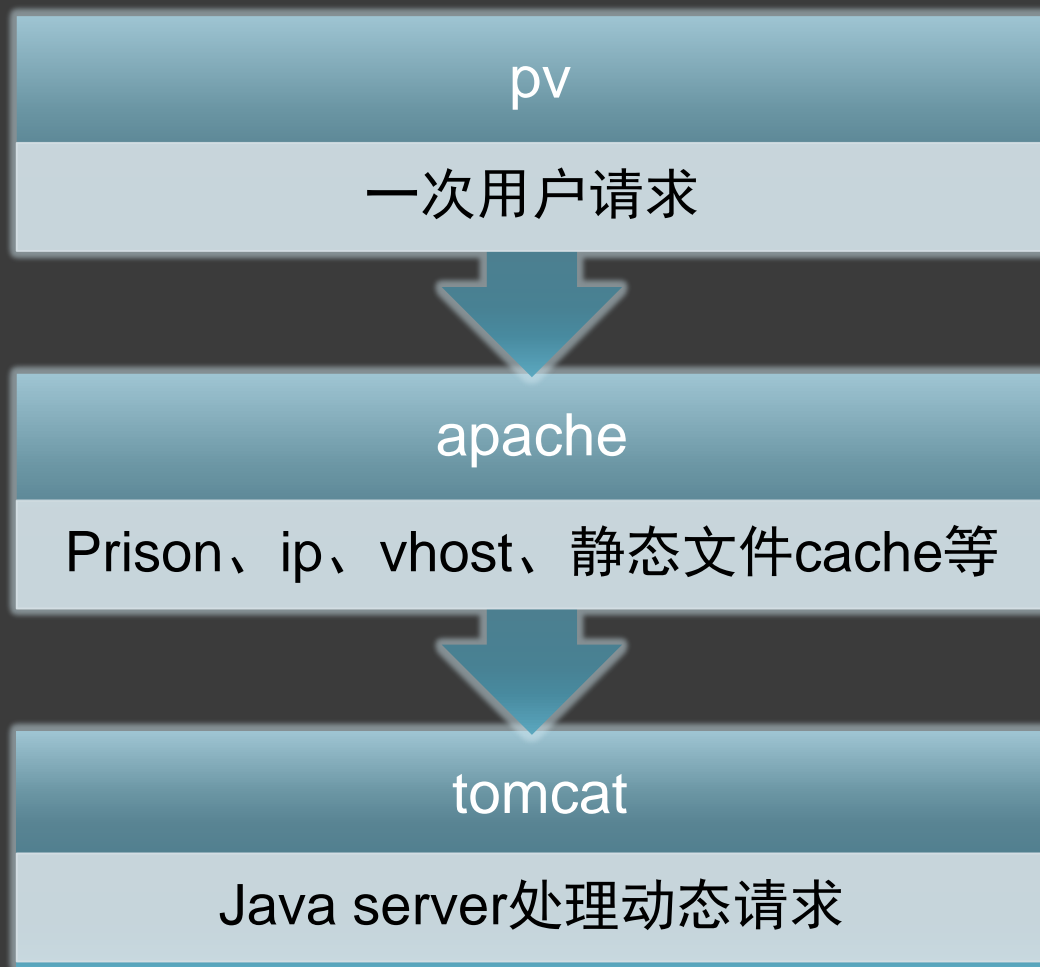
◎ Awt/swing?

- Swing功能增强、可移植性
- 一般来说知道有这个东西就行了

Outline

- ◎ 教科书上的java
- ◎ “实战”中的java
- ◎ 其他

其他：tomcat & apache



其他: tomcat & apache

- ◎ apache如何转发请求给tomcat?
 - mod_jk

```
JKMount /seek/* ajp13w
JKMount /tec_display/* ajp13w
JKMount /flash/* ajp13w
JkMount /login* ajp13w
    JkMount /logout* ajp13w
    JkMount /verifyCode* ajp13w
    JKMount /verifyCode* ajp13w
</IfModule>
```

其他: tomcat & apache

◎ apache如何根据不同域名配置不同服务

<http://show.hongmeiti.com>

<http://www.hongmeiti.com>

```
mubing:Desktop mubing$ ping www.hongmeiti.com
PING www.hongmeiti.com (220.181.112.159): 56 data bytes
64 bytes from 220.181.112.159: icmp_seq=0 ttl=49 time=93.286 ms
64 bytes from 220.181.112.159: icmp_seq=1 ttl=49 time=94.592 ms
^C
--- www.hongmeiti.com ping statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 93.286/93.939/94.592/0.653 ms
mubing:Desktop mubing$ ping show.hongmeiti.com
PING hongmeiti.com (220.181.112.159): 56 data bytes
64 bytes from 220.181.112.159: icmp_seq=0 ttl=49 time=103.616 ms
64 bytes from 220.181.112.159: icmp_seq=1 ttl=49 time=84.114 ms
64 bytes from 220.181.112.159: icmp_seq=2 ttl=49 time=94.629 ms
^C
```

其他: tomcat & apache

```
<VirtualHost *>
    DocumentRoot "/home/work/hongmeiti/apache/htdocs.hmt"
    ServerName www.hongmeiti.com
    ErrorLog "| /home/work/hongmeiti/apache/bin/cronolog /home/work/hongmeiti/apache/logs/error_log"
    CustomLog "| /home/work/hongmeiti/apache/bin/cronolog /home/work/hongmeiti/apache/logs/access_log" common
    Redirect 301 / http://hong.baidu.com/
</VirtualHost>

<VirtualHost *>
    DocumentRoot "/home/work/hongmeiti/apache/htdocs.hmt"
    ServerName hong.baidu.com
    ErrorLog "| /home/work/hongmeiti/apache/bin/cronolog /home/work/hongmeiti/apache/logs/error_log"
    CustomLog "| /home/work/hongmeiti/apache/bin/cronolog /home/work/hongmeiti/apache/logs/access_log" common
</VirtualHost>

<VirtualHost *>
    DocumentRoot "/home/work/hongmeiti/apache/htdocs"
    ServerName show.hongmeiti.com
    ErrorLog "| /home/work/hongmeiti/apache/bin/cronolog /home/work/hongmeiti/apache/logs/error_log"
    CustomLog "| /home/work/hongmeiti/apache/bin/cronolog /home/work/hongmeiti/apache/logs/access_log" common
```



其他： 设计模式

◎ Decorator: java.io

- 现实中：
 - 各种报表模块
 - 不同角色视图模块

◎ 依赖注入 spring

- 现实中：
 - 思考一下如何实现这个功能

其他：知识领域

◎ Java的cache系统？

- Ehcache
- Redis java client
- mangoDB java client
- BDB
- Etc

◎ 能自己实现一个么？

- 我是搞java的

其他：知识领域

HttpRequest

Web

Server

DB

storage

Discussion

kangyongliang@baidu.com

Hi: Peter_ylkang

