

Java程序设计

Java Programming Design
毛斐巧

感谢：教材《Java大学实用教程》的作者和其他老师提供PowerPoint讲义等资料！

说明：本课程所使用的所有讲义，都是在以上资料上修改的。

Java语言概述

主要内容

- 1.1 Java语言的诞生
- 1.2 学习Java的必要性
- 1.3 Java的特点
- 1.4 Java与C/C++之关系
- 1.5 Java运行平台
- 1.6 Java程序开发
- 1.7 JDK 1.6编译器的兼容性

1.1 Java语言的诞生

- Java诞生于1995年（注：1996发布Java 1.0），是Sun Microsystems公司组织开发的编程语言，主要贡献者是James Gosling。
- 1990年Sun Microsystems公司成立了由James Gosling领导的开发小组，开始致力于开发一种可移植的、跨平台的语言，该语言能生成正确运行于各种操作系统、各种CPU芯片上的代码。
- 他们的钻研和努力促成了Java语言的诞生。
- Java的发展得益于Internet和Web的出现，Internet上有各种不同的计算机，它们可能使用完全不同的操作系统和CPU芯片，但仍希望运行相同的程序，Java的出现标志着真正的分布式系统的到来。

主要内容

- 1.1 Java语言的诞生
- 1.2 学习Java的必要性
- 1.3 Java的特点
- 1.4 Java与C/C++之关系
- 1.5 Java运行平台
- 1.6 Java程序开发
- 1.7 JDK 1.6编译器的兼容性

1.2 学习Java的必要性

- Java不仅可以用来开发大型的桌面应用程序，而且特别适合于Internet的应用开发。
- 目前，Java语言不仅是一门正在被广泛使用的编程语言，而且已成为软件设计开发者应当掌握的一门基础语言。
- Java语言是面向对象编程的，并涉及到网络、多线程等重要的基础知识，而且很多新的技术领域都涉及到了Java语言，因此，学习和掌握Java已成为共识。国内外许多大学已将Java语言列入了本科教学计划；IT行业对Java人才的需求正在不断的增长，一些软件公司对其开发人员周期性地进行Java的基础培训工作。
- 在编程语言社区(TIOBE)中使用热度排第一（2018年数据）

主要内容

- 1.1 Java语言的诞生
- 1.2 学习Java的必要性
- 1.3 Java的特点
- 1.4 Java与C/C++之关系
- 1.5 Java运行平台
- 1.6 Java程序开发
- 1.7 JDK 1.6编译器的兼容性

1.3 Java的特点

- 简单易学
 - 风格类似于C++, 基本语法与C语言类似；摒弃了C++中容易引发程序错误的地方，如指针和内存管理；提供了丰富的类库。
- 平台无关性
 - Java源程序被编译成一种高层次的与机器无关的字节码（bytecode）格式语言，被设计在Java虚拟机上运行。Java语言和其他语言相比，最大的优势就是能够在所有的计算机上运行，这也是Java风靡全球的主要原因。“write once, run anywhere” (WORA)。

1.3 Java的特点

- 面向对象
 - 面向对象编程是一种先进的编程思想，更加容易解决复杂的问题。支持静态和动态风格的代码继承及重用，是完全面向对象的，它不支持类似C语言那样的面向过程的程序设计技术。
- 多线程
 - Java提供的多线程功能使得在一个程序里可同时执行多个小任务。多线程带来的好处是可以有更好的交互性能和实时控制性能。

1.3 Java的特点

- 可移植
 - Java应用程序可以在配备了Java解释器和Java运行环境（JRE）的任何计算机系统上运行，通过定义独立于平台的基本数据类型及其运算，Java数据得以在任何硬件平台上保持一致。
- 健壮
 - Java致力于检查程序在编译和运行时的错误。类型检查帮助检查出许多开发早期出现的错误。Java自己操纵内存，减少了内存出错的可能性。

1.3 Java的特点

- 安全
 - Java的安全性可从两个方面得到保证。1) 在Java语言里，**指针和释放内存**等C++功能被删除，避免了非法内存操作。2) 当Java用来创建浏览器应用时，语言功能和一些浏览器本身提供的功能结合起来，使它更安全。
- 动态
 - Java的动态特性是其面向对象设计方法的拓展，它允许程序动态地装入**运行过程中所需要的类**。

主要内容

- 1.1 Java语言的诞生
- 1.2 学习Java的必要性
- 1.3 Java的特点
- 1.4 Java与C/C++之关系
- 1.5 Java运行平台
- 1.6 Java程序开发
- 1.7 JDK 1.6编译器的兼容性

1.4 Java与C/C++之关系

- Java从C语言和C++语言继承了许多成分，Java是由C/C++语言发展和衍生的产物。比如，Java语言在**变量声明**、**操作符形式**、**参数传递**、**流程控制**等方面和C/C++语言相通。
- 但是，Java和C/C++语言又有许多**差别**。

1.4 Java与C/C++之关系

- Java中对内存的分配是动态的，它采用面向对象的机制，采用new为每个对象分配内存空间，而且内存还会随程序运行情况而改变。
- C语言是通过malloc()和free()这两个库函数分别实现分配内存和释放内存空间。
- C++语言中则通过new和delete来分配内存及释放内存空间。

1.4 Java与C/C++之关系

- Java不在所有类之外定义全局变量，而是在某个类中定义一种公用静态变量来完成全局变量的功能。比如：

```
class GlobalVar
{
    public static global_var;
    ...
}
```

在GlobalVar这个类中，定义了一个公用静态变量global_var，其他类可以访问或修改这个变量，所以，公用静态变量起到了全局变量的作用，但前者进行了较好的封装。

1.4 Java与C/C++之关系

- Java不用**goto语句**，而用try-catch-finally异常处理语句来代替**goto**与处理出错的功能。
- Java**不支持头文件**，而C/C++语言中都用头文件来定义类的原型、全局变量、库函数等。
- Java**不支持宏定义**，而用关键字**final**来定义**常量**，在C++中则采用宏定义来实现常量定义。

1.4 Java与C/C++之关系

- Java对每种数据类型都分配**固定长度**。例如，在Java中，int总是32位的，而在C/C++中，对于不同的平台，同一个数据类型分配不同的字节数，同样是int，在16位PC机中为2个字节，而在VAX-11中，则为32位。这使C/C++语言造成不可移植性，而Java则具有跨平台性。
- Java**不用指针**，从而不存在程序员对指针进行编程的问题，也**不允许通过指针来分配或释放某个内存空间**。而在C/C++中，常用指针来对内存地址进行**灵活**的操作，但这种操作也非常容易造成**不可预测的错误**。

主要内容

- 1.1 Java语言的诞生
- 1.2 学习Java的必要性
- 1.3 Java的特点
- 1.4 Java与C/C++之关系
- 1.5 Java运行平台
- 1.6 Java程序开发
- 1.7 JDK 1.6编译器的兼容性

1.5 Java运行平台

- **1.三种平台简介**
- 目前Java运行平台主要分为3个版本
 - **Java SE (Java Standard Edition)** 标准版（曾称J2SE），是桌面开发和低端商务应用的解决方案。
 - **Java EE (Java Enterprise Edition)** 企业版（曾称J2EE），是以**企业**为环境而开发应用程序的解决方案。
 - **Java ME (Java Micro Edition)** 小型版（曾称J2ME），是致力于消费产品和**嵌入式**设备的最佳解决方案。

1.5 Java运行平台

- **2. 安装Java SE平台**
- 学习Java必须从Java SE开始，基于Java SE来学习Java。
- 安装Java开发包（Java 2 Software Development Kit，JDK）是进行Java软件开发的前提。以下安装过程以JDK 8u144为例。软件可以在Oracle公司的官方网站下载到。

Step 1

<http://www.oracle.com/technetwork/java/index.html>

Step 2



Step 3



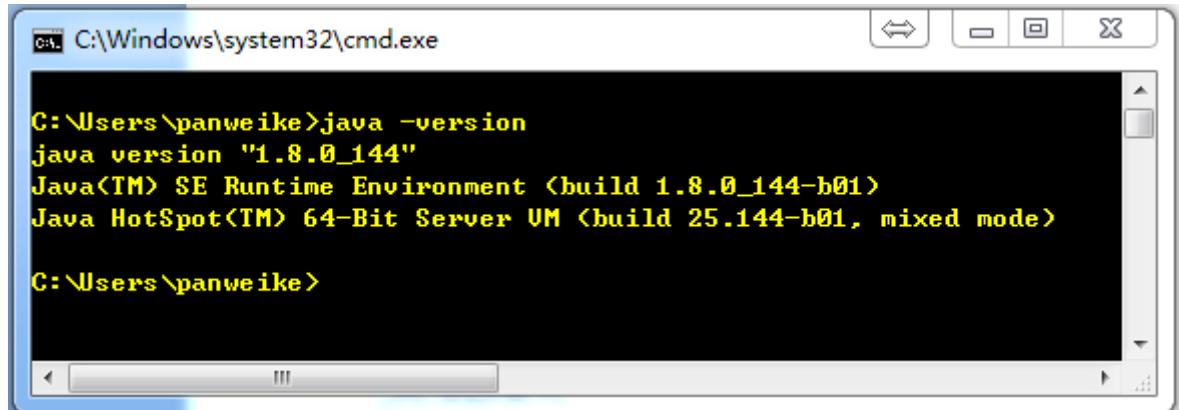
Step 4

Accept License Agreement

Step 5: 下载 Windows x86 or Windows x64

Java SE Development Kit 8u144		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.89 MB	jdk-8u144-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.83 MB	jdk-8u144-linux-arm64-vfp-hflt.tar.gz
Linux x86	164.65 MB	jdk-8u144-linux-i586.rpm
Linux x86	179.44 MB	jdk-8u144-linux-i586.tar.gz
Linux x64	162.1 MB	jdk-8u144-linux-x64.rpm
Linux x64	176.92 MB	jdk-8u144-linux-x64.tar.gz
Mac OS X	226.6 MB	jdk-8u144-macosx-x64.dmg
Solaris SPARC 64-bit	139.87 MB	jdk-8u144-solaris-sparcv9.Z
Solaris SPARC 64-bit	99.18 MB	jdk-8u144-solaris-sparcv9.tar.gz
Solaris x64	140.51 MB	jdk-8u144-solaris-x64.tar.Z
Solaris x64	96.99 MB	jdk-8u144-solaris-x64.tar.gz
Windows x86	190.94 MB	jdk-8u144-windows-i586.exe
Windows x64	197.78 MB	jdk-8u144-windows-x64.exe

1.5 Java运行平台



- 安装JDK
 - 安装: jdk-8u144-windows-i586.exe 或 jdk-8u144-windows-x64.exe
 - 设置: 计算机->属性->高级系统设置->环境变量->系统变量:
 - 新建**JAVA_HOME**设为[C:\Program Files\Java\jdk1.8.0_144](#)
 - 在**Path**系统变量后添加 ;[C:\Program Files\Java\jdk1.8.0_144\bin](#); [C:\Program Files\Java\jdk1.8.0_144\jre\bin](#)
 - 在**CLASSPATH**系统变量后添加 .;[C:\Program Files\Java\jdk1.8.0_144\lib\dt.jar](#);[C:\Program Files\Java\jdk1.8.0_144\lib\tools.jar](#)
 - 测试: 开始->输入CMD打开命令行模式->输入java -version, 如果显示版本号[1.8.0_144](#), 表示安装正确

1.5 Java运行平台

- JDK vs. JRE
 - JRE (Java Runtime Environment): **run** Java applications
 - JVM
 - Java类库
 - 一些支持文件
 - JDK (Java Development Kit): **develop** Java applications
 - 下载的JDK中包含了JRE，所以只要下载JDK就可以了

Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

1.5 Java运行平台

- Eclipse集成开发环境
 - Step 1. <http://www.eclipse.org/downloads/>



- Step 2. 点击Download Packages



- Step 3.

注：除了Eclipse Oxygen版本还有Eclipse Neon, Eclipse Mars, Eclipse Luna, Eclipse Kepler等

主要内容

- 1.1 Java语言的诞生
- 1.2 学习Java的必要性
- 1.3 Java的特点
- 1.4 Java与C/C++之关系
- 1.5 Java运行平台
- 1.6 Java程序开发
- 1.7 JDK 1.6编译器的兼容性

1.6 Java程序开发

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** eclipse-workspace - JavaPD/src/Welcome.java - Eclipse
- Menu Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help
- Toolbar:** Includes icons for New, Open, Save, Cut, Copy, Paste, Find, Replace, and Run.
- Package Explorer:** Shows the project structure:
 - JavaPD
 - JRE System Library [jre]
 - src
 - (default package)
 - Welcome.java
- Editor:** Displays the code for `Welcome.java`:

```
1 public class Welcome{
2
3     public static void main(String[] args){
4
5         String[] greeting = new String[3];
6         greeting[0] = "Welcome to Core Java";
7         greeting[1] = "by Cay Horstmann";
8         greeting[2] = "and Gary Cornell";
9
10    for (String g : greeting){
11        System.out.println(g);
12    }
13 }
14
15 }
```
- Console:** Shows the output of the program:

```
<terminated> Welcome [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (2017年9月5日 上午11:14:26)
Welcome to Core Java
by Cay Horstmann
and Gary Cornell
```
- Status Bar:** Writable, Smart Insert, 17 : 1

1.6 Java程序开发

```
public class Welcome{  
    public static void main(String[] args){  
        String[] greeting = new String[3];  
        greeting[0] = "Welcome to Core Java";  
        greeting[1] = "by Cay Horstmann";  
        greeting[2] = "and Gary Cornell";  
  
        for (String g : greeting){  
            System.out.println(g);  
        }  
    }  
}
```

must be public

must be public and static

class name = file name

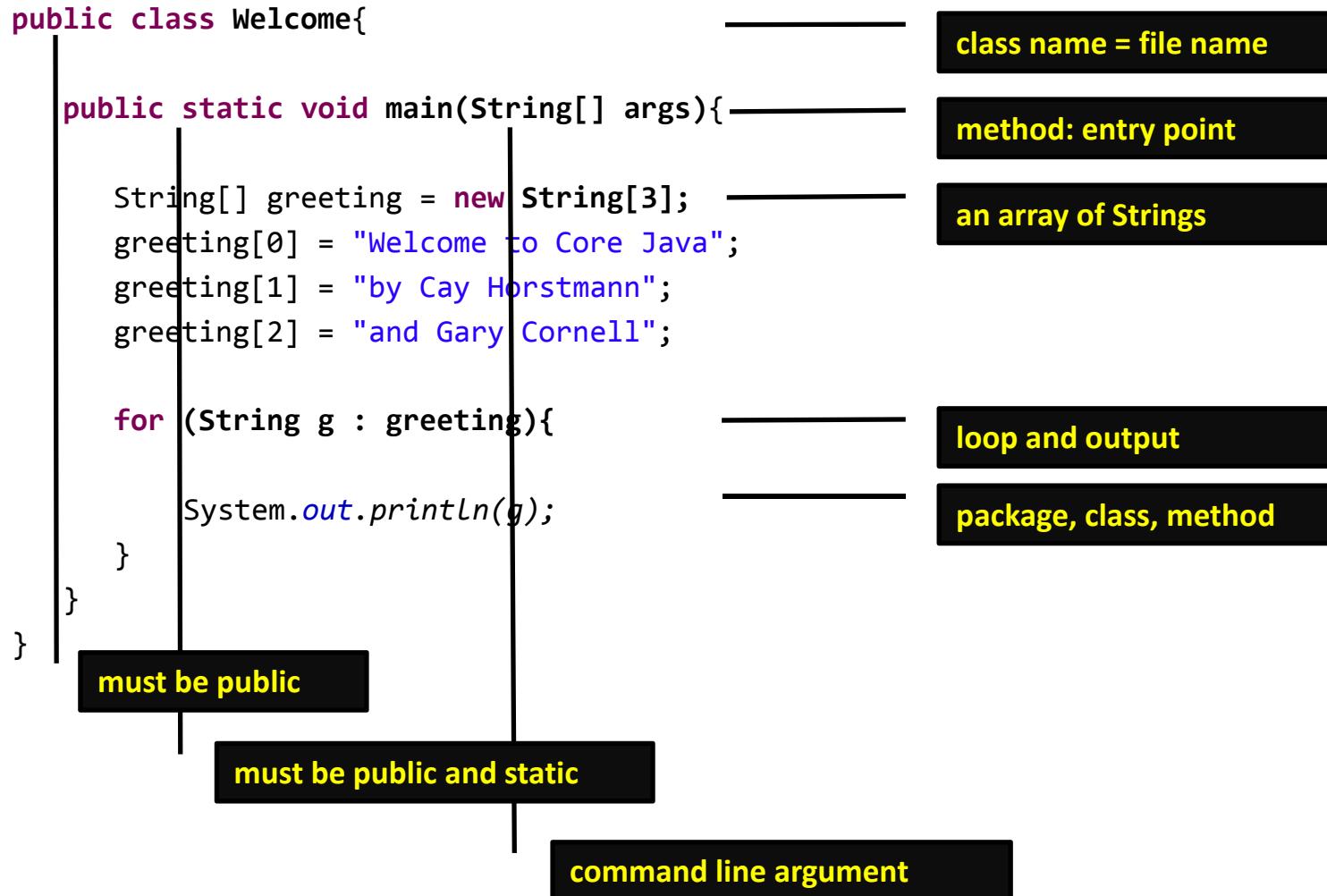
method: entry point

an array of Strings

loop and output

package, class, method

command line argument



1.6 Java程序开发

- 1.编写源文件
 - **Hello.java**
- 2.编译源文件
 - **javac Hello.java**
- 生成:
A.class
B.class
Hello.class
- 3.运行Java应用程序
 - **java Hello**

```
class A
{
    void f()
    {
        System.out.println("I am A");
    }
}
class B
{
}
public class Hello
{
    public static void main (String args[ ])
    {
        System.out.println("你好， 很高兴学习Java");
        A a=new A();
        a.f();
    }
}
```

注：每个文件只能包含一个**public**类

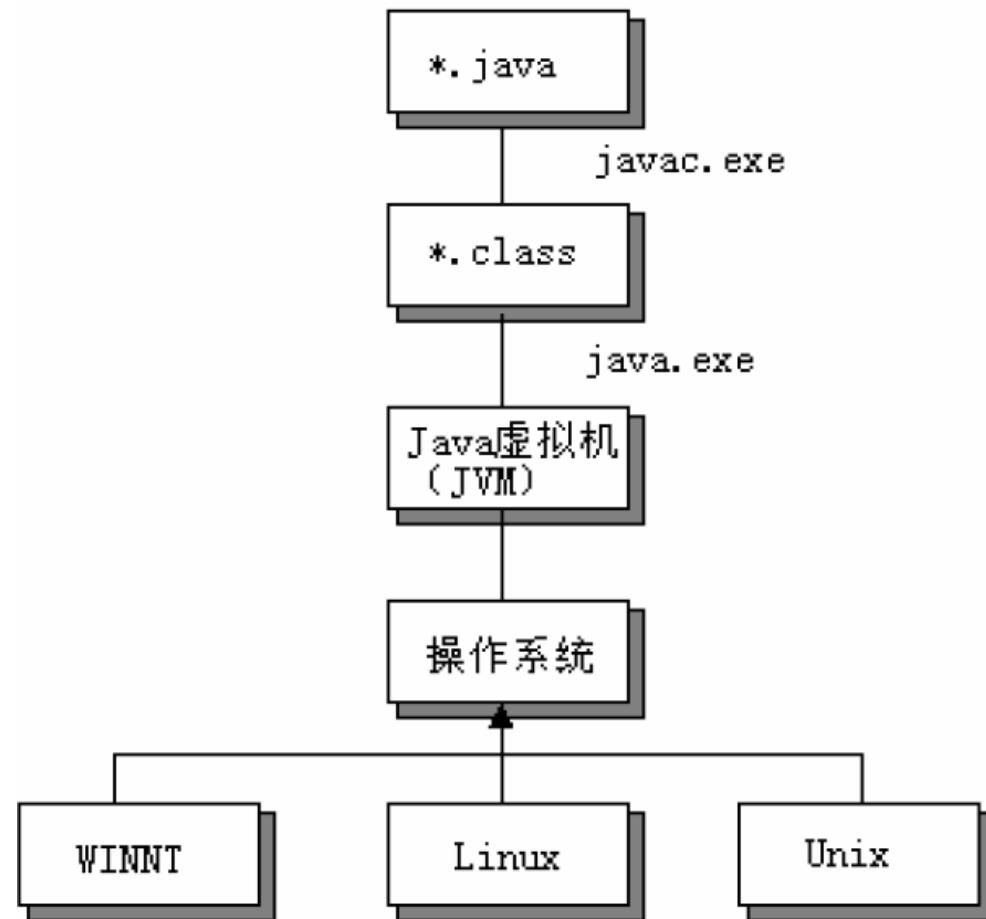
1.6 Java程序开发

- 1.编写源文件
 - **Tom.java**
- 2.编译源文件
 - **javac Tom.java**
- 生成: Tom.class
Example.class
- 3.运行Java应用程序
 - **java Example**

```
public class Tom{  
    int leg;  
    String head;  
    void cry(String s){  
        System.out.println(s);  
    }  
}  
class Example{  
    public static void main (String args[]){  
        Tom cat;  
        cat=new Tom();  
        cat.leg=4;  
        cat.head="猫头";  
        System.out.println("腿:"+cat.leg+"条");  
        System.out.println("头:"+cat.head);  
        cat.cry("我今天要和Jerry拼了");  
    }  
}
```

注：每个文件只能包含一个**public**类

1.6 Java程序开发



主要内容

- 1.1 Java语言的诞生
- 1.2 学习Java的必要性
- 1.3 Java的特点
- 1.4 Java与C/C++之关系
- 1.5 Java运行平台
- 1.6 Java程序开发
- 1.7 JDK 1.6编译器的兼容性

1.7 JDK 1.6编译器的兼容性

- 用JDK 1.6的编译器生成的字节码只能在安装了高于JDK 1.6或JRE 1.6的Java平台环境中运行。
- 可以使用 `-source` 参数约定字节码（`bytecode`）适合的Java平台，例如：

```
javac -source 1.2 Hello.java
```

小节

- 1.1 Java语言的诞生
- 1.2 学习Java的必要性
- 1.3 Java的特点
- 1.4 Java与C/C++之关系
- 1.5 Java运行平台
- 1.6 Java程序开发
- 1.7 JDK 1.6编译器的兼容性

} 重点

补充内容

Java风云人物

- James Gosling (May 19, 1955 -)
- Java之父
- http://en.wikipedia.org/wiki/James_Gosling



- 12岁做了一台电子游戏机，14岁学习计算机编程。80年代初，James Gosling获得博士学位后到IBM工作，设计IBM第一代工作站，后跳槽到Sun Microsystems，1991年开始从事Green项目，随后创造了Java。

补充内容

Java风云人物

- Joshua Bloch (August 28, 1961 -)
- Java之母，Java 2元勋
- http://en.wikipedia.org/wiki/Joshua_Bloch



- 将Java“哺育”成长，主要贡献：JDK 1.1 中的java.math、JDK 1.4中的assertions、Collections Framework、Tiger，2004年7月离开Sun Microsystems到Google公司。

补充内容

Java风云人物

- Bill Joy (November 8, 1954 -)
- http://en.wikipedia.org/wiki/Bill_Joy
- 兼通计算机软硬件技术的天才，Sun Microsystems的创始人之一，世界上最优秀的程序员之一。



补充内容

Java风云人物

- Scott McNealy (November 13, 1954 -)
- http://en.wikipedia.org/wiki/Scott_McNealy
- Sun Microsystems多年的掌舵者。



补充内容

Java风云人物

- Doug Lea
- http://en.wikipedia.org/wiki/Doug_Lea
- 世界上对Java影响力最大的个人，因为在Java的两次大变革中，他都起了举足轻重的作用，服务于纽约州立大学Oswego分校计算机科学系。



补充内容

Java风云人物

- Bruce Eckel (July 8, 1957 -)
- http://en.wikipedia.org/wiki/Bruce_Eckel
- *Thinking in Java*的作者。



补充内容

Java常用词汇

- JVM, Java Virtual Machine (Java虚拟机)
- JDK, Java Development Kit (Java开发工具包)
- JRE, Java Runtime Environment (Java运行环境)
- IDE, Integrated Development Environment (集成开发环境)