

Java 应用与开发

Java 技术概述及开发环境

王晓东

wangxiaodong@ouc.edu.cn

中国海洋大学

September 18, 2018



参考书目

1. 陈国君等编著, Java 程序设计基础 (第 5 版), 清华大学出版社
2. Bruce Eckel, Thinking in Java (3rd)



本章学习目标

1. 了解 Java 的发展历程
2. 理解 Java 平台的相关概念和机制
3. 掌握基本 Java 开发环境配置



大纲

Java 技术概述

Java 平台核心机制

Java 开发环境

Java 基本开发流程



接下来...

Java 技术概述

Java 平台核心机制

Java 开发环境

Java 基本开发流程



那些伟大的 LOGO



Sun 公司大事记

- 1982 Sun 公司成立（安迪·贝托谢姆和麦克尼利）。
- 1986 Sun 公司上市。
- 1985 Sun 公司推出著名的 Java 语言。
- 2001 9.11 事件前，Sun 市值超过 1000 亿美元；此后，由于互联网泡沫的破碎，其市值在一个月内跌幅超过 90%。
- 2004 Sun 公司和微软在旷日持久的 Java 官司中和解，后者支付前者高达 10 亿美元的补偿费。
- 2006 共同创始人麦克尼利辞去 CEO 一职，舒瓦茨担任 CEO 后尝试将 Sun 从设备公司向软件服务型公司转型，但不成功。
- 2010 Sun 公司被甲骨文公司收购。



Java 发展简史

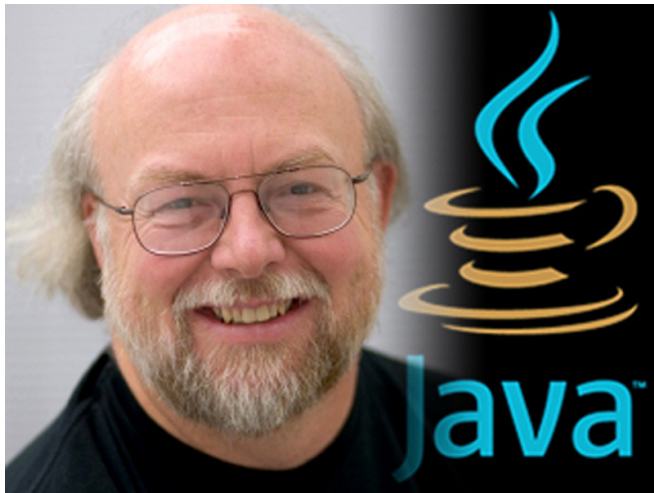
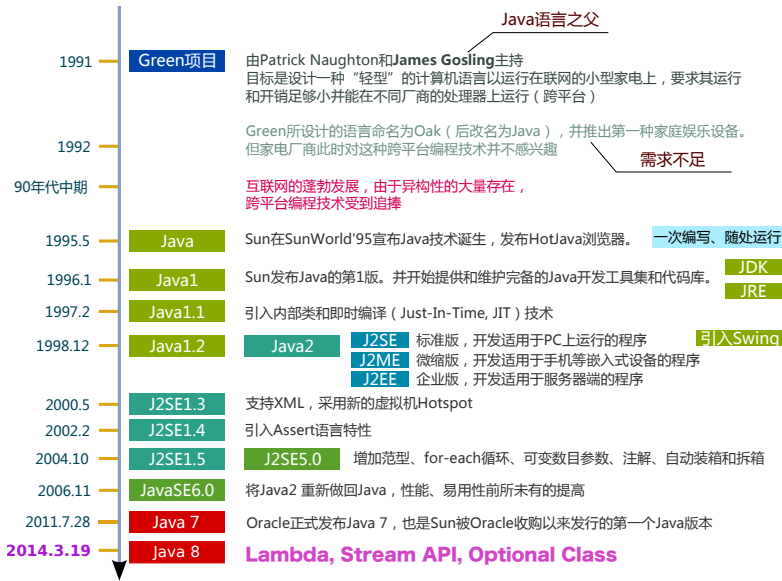


图: Java 之父詹姆斯·高斯林 (James Gosling)



Java 发展简史



Java 技术的特点

面向对象 Java 是一种以对象为中心，以消息为驱动的面向对象的编程语言。

平台无关性

分布式

可靠性

多线程

网络编程

编译和解释并存



Java 技术的特点

面向对象

平台无关性 分为源代码级（需重新编译源代码，如 C/C++）
和目标代码级 (Java) 平台无关。

分布式

可靠性

多线程

网络编程

编译和解释并存



Java 技术的特点

面向对象

平台无关性

分布式

可靠性

多线程

网络编程

编译和解释并存



Java 技术的特点

面向对象

平台无关性

分布式

可靠性 不支持直接操作指针，避免了对内存的非法访问；自动单元回收功能防止内存丢失等动态内存分配导致的问题；解释器运行时实施检查，可发现数组和字符串访问的越界；提供了异常处理机制。安全性。

多线程

网络编程

编译和解释并存



Java 技术的特点

面向对象

平台无关性

分布式

可靠性

多线程 C++ 没有内置的多线程机制，需调用操作系统的多线程功能来进行多线程程序设计；Java 提供了多线程支持。

网络编程

编译和解释并存



Java 技术的特点

面向对象

平台无关性

分布式

可靠性

多线程

网络编程

编译和解释并存



Java 技术的特点

面向对象

平台无关性

分布式

可靠性

多线程

网络编程

编译和解释并存 由编译器将 Java 源程序编译成字节码文件，再由运行系统解释执行字节码文件（解释器将字节码再翻译成二进制码运行）。



接下来...

Java 技术概述

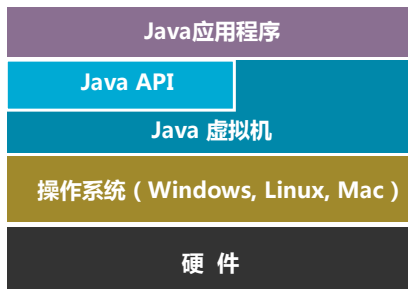
Java 平台核心机制

Java 开发环境

Java 基本开发流程



Java 平台

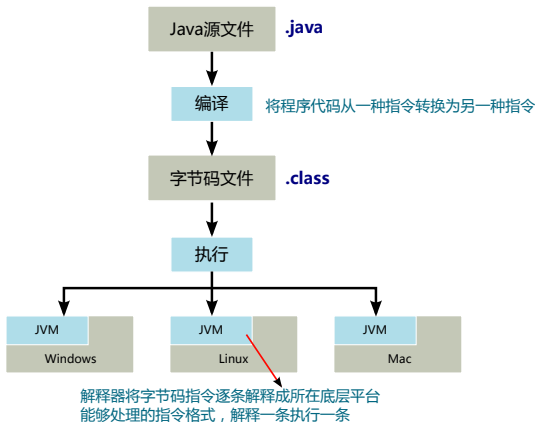


❖ 核心概念

- ▶ Java 虚拟机
- ▶ 垃圾回收机制
- ▶ Java 运行时环境 (Java Runtime Environment, JRE)



Java 程序的运行过程



JIT, Just-In-Time

传统解释器的解释执行是转换一条，运行完后就将其扔掉；JIT 会自动检测指令的运行情况，并将使用频率高（如循环运行）的指令解释后保存下来，下次调用时就无需再解释（相当于局部的编译执行），显著提高了 Java 的运行效率。



接下来...

Java 技术概述

Java 平台核心机制

Java 开发环境

Java 基本开发流程



获取和安装 Java 开发工具集

❖ Download

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

❖ Install

```
1 D:\Program Files\Java
```

```
1 /opt/jdk1.8.0_172
```

❖ Environment Variable

变量名 Path

变量值 D:\Program Files\Java\jdk1.8.0_172\bin



获取和安装 Java 开发工具集

❖ JDK Directories

```
1 bin COPYRIGHT db include javafx-src.zip
2 jre lib LICENSE man README.html release src.zip
3 THIRDPARTYLICENSEREADME-JAVAFX.txt THIRDPARTYLICENSEREADME.txt
```

bin Java 开发工具，包括编译器、虚拟机、调试器、反编译器等；

jre Java 运行时，包括 Java 虚拟机、类库和其他资源文件；

lib 类库和所需支持性文件；

include 用于调试本地方法（底层平台）的 C++ 头文件；

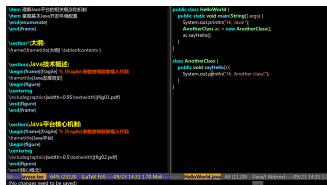
src.zip 类库的源代码；

db Java DB 数据库，JDK6.0 新增项目，一种纯 Java 的关系型数据库；



Java 开发工具

- ▶ Notepad
- ▶ Vim、Emacs

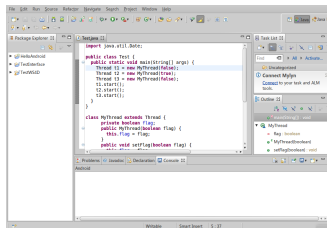


```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

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

class AnotherClass {
    public void sayHello() {
        System.out.println("Another class!");
    }
}
```

- ▶ Eclipse



接下来...

Java 技术概述

Java 平台核心机制

Java 开发环境

Java 基本开发流程



Java 基本开发流程

1. 创建源文件 HelloWorld.java，文件命名必须与类名相同。

```
1 public class HelloWorld {  
2     public static void main(String[] args) {  
3         System.out.println("Hi, Java!");  
4     }  
5 }
```

2. 将源文件编译为字节码文件

```
1 > javac HelloWorld.java && ls  
2 HelloWorld.class HelloWorld.java
```

3. 运行程序

```
1 > java HelloWorld  
2 Hi, Java!
```



Java 基本开发流程

1. 创建源文件 HelloWorld.java，文件命名必须与类名相同。

```
1 public class HelloWorld {  
2     public static void main(String[] args) {  
3         System.out.println("Hi, Java!");  
4     }  
5 }
```

2. 将源文件编译为字节码文件

```
1 > javac HelloWorld.java && ls  
2 HelloWorld.class HelloWorld.java
```

3. 运行程序

```
1 > java HelloWorld  
2 Hi, Java!
```



Java 基本开发流程

1. 创建源文件 HelloWorld.java，文件命名必须与类名相同。

```
1 public class HelloWorld {  
2     public static void main(String[] args) {  
3         System.out.println("Hi, Java!");  
4     }  
5 }
```

2. 将源文件编译为字节码文件

```
1 > javac HelloWorld.java && ls  
2 HelloWorld.class HelloWorld.java
```

3. 运行程序

```
1 > java HelloWorld  
2 Hi, Java!
```



Java 应用程序结构需掌握的几条规则

1. Java 语言拼写是大小写敏感的 (Case-Sensitive);
2. 一个源文件中可以定义多个 Java 类，但其中最多只能有一个类被定义为 Public 类;
3. 如果源文件中包含了 public 类，则源文件必须和该 public 类同名;
4. 一个源文件包含多个 Java 类时，编译后会生成多个字节码文件，即每个类都会生成一个单独的“.class”文件，且文件名与类名相同。



Java 应用程序结构需掌握的几条规则

1. Java 语言拼写是大小写敏感的 (Case-Sensitive);
2. 一个源文件中可以定义多个 Java 类，但其中最多只能有一个类被定义为 Public 类;
3. 如果源文件中包含了 public 类，则源文件必须和该 public 类同名;
4. 一个源文件包含多个 Java 类时，编译后会生成多个字节码文件，即每个类都会生成一个单独的“.class”文件，且文件名与类名相同。



Java 应用程序结构需掌握的几条规则

1. Java 语言拼写是大小写敏感的 (Case-Sensitive);
2. 一个源文件中可以定义多个 Java 类，但其中最多只能有一个类被定义为 Public 类;
3. 如果源文件中包含了 public 类，则源文件必须和该 public 类同名;
4. 一个源文件包含多个 Java 类时，编译后会生成多个字节码文件，即每个类都会生成一个单独的“.class”文件，且文件名与类名相同。



Java 应用程序结构需掌握的几条规则

1. Java 语言拼写是大小写敏感的 (Case-Sensitive);
2. 一个源文件中可以定义多个 Java 类，但其中最多只能有一个类被定义为 Public 类;
3. 如果源文件中包含了 public 类，则源文件必须和该 public 类同名;
4. 一个源文件包含多个 Java 类时，编译后会生成多个字节码文件，即每个类都会生成一个单独的“.class”文件，且文件名与类名相同。



本节习题

1. 安装配置 Eclipse Java 开发环境
2. 使用一个文本编辑器（记事本等）编写一个简单的 Java 程序，并从命令行编译执行该程序



THE END

wangxiaodong@ouc.edu.cn

