

1、纲要.....	1
2、内容.....	1
2.1、计算机语言发展简史.....	1
2.2、Java 发展简史.....	1
2.3、Java 语言特性.....	2
2.4、Java 的加载与执行.....	2
2.5、开始第一个 Java 程序.....	3
2.5.1、J2SE 的安装.....	3
2.5.2、搭建开发环境.....	3
2.6、public class 和 class 的区别.....	5

动力节点

# 1、纲要

计算机语言发展简史

Java 发展简史

Java 特性

Java 程序的加载与执行

J2SE 安装与配置

开始第一个 Java 程序

public class 和 class 的区别

## 2、内容

### 2.1、计算机语言发展简史

主要分为三个阶段：

- a) 第一代：机器语言
- b) 第二代：低级语言
- c) 第三代：高级语言

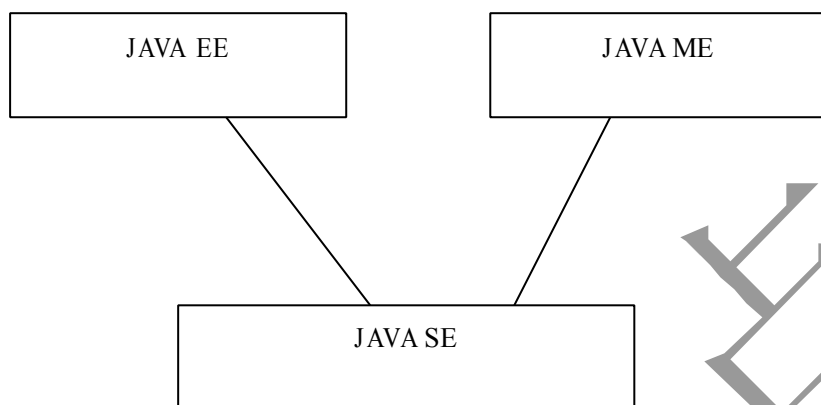
### 2.2、Java 发展简史

1. Java 是一种语言，具有自己的语法和数据结构
2. Java 来自于 sun, sun 指的是斯坦福大学网络
3. 具体发展历程

时间	描述
1991~1995	Sun 为了占领智能消费电子产品市场，由 james <b>gosling</b> 负责该项目，来开发 Oak 语言
1995	将 Oak 改名为 Java
1996	发布 JDK1.0
1997	发布 JDK1.1
1998	发布 JDK1.2, 将该版本命名为 <b>J2SDK</b> , 将 Java 更名为 <b>Java 2</b>
1999	将 java 分为三大块： <b>J2SE</b> (Java 标准版)、 <b>J2EE</b> (Java 企业版)、 <b>J2ME</b> (Java 微型版本)
2000	发布 J2SE1.3
2002	发布 J2SE1.4
2004	此时不再叫 J2SE1.5, 叫 5.0
2005	2005 Java 10 周年，将 J2SE 改为 <b>JAVA SE</b> 、将 J2EE 改为 <b>JAVA EE</b> 、将 J2ME 改为 <b>JAVA ME</b>

	ME
2006	发布 JAVA SE 6
2011 年 7 月	JAVASE 7

JAVA SE、JAVA EE、JAVA ME 之间的关系



## 2.3、Java 语言特性

简单性

面向对象

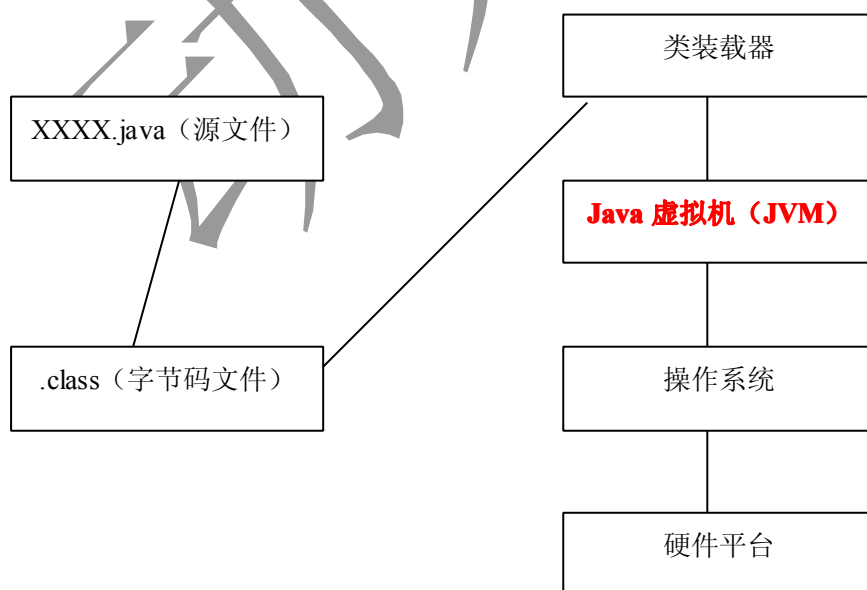
可移植性

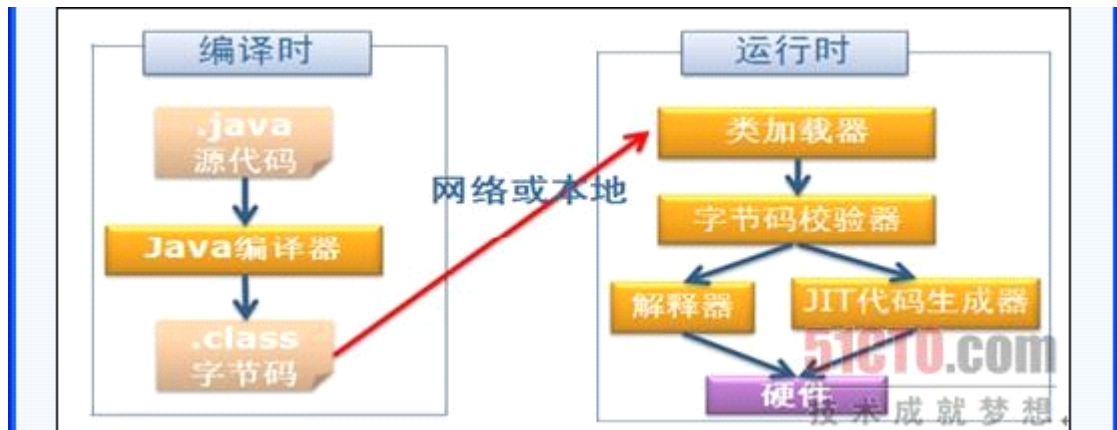
健壮性（自动垃圾回收机制，GC）

多线程

一次编译，到处运行

## 2.4、Java 的加载与执行





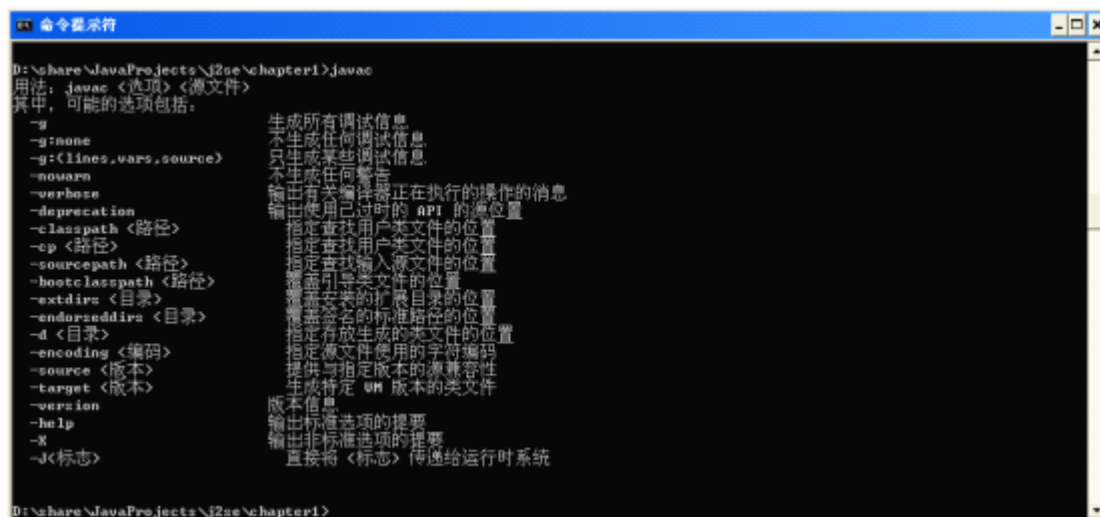
## 2.5、开始第一个 Java 程序

### 2.5.1、J2SE 的安装

1. 下载 J2SE 安装程序, [www.java.sun.com](http://www.java.sun.com)
2. 进行安装

### 2.5.2、搭建开发环境

1. 显示扩展名
2. 标题和地址栏最好显示完整路径
3. UltraEdit 最好去除备份
4. 常用命令
  - Cd 目录名称, 进入目录
  - Cd .., 退回上级目录
  - Dir, 显示文件或目录
  - Del 文件名, 删除文件
5. path, classpath (环境变量)
  - 主要是为了检索文件方便, 如果执行某个文件, 不用到文件所在目录执行, 那么操作系统会在 path 路径中检索
  - Classpath 为类路径, 是 java 运行时查找类文件的路径
6. 在命令行执行 javac, 如果出现如下显示, 说明正常

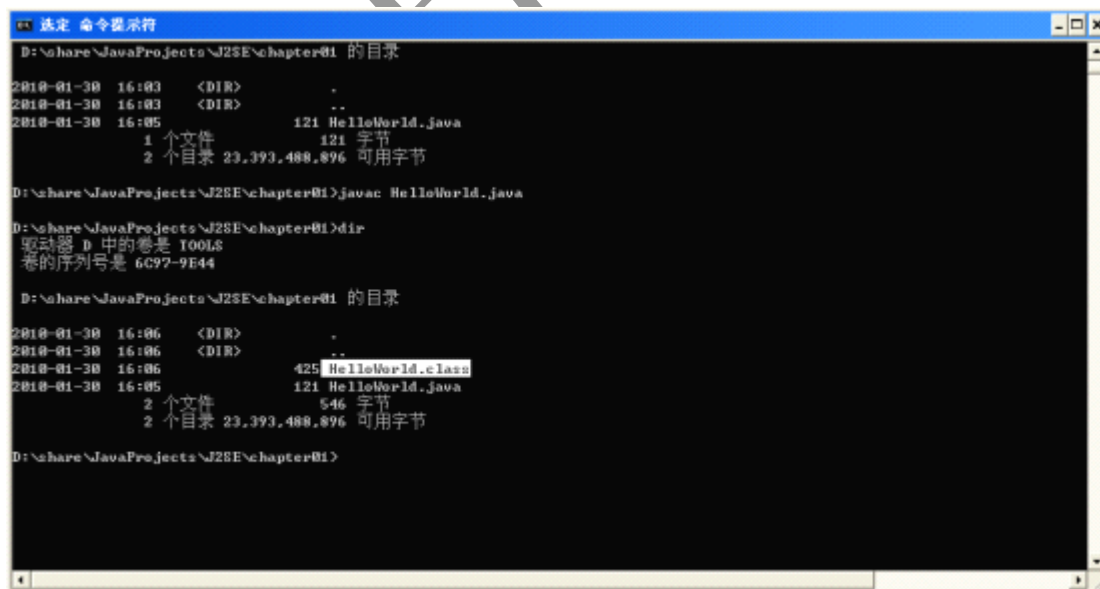


编译 HelloWorld 程序

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

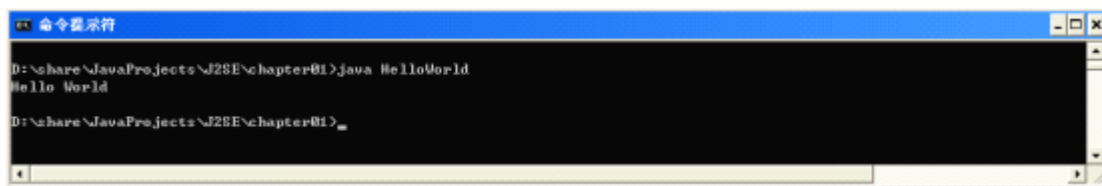
如何编译

**Javac** 文件名(完整的文件名称, 含扩展名), 如: javac HelloWorld.java



以上会形成一个 HelloWorld.class 文件, 说明编译成功了

开始运行 HelloWorld 程序,如何运行? 采用 java class 文件名 (不要扩展名), 如: java HelloWorld



第一个程序，成功运行

更换目录执行 HelloWorld 程序，会出现类不能找到，主要原因是没有设置 classpath, classpath 主要的作用是,方便 java 虚拟机加载 class 文件,配置方式有两种:

我的电脑—环境变量

命令 set classpath

解释 HelloWorld 程序

```
//public 代表，全局的含义
//采用 class 来定义类的名称
//HelloWorld 是类的名称，注意类的名称区分大小写
public class HelloWorld {

    //public 代表，全局的含义
    //static,表示是静态的
    //void,表示没有返回值
    //main,入口方法，方法的参数也不能改变
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

Java 注释在程序运行时是不执行的，注释主要有三种方式

- // 单行注释
- 多行注释  
/\*  
多行注释  
\*/
- 多行注释，可以生成 java doc 文档  
/\*\*  
\* 多行注释  
\* 多行注释  
\*/

## 2.6、public class 和 class 的区别

类的定义有两种方式:

- **public** class 类名
- class 类名

采用 **public class** 来声明 class，那么文件名必须和类名完全一致（包括大小写），如果文件名和类名不一致，将会出现如下错误:

```
命令提示符
D:\Share\JavaProjects\J2SE\chapter01>java HelloWorld1.java
HelloWorld1.java:1: 类 HelloWorld 是公共的, 应在名为 HelloWorld.java 的文件中声明
public class HelloWorld {
^
  错误
D:\Share\JavaProjects\J2SE\chapter01>
```

代码示例, 文件名为 HelloWorld1.java

```
public class HelloWorld {

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

采用 **class** 来声明类, 文件名可以任何合法的文件名称, 文件名不需要和 **Class** 类一致但生成的 **class** 文件名称和类名一致, 不和原文件名称一致

```
命令提示符
D:\Share\JavaProjects\J2SE\chapter01 的目录
2010-01-30 16:47 <DIR> .
2010-01-30 16:47 <DIR> ..
2010-01-30 16:06 425 HelloWorld.class
2010-01-30 16:05 121 HelloWorld.java
2010-01-30 16:05 121 HelloWorld1.java
2010-01-30 16:47 429 HelloWorld123.class
2010-01-30 16:47 117 HelloWorld2.java
5 个文件 1,213 字节
2 个目录 23,393,329,452 可用字节
D:\Share\JavaProjects\J2SE\chapter01>
```

如何执行?

```
命令提示符
D:\Share\JavaProjects\J2SE\chapter01 的目录
2010-01-30 16:47 <DIR> .
2010-01-30 16:47 <DIR> ..
2010-01-30 16:06 425 HelloWorld.class
2010-01-30 16:05 121 HelloWorld.java
2010-01-30 16:05 121 HelloWorld1.java
2010-01-30 16:47 429 HelloWorld123.class
2010-01-30 16:47 117 HelloWorld2.java
5 个文件 1,213 字节
2 个目录 23,393,329,452 可用字节
D:\Share\JavaProjects\J2SE\chapter01>java HelloWorld123
Hello World
D:\Share\JavaProjects\J2SE\chapter01>
```

在一个 Java 文件中, 可以声明多个类, 如:

```
public class HelloWorld3 {

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

//在一个 java 源文件中只能有一个 class 被 public 修饰

```
//public class A{
```

```
//}
```

```
class A {
```

```
}
```

需要注意：只能有一个类采用 `public class` 进行声明

动力节点