

面向对象程序设计 第2章

部分内容摘自

《Java面向对象编程》，孙卫琴

《JAVA 程序设计》，唐大仕

2. 1 Java编程语言的主要特性

- 面向对象
- 可移植性，跨平台
- 支持分布式的网络应用
- 安全性和健壮性



Java虚拟机

■ 概念

- Java虚拟机 (Java Virtual Machine) 在实际的计算机上通过软件模拟来实现。
- Java虚拟机有自己想象中的硬件。

■ 功能

- 提供垃圾回收功能
- 提供运行时环境

关于跨平台的概念

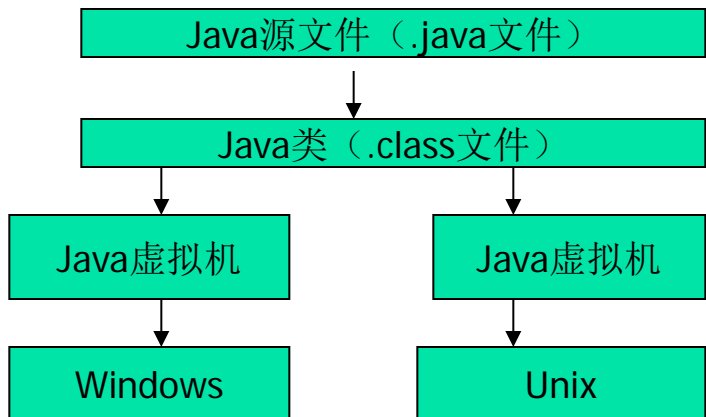
- Java**是一次编译**到处运行

Java编译为.class文件后，到其他操作系统无需再次编译

- Java**不是一次编写**到处运行

c++文件在windows 上编写、编译成exe文件运行，同样的代码在Linux要再次编译为Linux上可执行文件

Java虚拟机



- .class可以跨平台（操作系统）
- Java虚拟机每个平台（操作系统）不一样

Java程序运行模式

- 1 网页中的Applet，依托浏览器运行，目前已经很少使用了
- 2 Web方式（需要依托webserver如tomcat/websphere...）。
- 3 Application，本课程重点讨论。

2.2 配置开发环境 见《00如何配置开发环境.docx》

- 安装JDK
- 配置环境变量
- 安装Eclipse
- 建立空项目
- 修改默认字符集
- 导入课程源代码

2.3Java程序最基础知识

一个Java应用程序 例子: ch02.HelloWorldApp

这个文件名字必须是src/ch02/HelloWordApp. java

```
package ch02; //包的名字, 注意一定要和目录严格匹配
```

```
// 注意如何定义一个类, 每个.java文件只能定义一个public 类, 类名字必须和文件名完全符合
```

```
public class HelloWorldApp {
```

```
    // 任何一个Application程序必须要有一个 public static void main(String args[])方法
```

```
    //但是不是所有的类都需要 main方法
```

```
    public static void main(String args[]) {
```

```
        System.out.println("Hello World!");
```

```
        // System.out.println("")等价于C++中的printf()
```

```
    }
```

```
}
```

关于包

- 每个包 (package) 对应了源程序的一个目录，每个包可以有子包（子目录）
- JDK 的 Java 类库中的几个重要包： `java.lang`, `java.io`, `java.awt` , `java.net` , `java.util`
- 包不同层级之间用 `.` 分割，正如目录用 `/` 分割。
- 一个包下包含了若干类文件或者子包
- 包 `ch02` 对应了 `ch02/`
- 包 `ch02.sub01` 对应了 `ch02/sub02/`

关于Java的源文件

- 初学者可以认为一个源文件(. java)对应一个类的定义（实际上这种说法并不准确，初学者可以这样认为）
- . java文件经过编译后生成一个.class文件
- Java虚拟机解释.class文件执行程序(java不能直接生成.exe文件)

关于main方法

- 每个能够以Java Application模式运行的类中必须有一个main方法
- `public static void main(String[] args)`
- 或: `static public void main(String[] args)`

JAVA最基本语法

- { } 程序段

- 每行程序以分号 ; 结束

// 单行注释

/*

*/ 多行注释

- System.out.println(); 输出并换行

- System.out.print(); 输出不换行

Java对大小写**非常敏感**！

- 包名、类名、方法名、变量名均大小写敏感！

字符界面的输入与输出

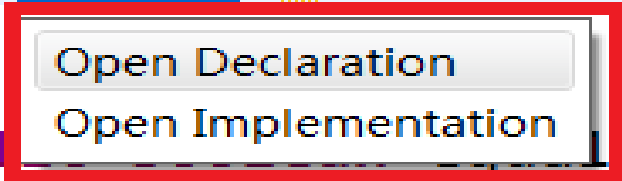
- java.io包
- System.in.read()
- System.out.print()/println()
- 程序：ch02.AppCharInOut
- 程序：ch02.AppLineInOut
- 程序：ch02.AppNumInOut

2.4 如何查看到JDK源代码

- 以Eclipse oxygen 为例
- 操作系统 Windows7 64位
- 其他操作系统及软件版本操作类似

1 按住ctrl 鼠标点击对应类名

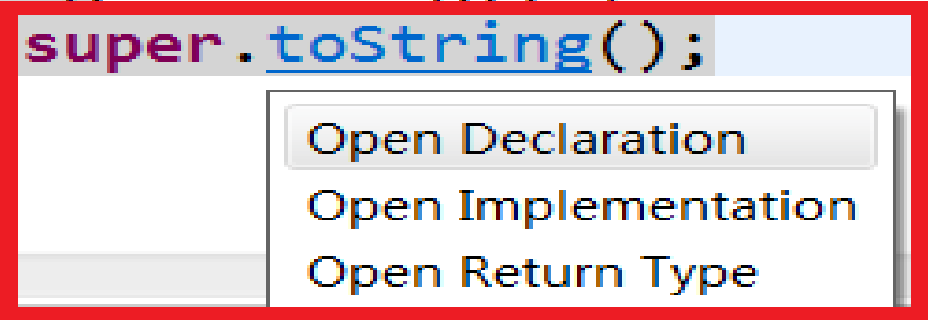
```
public class EqDemo {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        Object o;  
    }  
    public void test(EqDemo demo) {  
        return true;  
    }  
}
```



或者ctrl 鼠标点击方法名

```
public boolean equals(EqDemo demo) {  
    return true;  
}
```

```
public String toString() {  
    return super.toString();  
}
```



- Open Declaration
- Open Implementation
- Open Return Type

-
- 如果能看到JDK源代码
则不用看本PPT以后的内容, 否则继续

如果你看到这个页面 请继续

Class File Editor

Source not found

The JAR file C:\Program Files\Java\jre1.8.0_102\lib\rt.jar has no source attachment.
You can attach the source by clicking Attach Source below:

[Attach Source...](#)

```
// (version 1.8 : 52.0, super bit)
public class java.lang.Object {

    // Method descriptor #8 ()V
    // Stack: 0, Locals: 1
    public Object();
        0 return
```

回到方才失败的页面

Class File Editor

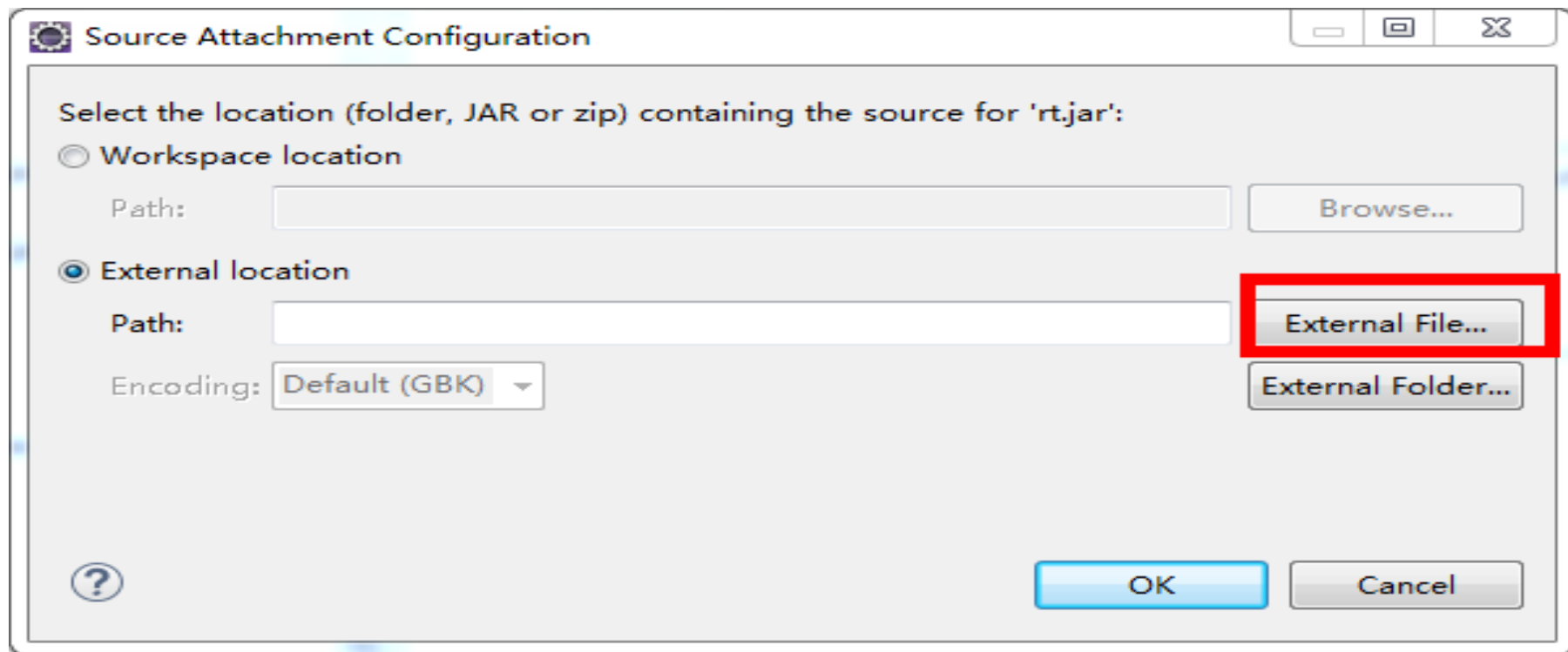
Source not found

The JAR file C:\Program Files\Java\jre1.8.0_102\lib\rt.jar has no source code.
You can attach the source by clicking Attach Source below:

[Attach Source...](#)








```
// (version 1.8 : 52.0, super bit)
public class java.lang.Object {
```

点击 External Location >>External File



选择JDK安装路径下的源代码文件压缩包

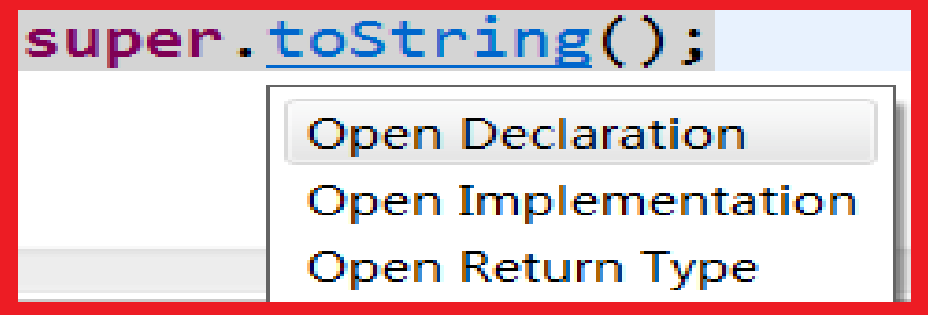
▶ Program Files ▶ Java ▶ jdk1.8.0_102 ▶

名称	修改日期
 bin	2017/9/18 15:34
 db	2017/9/18 15:34
 include	2017/9/18 15:35
 jre	2017/9/18 15:35
 lib	2017/9/18 15:36
 javafx-src.zip	2017/9/18 15:35
 src.zip	2016/6/22 13:37

代码绑定完毕！ 重新方法》 右键

```
public boolean equals(EqDemo demo) {  
    return true;  
}
```

```
public String toString() {  
    return super.toString();  
}
```



- Open Declaration
- Open Implementation
- Open Return Type