

Java概述

发展历史

1990年Sun成立智能电器项目，项目失败，改为支持浏览器小程序（Applet）

1995年发行Java语言

1996年JDK 1.0

1998年JDK1.2, 包含Servlet/JSP

2002年JDK1.4, 最为稳定成熟的版本

2004年JDK1.5, 改名Java SE 5.0

2009年，Oracle收购Sun

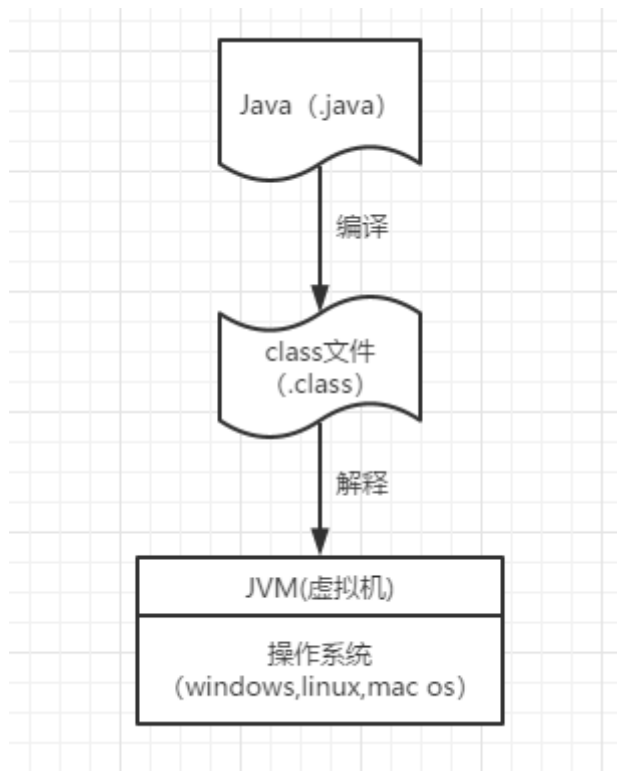
目前最新版本Java SE 15，开发用的主流版本1.8

Java之父：James Gosling



Java的运行机制

Java既有编译，也有解释过程。



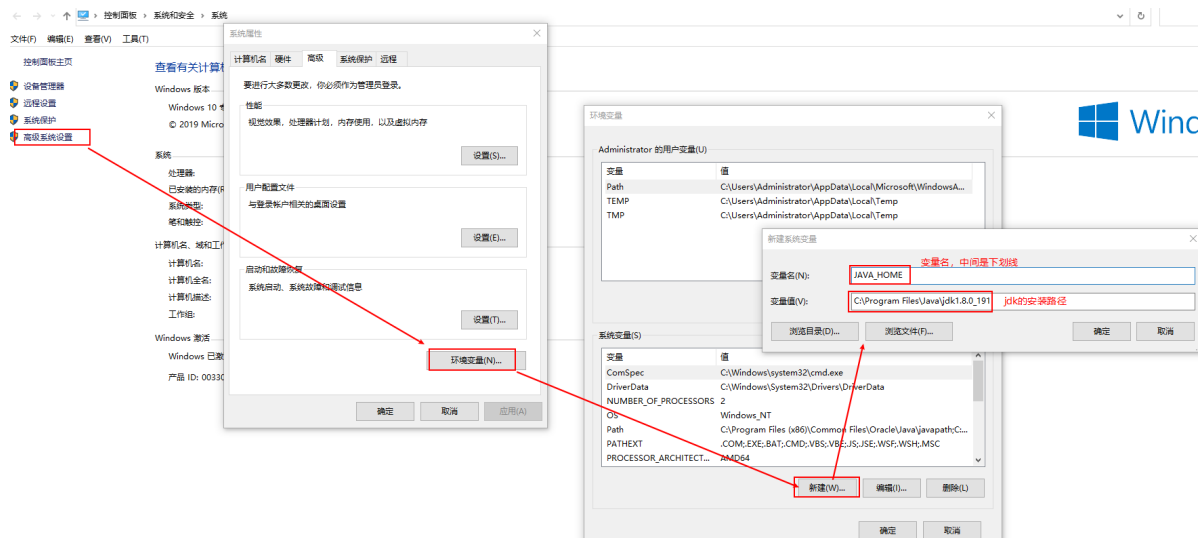
jdk的安装配置

安装JDK

按提示一步一步安装完毕

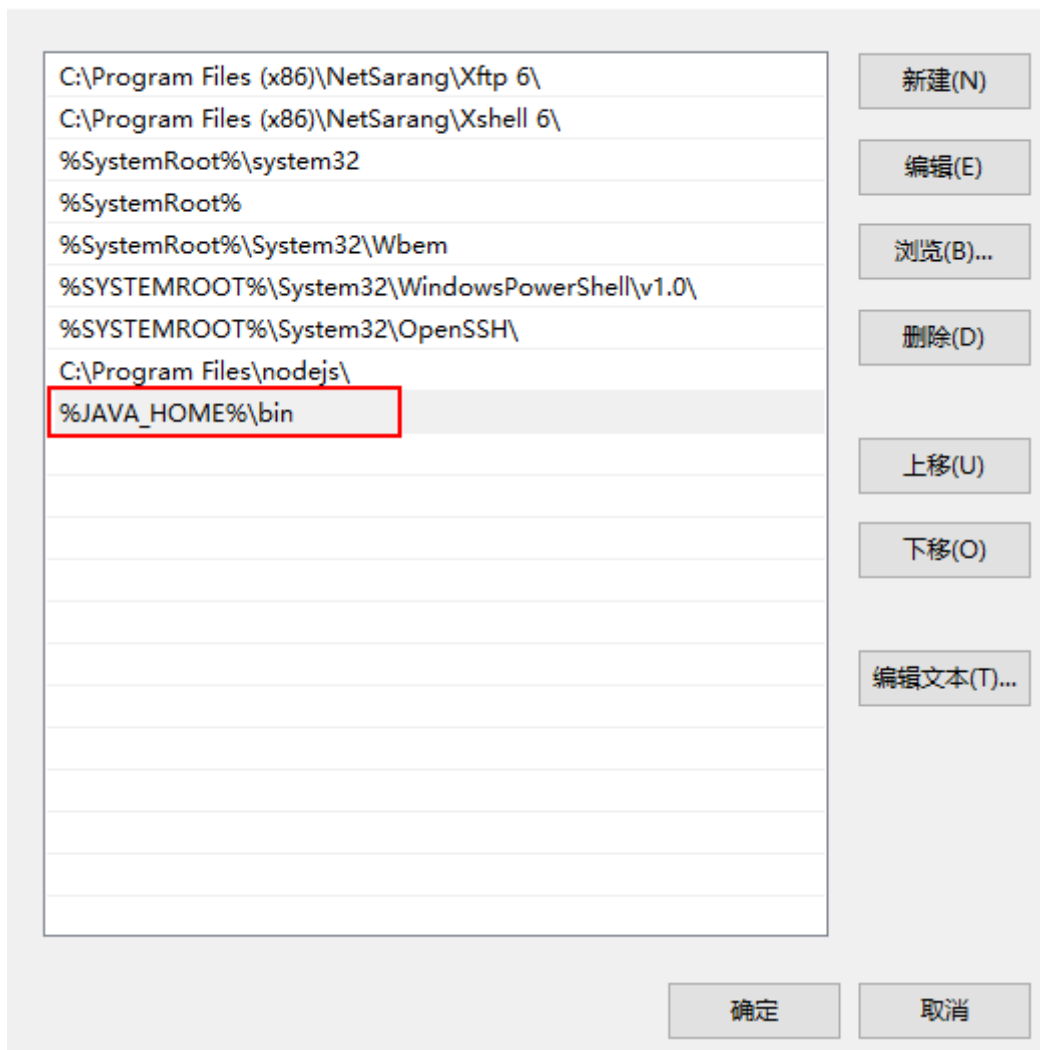
配置环境变量

- 配置JAVA_HOME



- 配置path

在path变量中增加一项，如下图：



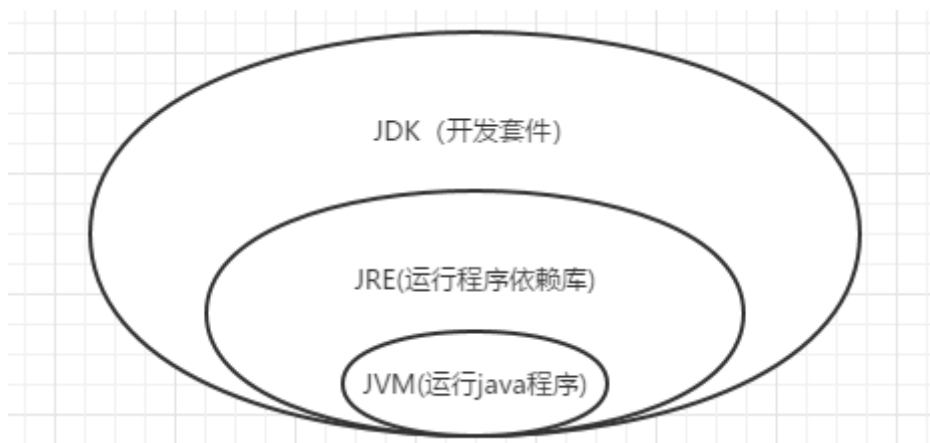
验证jdk是否正确安装

重新打开命令行窗口（win+r, 输入cmd并回车），因为命令行窗口不会自动刷新环境变量配置，所以修改环境变量配置后一定要重新打开命令行窗口

输入名：java -version

如果正确配置，会显示jdk的版本号

JVM, JRE, JDK三者之间的关系？



Java SE, Java EE , Java ME?

Java SE : Java Standard Edition, Java标准开发版本

Java EE: Java Enterprise Edition, Java企业开发版本

Java ME: Java Micro Edition, Java嵌入式开发版本

手动编写第一个Java程序并编译执行

创建java源文件(Hello.java)

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

编译源文件为class文件

执行命令: javac Hello.java

执行class文件

执行命令: Java Hello

注意: 执行时不需要写.class扩展名

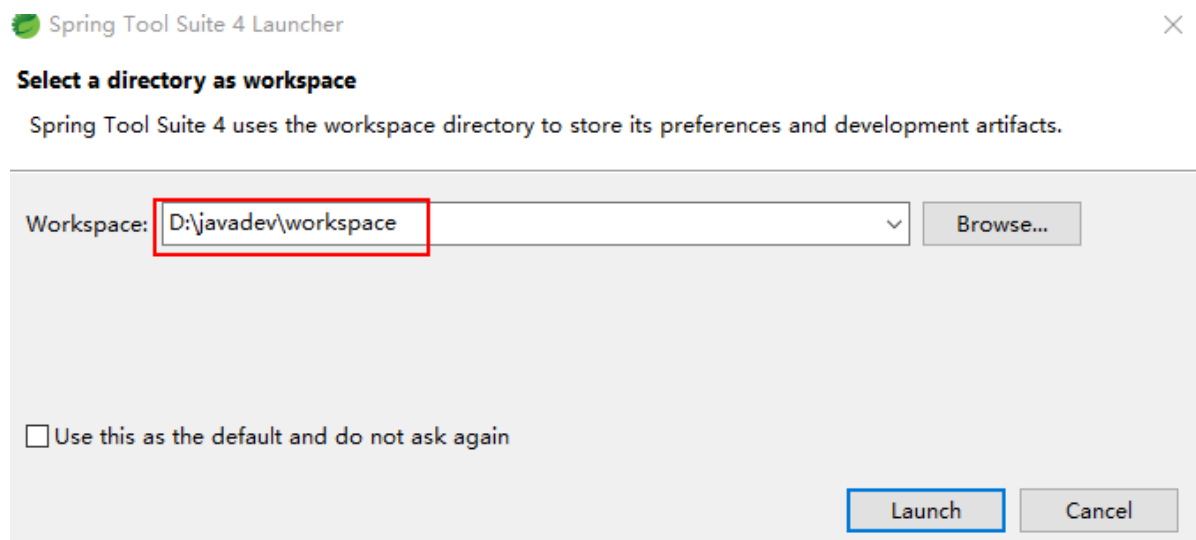
Java集成开发工具 (IDE)

Java开发一般用Eclipse或IDEA这两种工具。

STS: 封装Eclipse, 集成Spring开发插件的一个IDE

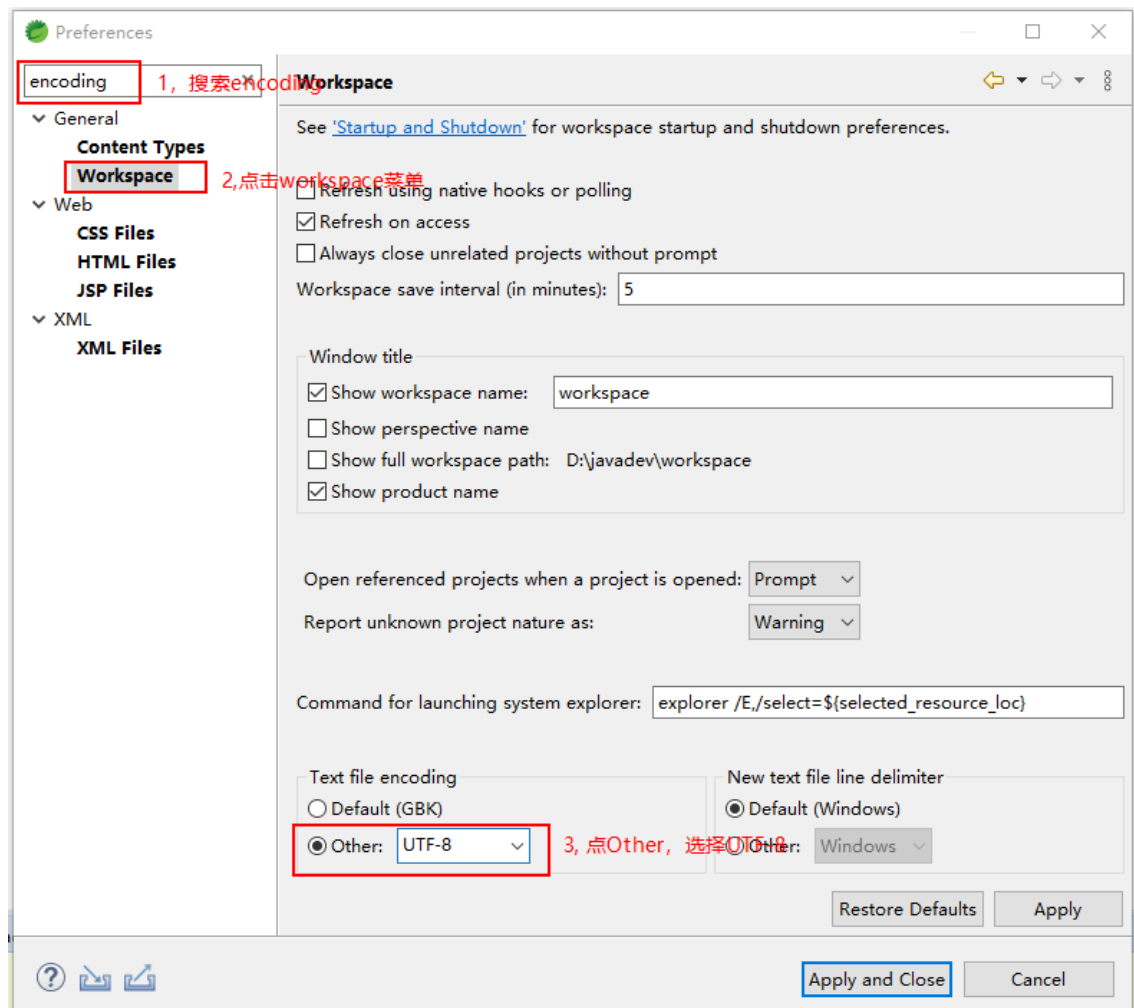
STS的安装配置

- 解压STS包到放软件的目录, 把可执行文件 (SpringToolSuite4.exe) 发送快捷方式到桌面
- 在开发目录下创建一个文件夹叫workspace
- 启动STS, 在工作空间选择界面选择我们刚才创建的workspace目录



- 修改工作空间编码

点击window菜单 --> Preferences, 打开下面创建, 按步骤修改:



STS中创建项目 (Project) 并运行

- file --> new --> Java Project

在创建窗口输入项目名称, 点击完成

- 展开项目目录, 在src目录上点击鼠标右键, 点击菜单: new --> Class

在弹出窗口里输入文件名 (不需要输入扩展名), 勾选生成main方法

New Java Class

Java Class

Create a new Java class.

Source folder: Browse...

Package: Browse...

☐ Enclosing type: Browse...

Name: 1, 输入文件名

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☒ public static void main(String[] args) 2, 勾选创建main方法
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

Finish Cancel

- 在main方法中输入一条打印语句
快捷操作：输入syso，然后按组合键Alt+/, 会自动补全语句。在方法参数里面输入要打印的内容。
完整的文件如下：

```
package javabaseday01_t;  
  
public class Hello {  
  
    public static void main(String[] args) {  
        System.out.println("Hello HQYJ");  
    }  
  
}
```

- 运行程序
在代码窗口点击鼠标右键：Run as --> Java Application
运行后在Console控制台看到输出结果

Java程序的规则

- Java类（源文件）是Java代码的最小单位，一个工程可以包含若干个Java类
- Java类的名字与文件名相同，并且首字母大写，使用驼峰命名法，文件名的后缀名是.java
- Java类的入口方法叫main，一个类只有一个main方法，main方法的格式是固定写法
- 每个语句用分号结束。一个语句可以跨多行；一行可以写多个语句
- 方法名以小写字母开头，用驼峰命名方法

Java注释

单行注释（Ctrl+/）

- 用于添加代码的说明，可以在代码行上面添加注释；或者在代码末尾添加注释，注释内容以//开头，

```
//      打印欢迎信息
System.out.println("Hello HQYJ");//打印欢迎信息
```

- 用于屏蔽暂时不想让他生效的代码

```
//      System.out.println("Hello HQYJ");
```

多行注释(Ctrl+Shift+/; Ctrl+Shift+\)

注释说明文字有多行，或者多行代码暂时屏蔽掉

```
/*
    * 说明文字
    * 占用多行
    */
/*
    * System.out.println("Hello HQYJ"); System.out.println("Hello HQYJ");
    * System.out.println("Hello HQYJ"); System.out.println("Hello HQYJ");
    * System.out.println("Hello HQYJ"); System.out.println("Hello HQYJ");
    * System.out.println("Hello HQYJ"); System.out.println("Hello HQYJ");
    * System.out.println("Hello HQYJ");
    */
```

文档注释

用于类或方法上，描述这个类或方法的使用方式。JavaDoc可以导出成API接口供其它人参考。（附录：JDK8的API路径：<https://docs.oracle.com/javase/8/docs/api/>）

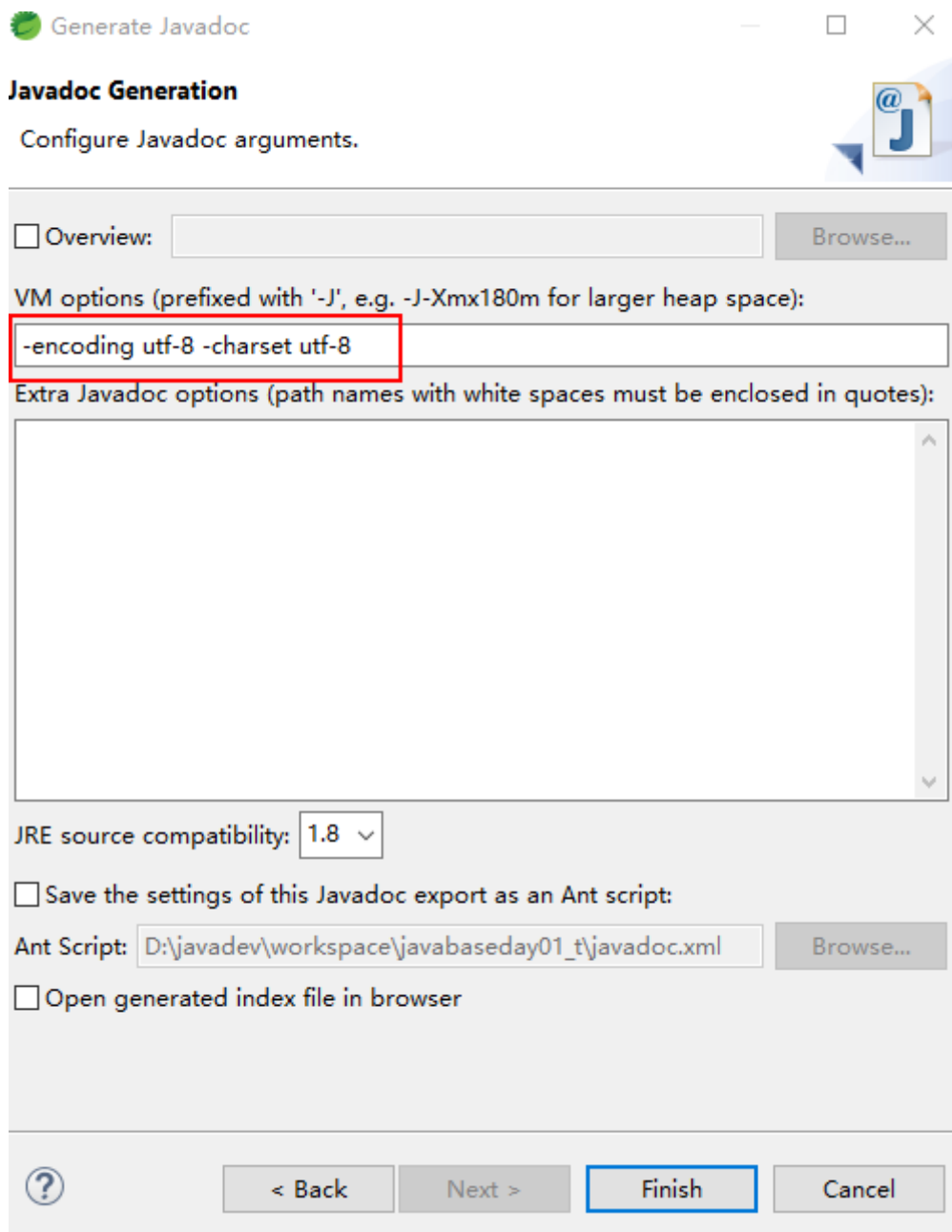
文档注释以/**开头，以*/结束

```
/**
 * 这是第一个Java程序
 * @author Administrator
 *
 */
public class Hello {

    /**
     * 入口方法
     * @param args 参数
     */
    public static void main(String[] args) {
```

文档注释导出为API的步骤：

- 在项目上面点击右键： export --> java --> Javadoc, 在最后一步写上VM编码设置（否则会中文乱码）



- 执行完成后在项目目录下面会生成doc目录，所有API文档就存放在这个目录

今日作业

1. JDK的环境变量配置？（假设jdk安装目录是 d:\java\jdk15_121）
2. Java程序是编译执行？ 还是解释执行？
3. Java程序的原件和可执行文件的扩展名分别是什么？
4. 程序题。

创建Teacher, Student两个类

分别为这连个类写文档注释，并导出API文档

5. 程序题。创建一个类，添加main方法，打印如下图形：

