

第一讲 Java语言基础

初版时间: 2017年8月1日

初版制作者: 林焕彬

教材版本号: szlanyou-V1.1

修订时间:

修订者:

教材版本号: szlanyou-V1.1

讲师介绍

姓 名：林焕彬
部 门：研发一部
岗 位：Java架构师

简要介绍：十多年的Java项目开发
架构管理经验，曾在学校、机构
和企业培训的经历，曾在北大青
鸟做高级培训讲师。

联系电话：
Email:linhuanbin@szlanyou.com

照片太帅无法显示

目录



Java基础设置

Java代码规范

Java类型转换

Java选择结构

Java循环结构

一、Java的发展历史 1.1

1991年，Sun 公司Green 项目的Oak 语言（后改名Java）

1995年，Sun 命名并发布了 Java 1.0, JDK 1.0

1998年推出Java 1.2, JDK 1.2, Java 2

2005年，Java10周年，重新命名：

J2SE → Java SE → Standard Edition

J2ME → Java ME → Micro Edition

J2EE → Java EE → Enterprise Edition

2010年，Sun被Oracle并购

一、Java虚拟机与跨平台原理 1.2

编辑 .java文件
编译 .class文件
运行



一、Java环境设置 1.3

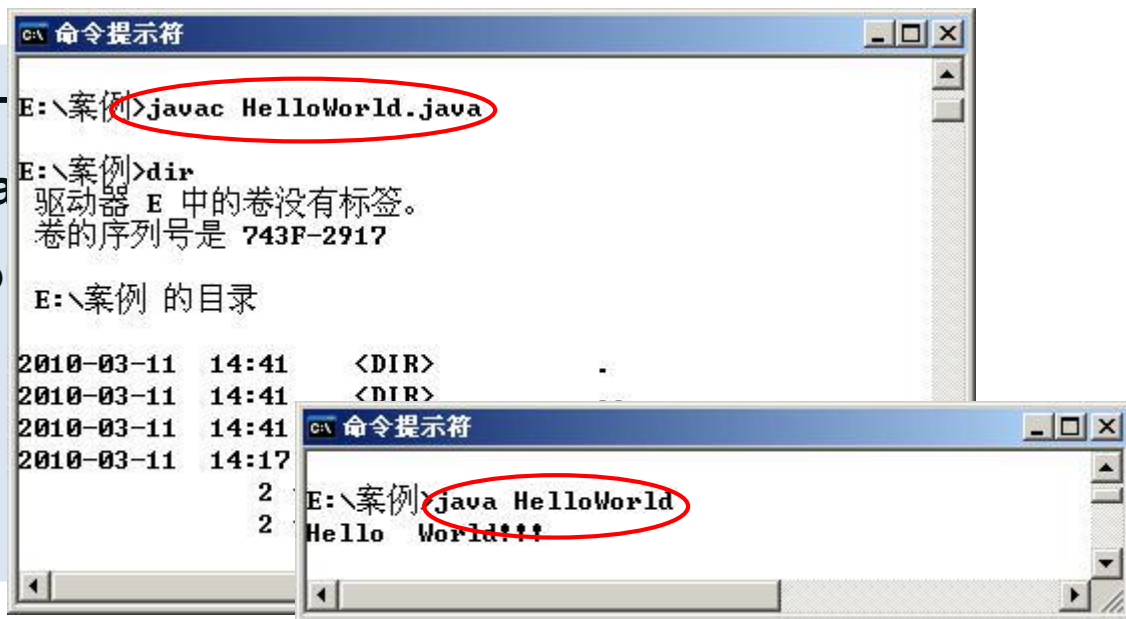
- ◆ 首先下载安装JDK安装包
- ◆ 打开我的电脑--属性--高级--环境变量
 - 设置环境变量JAVA_HOME指向JDK安装目录
 - 设置环境变量Path=%JAVA_HOME%/bin
 - 设置环境变量
ClassPath=%JAVA_HOME%/lib/dt.jar;%JAVA_HOME%/lib/tools.jar
- ◆ 使用java和javac命令检查java环境是否设置成功

一、使用记事本开发Java程序 1.4

练习

使用记事本编辑源程序，以.java为后缀名保存
使用javac命令编译.java文件，生成.class文件
使用java命令运行.class文件，输出程序结果

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



演示示例：使用记事本开发Java程序

目录



Java基础设置

Java代码规范

Java类型转换

Java选择结构

Java循环结构

二、Java代码规范 2.1

- 编码规范是作为一个程序员应该遵守的基本规则，是行业内大家都默守的做法
- Java编码基础规范
 - 类名使用public修饰
 - 一行只写一条语句
 - { }的使用及位置
 - 代码缩进
- 尽量使用有意义的英文命名，少用拼音，不用中文

二、Java代码规范 2.2



java代码规范.doc

二、Java程序的注释 2.3

为了美观，多行注释的每一行
开头可写一个或多个 *

多行注释以 “/*” 开头
以 “*/” 结尾

```
/*  
 * HelloWorld.java           ← 文件的名称  
 * 2013-03-21               ← 日期  
 * 我的第一个Java程序      ← 功能说明  
*/  
  
public class HelloWorld{  
    public static void main(String[ ] args){  
        //输出消息到控制台  
        System.out.println("Hello World!!!");  
    }  
}
```

注释以 // 开始
以行末结束

目录



Java基础设置

Java代码规范

Java类型转换

Java选择结构

Java循环结构

三、自动类型转换 3.1



示例

某班第一次学士后Java考试平均分81.29，第二次比第一次多2分，计算第二次考试平均分。

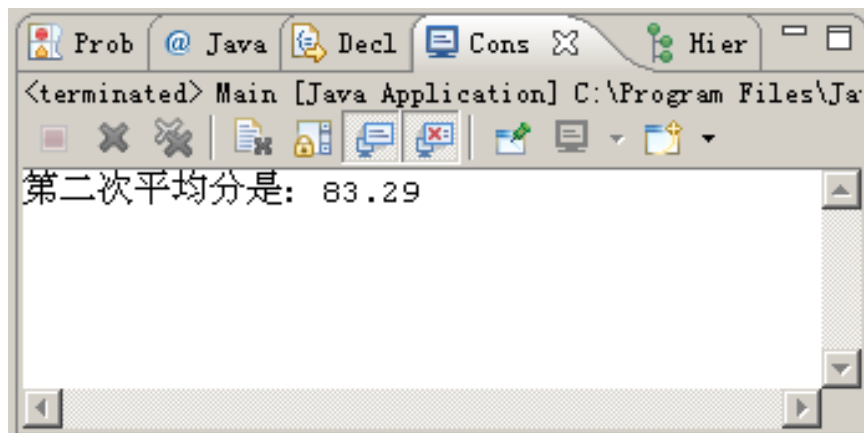
```
double firstAvg = 81.29; //第一次平均分
```

```
double secondAvg;      //第二次平均分
```

```
int rise = 2;
```

```
secondAvg = firstAvg + rise;
```

```
System.out.println("第二次平均分是： " + secondAvg);
```



三、自动类型转换规则 3.2

如果一个操作数为double型，则整个表达式可提升为double型

满足自动类型转换的条件

两种类型要兼容

数值类型（整型和浮点型）互相兼容

目标类型大于源类型

例如：double 型大于 int 型

三、强制类型转换 3.3



语法

(数据类型) 表达式



问题

```
int b = (int)10.2;  
double a = 10;  
int c = (int)a;
```

去年Java所占市场份额是20，今年增长的市场份额是9.8，求今年所占份额？

```
int before = 20;    //去年市场份额  
double rise = 9.8; //增长的份额
```

编译出错：不能完成自动类型转换

```
int now = before + rise; //现在的份额
```

强制类型转换

更改为

```
int now = before + (int)rise;
```

 演示示例：强制类型转换

目录



Java基础设置

Java代码规范

Java类型转换

Java选择结构

Java循环结构

四、基本的if选择结构 4.1

if选择结构是根据条件判断结果选择不同的处理

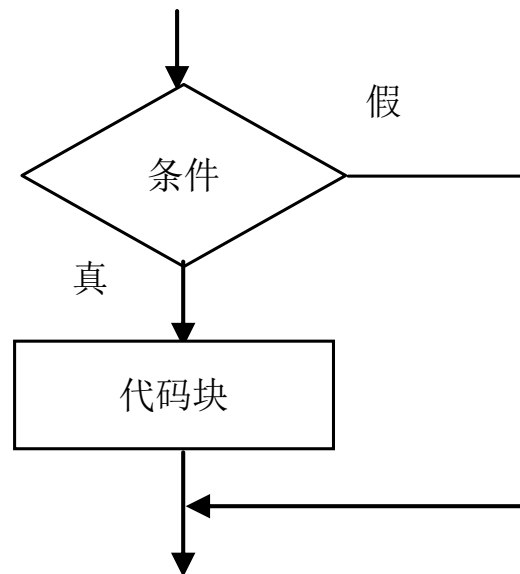
语法 基本的if选择结构

```
if ( 条件 ) {  
    //代码块  
}
```

只有一条语句时，
建议不省略{ }

示例

```
if (王梁的Java考试成绩 > 90 ) {  
    //获得一个iPhone5作为奖励;  
}
```



四、使用复杂条件的if选择结构 4.2



王梁Java成绩大于90分，并且音乐成绩大于80分时，或者Java成绩等于100分，音乐成绩大于70分时，老师奖励他，该怎么做呢？

运算符	表达式	说 明
&&	条件1 && 条件2	仅仅两个条件同时为真，结果为真
	条件1 条件2	只要两个条件有一个为真，结果为真
!	! 条件	条件为真时，结果为假 条件为假时，结果为真

四、用复杂条件的if选择结构 4.3

提示 运算符的优先级

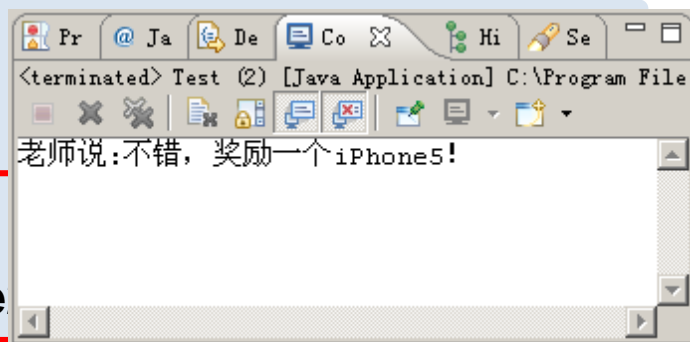
刚提出的问题条件应该怎么写呢？
最高的优先级

- 最低的优先级：=
- 优先级顺序：！ > 算术运算符 > 比较运算符 > && > ||



示例

```
int score1 = 100; //王梁的Java成绩
int score2 = 72; //王梁的音乐成绩
if ( ( score1 > 90 && score2 > 80 )
    || ( score1 == 100 && score
System.out.println("老师说:不错，奖励一个iPhone5!
");
}
```



演示示例：复杂条件下的if选择结构

目录



Java基础设置

Java代码规范

Java类型转换

Java选择结构

Java循环结构

五、for循环 5.1



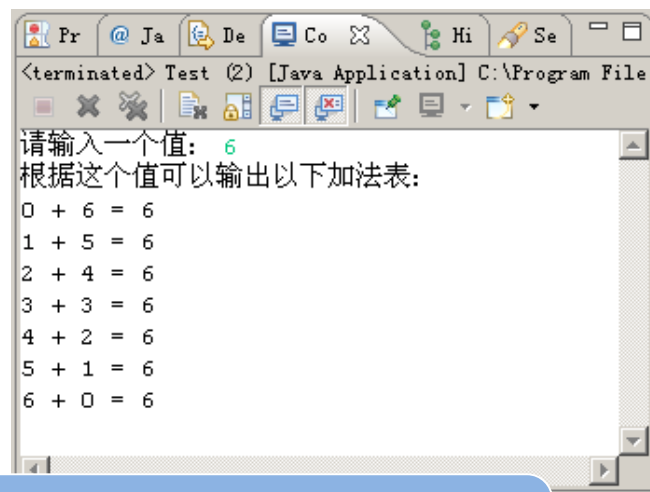
问题

输出加法表



分析

1. 循环初始化: $i = 0$; $j = \text{输入值}$
2. 循环条件: $i \leq \text{输入值}$
3. 循环操作: 计算 $i+j$
4. 循环变量的改变: $i++$, $j--$



表达式3可以用“,”隔开的多个表达式, 运算顺序从左到右

```
for( int i = 0, j = val; i <= val; i++, j-- ){  
    System.out.println(i + " + " + j + " = " + (i+j));  
}
```

表达式1中可以声明多个同一类型的值并赋值, 用“,”隔开



演示示例: 使用for循环结构输出加法表

五、上机练习 5.2

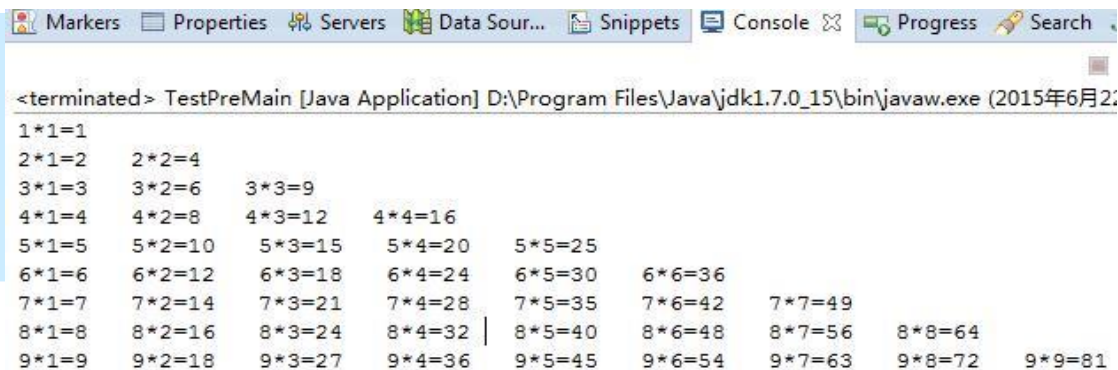


需求说明

- 打印出九九乘法表

提示

- 使用嵌套for循环



```
<terminated> TestPreMain [Java Application] D:\Program Files\Java\jdk1.7.0_15\bin\javaw.exe (2015年6月21日 14:14:14)
1*1=1
2*1=2  2*2=4
3*1=3  3*2=6  3*3=9
4*1=4  4*2=8  4*3=12  4*4=16
5*1=5  5*2=10  5*3=15  5*4=20  5*5=25
6*1=6  6*2=12  6*3=18  6*4=24  6*5=30  6*6=36
7*1=7  7*2=14  7*3=21  7*4=28  7*5=35  7*6=42  7*7=49
8*1=8  8*2=16  8*3=24  8*4=32  8*5=40  8*6=48  8*7=56  8*8=64
9*1=9  9*2=18  9*3=27  9*4=36  9*5=45  9*6=54  9*7=63  9*8=72  9*9=81
```

完成时间：10分钟

共性问题集中讲解

总结

提问

Java基础环境设置，了解Java虚拟机与跨平台原理

Java代码规范和程序的注释

Java自动类型转换和强制类型转换

Java基本和复杂条件的if选择结构

Java for 循环结构

Thank you

Q&A

