第01天 java基础知识

今日内容介绍

* Java概述、开发环境搭建
* Eclipse的概述、配置及使用
* **注释、关键字、常量、变量、数据类型、标识符**

第1章 Java概述

## 1.1 Java语言发展史

A:Java语言发展史

詹姆斯·高斯林（James Gosling）1977年获得了加拿大卡尔加里大学计算机科学学士学位，1983年获得了美国卡内基梅隆大学计算机科学博士学位，毕业后到IBM工作，设计IBM第一代工作站NeWS系统，但不受重视。后来转至Sun公司，1990年，与Patrick，Naughton和Mike Sheridan等人合作“绿色计划”，后来发展一套语言叫做“Oak”，后改名为Java。

SUN(Stanford University Network，斯坦福大学网络公司)

B:Java语言版本

\* JDK 1.1.4 Sparkler 宝石 1997-09-12

\* JDK 1.1.5 Pumpkin 南瓜 1997-12-13

\* JDK 1.1.6 Abigail 阿比盖尔--女子名 1998-04-24

\* JDK 1.1.7 Brutus 布鲁图--古罗马政治家和将军 1998-09-28

\* JDK 1.1.8 Chelsea 切尔西--城市名 1999-04-08

\* J2SE 1.2 Playground 运动场 1998-12-04

\* J2SE 1.2.1 none 无 1999-03-30

\* J2SE 1.2.2 Cricket 蟋蟀 1999-07-08

\* J2SE 1.3 Kestrel 美洲红隼(sǔn) 2000-05-08

\* J2SE 1.3.1 Ladybird 瓢虫 2001-05-17

\* J2SE 1.4.0 Merlin 灰背隼 2002-02-13

\* J2SE 1.4.1 grasshopper 蚱蜢 2002-09-16

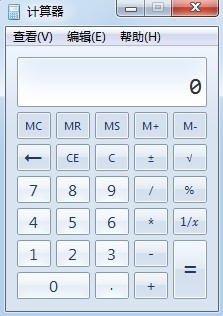
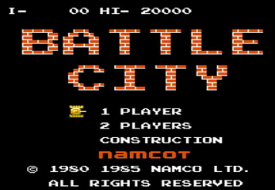
\* J2SE 1.4.2 Mantis 螳螂 2003-06-26

\* JAVASE 5.0 (1.5.0) Tiger 老虎

\* JAVASE 5.1 (1.5.1) Dragonfly 蜻蜓

\* JAVASE 6.0 (1.6.0) Mustang 野马

\* JAVASE 7.0 (1.7.0) Dolphin 海豚



基

础

基

础

**JavaEE**(Java Platform Enterprise Edition)企业版

\* 是为开发企业环境下的应用程序提供的一套解决方案,该技术体系中包含的技术如 Servlet、Jsp等，主要针对于Web应用程序开发

**JavaSE**(Java Platform Standard Edition)标准版

\* 是为开发普通桌面和商务应用程序提供的解决方案,该技术体系是其他两者的基础，可以完成一些桌面应用程序的开发。

**JavaME**(Java Platform Micro Edition)小型版

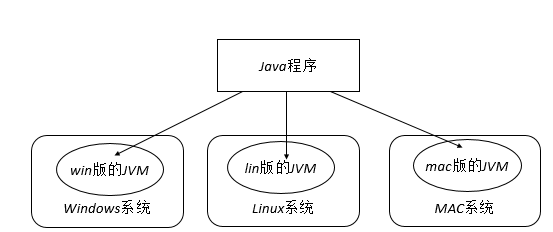
\* 是为开发电子消费产品和嵌入式设备提供的解决方案

## 1.2 跨平台原理

平台：指的是操作系统(Windows，Linux，Mac)

跨平台：Java程序可以在任意操作系统上运行，一次编写到处运行

原理：实现跨平台需要依赖Java的虚拟机 JVM （Java Virtual Machine）



## 1.3 JVM JRE JDK

A:什么是JVM

JVM是java虚拟机(JVM Java Virtual Machine)，java程序需要运行在虚拟机上，不同平台有自己的虚拟机，因此java语言可以跨平台

B:什么是JRE

包括Java虚拟机(JVM Java Virtual Machine)和Java程序所需的核心类库等如果想要运行一个开发好的Java程序，计算机中只需要安装JRE即可。

JRE:JVM+类库。

C:什么是JDK

JDK是提供给Java开发人员使用的，其中包含了java的开发工具，也包括了JRE。所以安装了JDK，就不用在单独安装JRE了。

其中的开发工具：编译工具(javac.exe)  打包工具(jar.exe)等

  JDK:JRE+JAVA的开发工具。

D:为什么JDK中包含一个JRE

为什么JDK中包含一个JRE呢？

开发完的程序，需要运行一下看看效果。

E:JDK,JRE,JVM的作用和关系

JDK包含JRE 和开发工具包

JRE 包含 核心类库和JVM

## 1.4 JDK下载

请参考《JDK下载文档.doc》安装步骤

## 1.5 Java开发环境搭建

### 1.5.1 JDK安装

安装JDK (演示)

傻瓜式安装，下一步即可。

建议：安装路径不要有中文或者特殊符号如空格等。开发工具最好安装目录统一，最好和老师一样。

### 1.5.2 环境变量配置及测试

* 如何配置
  1. 创建新的变量名称：JAVA\_HOME
     1. 计算机-右键属性-高级系统设置-高级-环境变量-系统变量
  2. 为JAVA\_HOME添加变量值：JDK安装目录
  3. 修改path环境变量
     1. 在path环境变量最前面添加：%JAVA\_HOME%\bin;
* 如何测试
  1. 在控制台输入java和javac能看到和我一样的内容
  2. 如何打开控制台
     1. win+R,输入cmd，然后回车

第2章 Eclipse开发工具

## 2.1 Eclipse概述和安装

Eclipse是一个IDE(集成开发环境)IDE(Integrated Development Environment)

集成了代码编写功能，分析功能，编译功能，调试功能等一体化的开发软件。

Eclipse的特点描述：免费、纯Java语言编写、免安装、扩展性强

下载和安装、下载 http://eclipse.org/

安装：绿色版、解压就可以使用(Eclipse)

## 2.2 Eclipse的基本使用

### 2.2.1 Eclipse基本操作

选择工作空间

工作空间 其实就是我们写的源代码所在的目录

用Eclipse来完成一个HelloWorld案例

A:创建Java项目：点击File或者在最左侧空白处，选择Java项目，在界面中写一个项目名称，然后Finish即可。

B:创建包：展开项目，在源包src下建立一个包com.itheima

C:创建类：在com.ithiema包下建立一个类HelloWorld

在界面中写一个类名：HelloWorld，然后finish即可。

D:编写代码：在HelloWorld类写main方法，在main方法中写

一条输出语句：我是黑马程序员，我骄傲，我自豪。

E:编译：自动编译，在保存的那一刻帮你做好了

F:运行 选择要运行的文件或者在要运行的文件内容中

右键 -- Run as - Java Application即可

### 代码案例一

**package** com.itheima;

**public** **class** HelloWorld {

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

System.***out***.println("我是黑马程序员,我骄傲,我自豪");

}

}

## 2.3 Eclipse工作空间的基本配置

A:行号的显示和隐藏

显示：在代码区域的最左边的空白区域，右键 -- Show Line Numbers即可。

隐藏：把上面的动作再做一次。

B:字体大小及颜色

a:Java代码区域的字体大小和颜色：

window -- Preferences -- General -- Appearance -- Colors And Fonts -- Java -- Java Edit Text Font

b:控制台

window -- Preferences -- General -- Appearance -- Colors And Fonts -- Debug -- Console font

c:其他文件

window -- Preferences -- General -- Appearance -- Colors And Fonts -- Basic -- Text Font

C:窗体给弄乱了，怎么办?

window -- Perspective -- Reset Perspective

D:控制台找不到了，怎么办?

Window--Show View—Console

## 2.4 Eclipse中项目的删除和导入

A:删除项目

选中项目 – 右键 – 删除

从项目区域中删除

从硬盘上删除

B:导入项目

在项目区域右键找到import

找到General，展开，并找到

Existing Projects into Workspace

点击next,然后选择你要导入的项目

注意：这里选择的是项目名称

第3章 语法格式

## 3.1 注释

### 3.1.1注释概述

A: 什么是注释

* 用于解释说明程序的文字

B: Java中注释分类

单行注释

* + 格式： //注释文字

多行注释

* + 格式： /\* 注释文字 \*/

文档注释

* + 格式：/\*\* 注释文字 \*/

C: 注释的作用

a:解释说明程序

b:帮助我们调试错误

### 3.1.2 案例代码二

package com.itheima;

/\*

\* 注释：用于解释说明程序的文字

\*

\* 分类：

\* 单行

\* 多行

\*

\* 作用：解释说明程序，提高程序的阅读性。

\*/

//这是我的HelloWorld案例，class用于定义类。

public class HelloWorld {

/\*

\* 这是main方法

\* main方法是程序的入口方法

\* 所有程序的执行都是从main方法开始的

\*/

public static void main(String[] args) {

//这是输出语句。

System.out.println("我是黑马程序员,我骄傲,我自豪");

}

}

## 3.2 关键字

### 3.2.1 关键字概述

* + 被Java语言赋予特定含义的单词

### 3.2.2 关键字特点

* + 组成关键字的字母全部小写
  + 常用的代码编辑器,针对关键字有特殊的颜色标记，非常直观，所以我们不需要去死记硬背，在今后的学习中重要的关键字也会不断的出来。

### 3.2.3 案例代码三

package com.itheima;

/\*

\* 内容辅助键：alt+/

\* main方法：main,然后alt+/,回车

\* 输出语句：syso,然后alt+/,回车

\*

\* 关键字：被Java语言赋予了特定含义的单词。

\*

\* 特点：

\* A:组成关键字的字母全部小写

\* B:常见的代码编辑器，对关键字有特殊的颜色标记

\*/

public class KeyWordDemo {

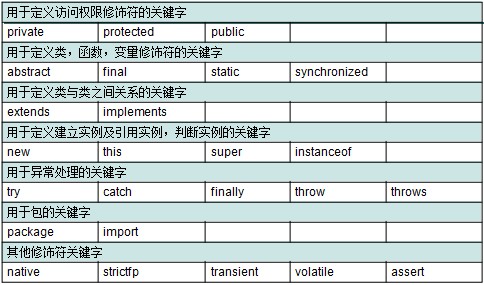
public static void main(String[] args) {

System.out.println("HelloWorld");

}

关键字举例

}



## 3.3 常量

### 3.3.1 常量概述

* + 在程序执行的过程中，其值不可以发生改变的量

### 3.3.2 常量分类

* + 字符串常量 用双引号括起来的内容(“HelloWorld”)
  + 整数常量 所有整数(12,-23)
  + 小数常量 所有小数(12.34)
  + 字符常量 用单引号括起来的内容(‘a’,’A’,’0’)
  + 布尔常量 较为特有，只有true和false
  + 空常量 null(数组部分讲解)

### 3.3.3 案例代码四

/\*

常量：在程序执行的过程中，其值不可以发生改变的量

常量分类：

A:字符串常量 "HelloWorld"

B:整数常量 12,-23

C:小数常量 12.34

D:字符常量 'a','0'

E:布尔常量 true,false

F:空常量 null(后面讲解)

\*/

public class ChangLiang {

public static void main(String[] args) {

//字符串常量

System.out.println("HelloWorld");

//整数常量

System.out.println(12);

System.out.println(-23);

//小数常量

System.out.println(12.34);

//字符常量

System.out.println('a');

System.out.println('0');

//布尔常量

System.out.println(true);

System.out.println(false);

}

}

## 3.4 变量

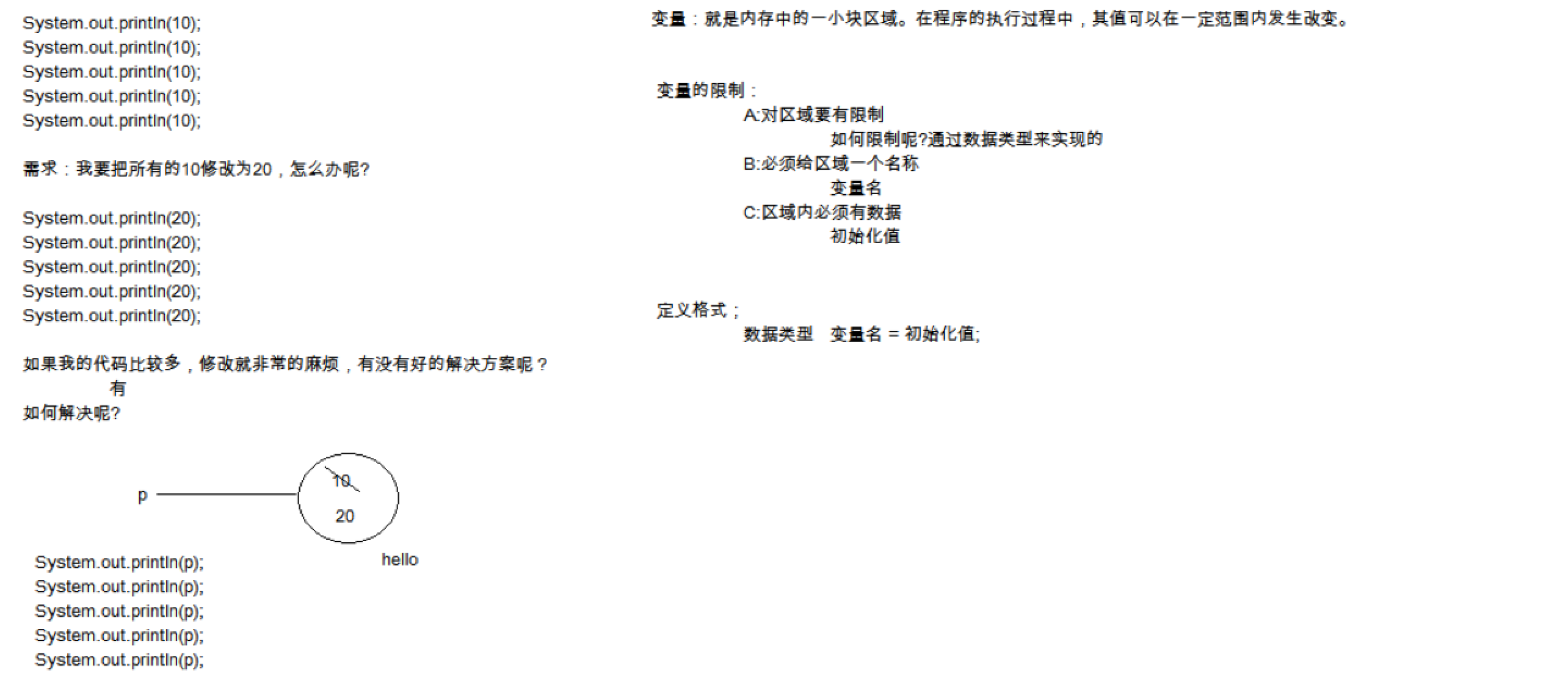
### 3.4.1 变量概述

* + 在程序执行的过程中，在某个范围内其值可以发生改变的量
  + 从本质上讲，变量其实是内存中的一小块区域

### 3.4.2 变量定义格式

* + 数据类型 变量名 = 初始化值;
  + 注意：格式是固定的，记住格式，以不变应万变

### 3.4.3 变量图解



## 3.5 数据类型

### 3.5.1 计算机存储单元

变量是内存中的小容器，用来存储数据。那么计算机内存是怎么存储数据的呢？无论是内存还是硬盘，计算机存储设备的最小信息单元叫“位（bit）”，我们又称之为“比特位”，通常用小写的字母b表示。而计算机最小的存储单元叫“字节（byte）”，通常用大写字母B表示，字节是由连续的8个位组成。

除了字节外还有一些常用的存储单位，大家可能比较熟悉，我们一起来看看：

* + 1B（字节） = 8bit
  + 1KB = 1024B
  + 1MB = 1024KB
  + 1GB = 1024MB
  + 1TB = 1024GB

### 3.5.2 数据类型概述和分类

A:为什么有数据类型

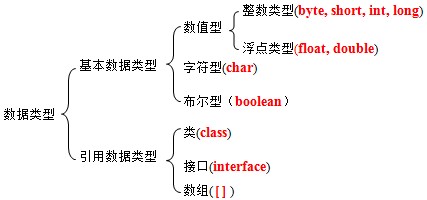
Java语言是强类型语言，对于每一种数据都定义了明确的具体数据类型，在内存中分配了不同大小的内存空间

B:Java中数据类型的分类

基本数据类型

引用数据类型

面向对象部分讲解



### 3.5.3 基本数据类型(4类8种)



## 3.6 标识符

### 3.6.1 标识符概述

A 作用

* + 给包,类,方法,变量等起名字

B 组成规则

* + 由字符，下划线\_，美元符$组成
    - 这里的字符采用的是unicode字符集，所以包括英文大小写字母，中文字符，数字字符等。
  + 注意事项
  + 不能以数字开头
  + 不能是Java中的关键字

C : 命名原则:见名知意

a包

最好是域名倒过来,要求所有的字母小写

b类或者接口

如果是一个单词首字母大写

如果是多个单词每个单词首字母大写(驼峰标识)

c方法或者变量

如果是一个单词全部小写

如果是多个单词,从第二个单词首字母大写

d常量

如果是一个单词,所有字母大写

如果是多个单词,所有的单词大写,用下划线区分每个单词

### 3.6.2 案例代码五

package com.itheima;

/\*

\* 标识符：就是用来给包，类，变量，方法等起名字的符号。

\*

\* 组成规则：

\* A:unicode字符

\* 数字字符，英文大小写字母，汉字(不建议使用汉字)

\* B:下划线\_

\* C:美元符$

\*

\* 注意事项：

\* A:不能以数字开头

\* B:不能是java中的关键字

\*/

public class IdentifierDemo {

public static void main(String[] args) {

//数据类型 变量名 = 初始化值;

int age = 20;

//不能以数字开头

//正确写法

//int b2 = 30;

//错误写法

//int 2b = 30;

//不能是java中的关键字

//int class = 40;

}

}

## 3.7 定义变量

### 3.7.1 基本数据类型变量的定义和使用

变量的定义格式：

数据类型 变量名 = 初始化值;

基本数据类型：

byte,short,int,long,float,double,char,boolean

注意：

整数默认是int类型，定义long类型的数据时，要在数据后面加L。

浮点数默认是double类型，定义float类型的数据时，要在数据后面加F。

### 3.7.2 案例代码六

package com.itheima;

/\*

\* 变量的定义格式：

\* 数据类型 变量名 = 初始化值;

\*

\* 基本数据类型：

\* byte,short,int,long,float,double,char,boolean

\*

\* 注意事项：

\* A:整数默认是int类型，定义long类型变量的时候，建议加L或l。

\* B:浮点数默认是double类型，定义float类型变量的时候，建议加F或f。

\*/

public class VariableDemo {

public static void main(String[] args) {

//byte类型的变量

byte b = 10;

System.out.println(10);

System.out.println(b);

//short类型的变量

short s = 100;

System.out.println(s);

//int类型的变量

int i = 1000;

System.out.println(i);

//long类型的变量

//long l = 10000;

//System.out.println(l);

long l = 10000000000L;

System.out.println(l);

//float类型的变量

float f = 12.34F;

System.out.println(f);

//double类型的变量

double d = 12.34;

System.out.println(d);

//char类型的变量

char ch = 'a';

System.out.println(ch);

//boolean类型的变量

boolean bb = true;

System.out.println(bb);

}

} //定义boolean类型的变量

boolean bb = false;

System.out.println(bb);

}

}

### 3.7.3 变量定义的注意事项

* 变量未赋值,不能直接使用
  + 引出变量的第二种使用格式
* 变量只在它所属的范围内有效。
  + 变量在哪对大括号内，变量就属于哪对大括号
* 一行上可以定义多个变量，但是不建议

### 3.7.4 案例代码七

package com.itheima;

/\*

\* 变量的注意事项：

\* A:变量未赋值，不能直接使用

\* B:变量只在它所属的范围内有效

\* 变量属于它所在的那对大括号

\* C:一行上可以定义多个变量，但是不建议

\*/

public class VariableDemo2 {

public static void main(String[] args) {

//定义一个变量

int a = 10;

System.out.println(a);

int b;

b = 20;

System.out.println(b);

{

//代码块

int c = 30;

System.out.println(c);

}

//System.out.println(c);

System.out.println(b);

/\*

int aa,bb;

aa = 10;

bb = 20;

System.out.println(aa);

System.out.println(bb);

\*/

int aa = 10;

int bb = 20;

System.out.println(aa);

System.out.println(bb);

}

}

## 3.8 数据类型转换

### 3.8.1 隐式数据类型转换

取值范围小的数据类型与取值范围大的数据类型进行运算,会先将小的数据类型提升为大的,再运算

### 3.8.2 案例代码八

package com.itheima;

/\*

\* +:做加法的符号。

\*

\* 类型转换：

\* 隐式转换

\* 强制转换

\*

\* 隐式转换：

\* byte,short,char -- int -- long -- float -- double

\*

\* boolean类型不参与这样的运算。

\*/

public class ConversionDemo {

public static void main(String[] args) {

//定义两个int类型的变量

int a = 10;

int b = 20;

System.out.println(a+b);

//我可以把a+b的结果进行输出，说明这个计算的结果是没有问题的

//那么，我应该也可以把这个结果接收一下。

int c = a + b;

System.out.println(c);

System.out.println("-----------------");

//定义两个变量，一个int类型，一个byte类型

int aa = 10;

byte bb = 20;

System.out.println(aa+bb);

//下面这种写法报错了

//byte cc = aa+bb;

int cc = aa + bb;

System.out.println(cc);

}

}

### 3.8.3 强制类型数据转换

强制转换的格式

\* 目标类型 变量名=(目标类型)(被转换的数据);

强制转换的注意事项

\* 如果超出了被赋值的数据类型的取值范围得到的结果会与你期望的结果不同

### 3.8.4 案例代码九

package com.itheima;

/\*

\* 强制转换：

\* 目标类型 变量名= (目标类型)(被转换的数据);

\*

\* 虽然可以做强制转换，但是不建议。因为强制转换可能会有数据的丢失。

\*/

public class ConversionDemo2 {

public static void main(String[] args) {

//定义两个变量，一个int类型，一个byte类型

int a = 10;

byte b = 20;

int c = a + b;

System.out.println(c);

byte d = 30;

byte e = (byte)(a + b);

System.out.println(e);

}

}