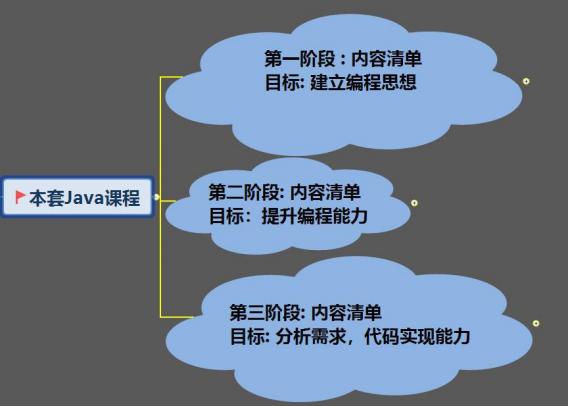
第 1 章 内容介绍

1.1 本套 Java 课程内容

1.1. 1课程的三个阶段



1.1.2关于课程的说明

关于这套课程，老韩要给小伙伴们说：

1) 以培养编程思想、提升编程能力为核心目标，达到能分析业务需求，并能用代码实现水平

2) 抛弃传统的以强灌知识点授课方式，每个阶段都有多个项目把知识点真正应用到项目中, 小伙伴在学习过程中有满

满成就感，充分体会编程乐趣

3) 课程成完整体系，超全，超详细。

4) 充分考虑零基础小伙伴，通俗易懂

5) 宁肯慢点，也不遗漏细节。[包括软件安装、开发环境配置和使用]

6) 项目是从小到大的过程，项目分解成不同的功能模块，随功能的增加,将知识点融入到项目，整个过程是循序渐进的。

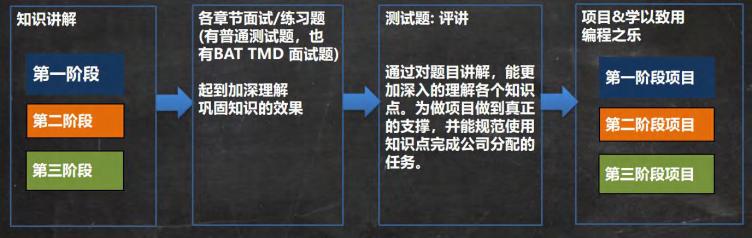
让大家轻松的学，高效的学。

7) 为减轻小伙伴学习压力，将课程分为三个阶段，小伙伴可以根据当前水平，选择从某个阶段开始学习

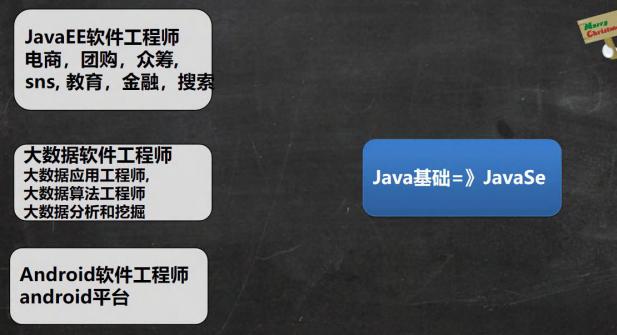
8) 零基础的小伙伴，老韩强烈建议从第一阶段开始学习.

9) 因为课程内容非常全，录制周期长，老韩也会根据实际情况做微调

1.1.3课程特色



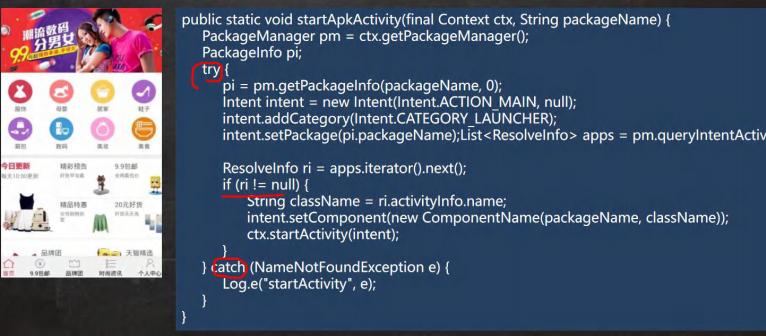
1.2 Java 就业方向



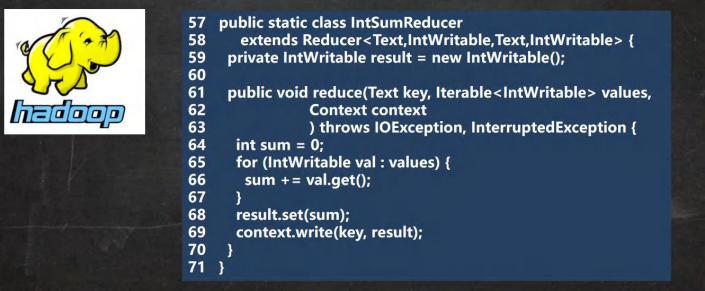
1.3 Java 开发场景举例 1-SSM



1.4 Java 开发场景举例 2：Android 核心代码



1.5 Java 开发场景举例 3：大数据-hadoop



1.6 Java 的应用领域

1.6. 1企业级应用

主要指复杂的大企业的**软件系统**、各种类型的**网站**。应用领域包括金融、电信、交通、电子商务等。



1.6.2Android 平台应用

Android 应用程序使用 Java 语言编写。Android 开发水平的高低很大程度上取决于 Java语言核心能力是否扎实。



1.6.3移动领域应用

主要表现在消费和嵌入式领域，是指在各种小型设备上的应用，包括机顶盒、车载的大屏影音娱乐设备、汽车通信

设备、扫码的 POS 机等。

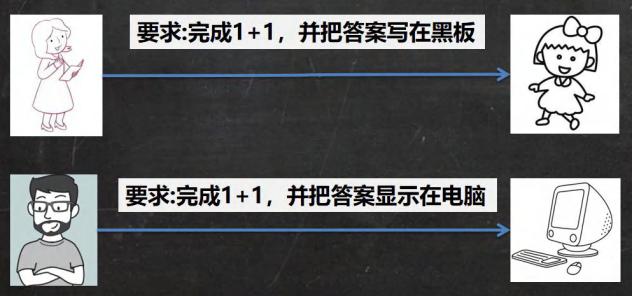


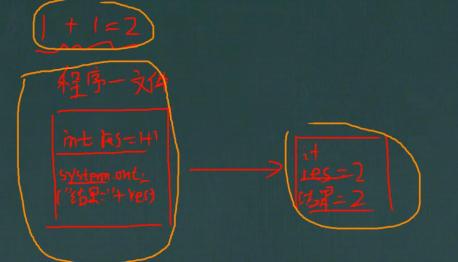
第 2 章 Java 概述

2.1 什么是程序

程序：计算机执行某些操作或解决某个问题而**编写的一系列有序指令的集合**

举例说明:



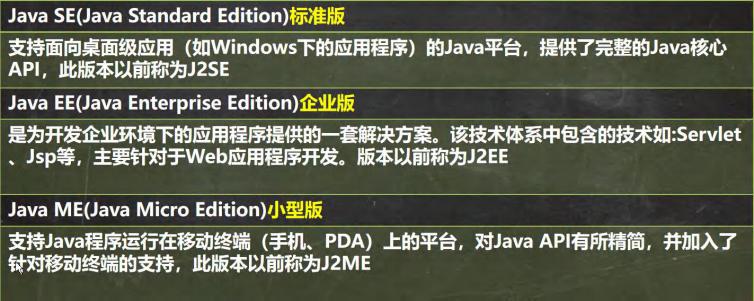


2.2 Java 诞生小故事





2.3 Java 技术体系平台

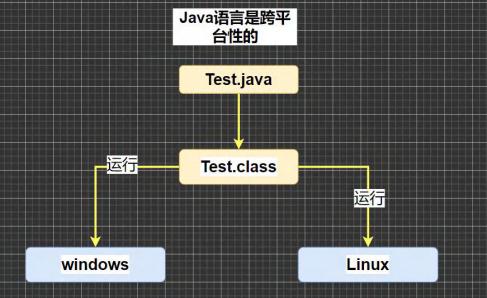


2.4 Java 重要特点

1) Java 语言是面向对象的(**oop**)

2) Java 语言是健壮的。Java 的强类型机制、异常处理、垃圾的自动收集等是 Java程序健壮性的重要保证

3) Java 语言是**跨平台性的**。[即: 一个编译好的.class 文件可以在多个系统下运行，这种特性称为跨平台]



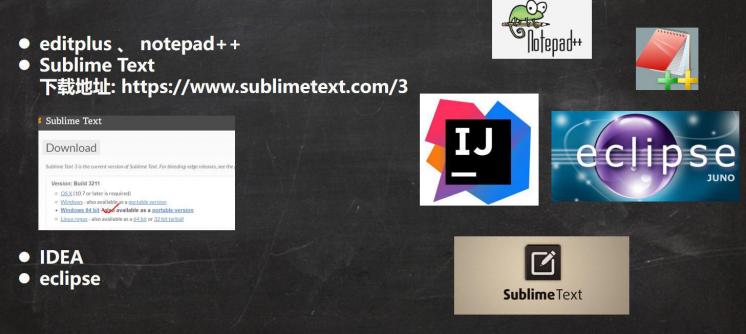
4) Java 语言是解释型的[了解]

解释性语言：javascript,PHP, java 编译性语言: c / c++

区别是：解释性语言，编译后的代码，不能直接被机器执行,需要解释器来执行, 编译性语言, 编译后的代码, 可

以直接被机器执行, c /c++

2.5 Java 的开发工具



2.5. 1工具选择

 如何选择开发工具

我们先选择用文本编辑器本 sublime ，到大家对java 有一定了解后，我们再使用 IDEA 和 Eclipse 开发工具。

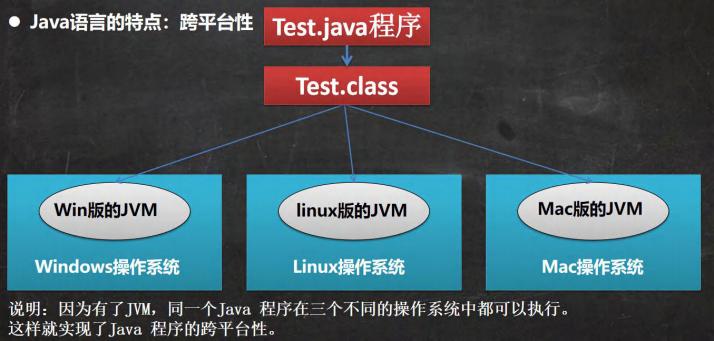
 这是为什么呢

更深刻的理解java 技术,培养代码感。【面试时，往往程序员手写代码】

有利于公司面试。

2.6 Java 运行机制及运行过程

2.6. 1Java 语言的特点：跨平台性



2.6.2Java 核心机制-Java 虚拟机 [JVM java virtual machine]

 基本介绍

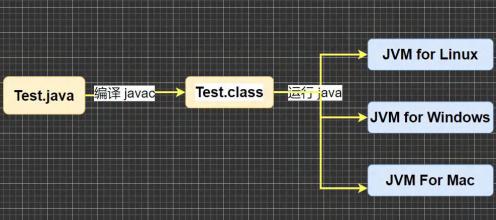
1) JVM 是一个虚拟的计算机，具有指令集并使用不同的存储区域。负责执行指令，管理数据、内存、寄存器，包含在

JDK 中.

2) 对于不同的平台，有不同的虚拟机。

3) Java 虚拟机机制屏蔽了底层运行平台的差别，实现了“一次编译，到处运行” [说明]

 示意图



2.7 什么是 JDK ，JRE

2.7. 1JDK 基本介绍

1) JDK 的全称(Java Development Kit Java 开发工具包)

JDK = JRE +java 的开发工具 [java, javac,javadoc,javap 等]

2) 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。

2.7.2JRE 基本介绍

1) JRE(Java Runtime Environment Java 运行环境)

JRE = JVM + Java 的核心类库[类]

2) 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。

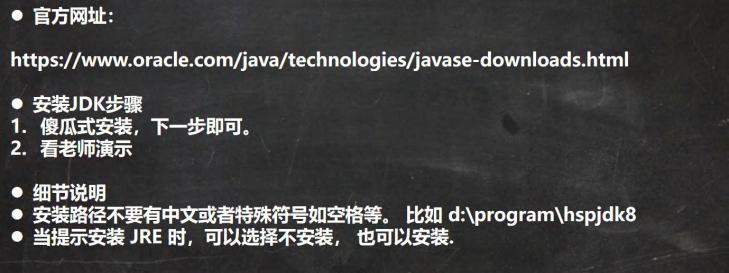
2.7.3JDK 、JRE 和 JVM 的包含关系

1) **JDK = JRE + 开发工具集**（例如 Javac,java 编译工具等)

2) **JRE = JVM + Java SE 标准类库**（java 核心类库）

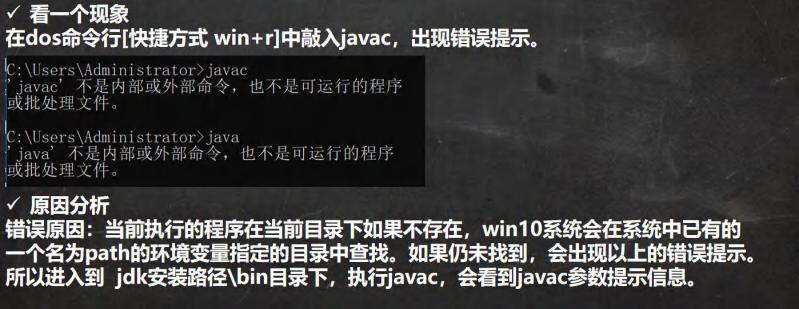
3) 如果只想运行开发好的 .class 文件 只需要 JRE

2.8 下载、安装 JDK

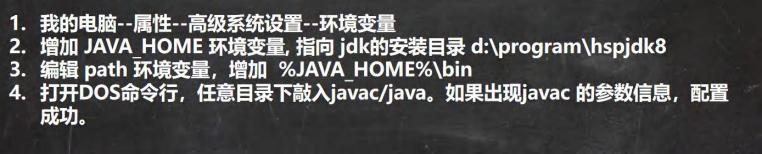


2.9 配置环境变量 path

2.9. 1为什么要配置 path

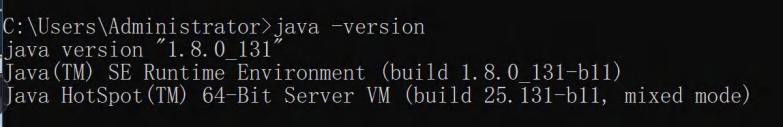


2.9.2配置环境变量 path 的步骤

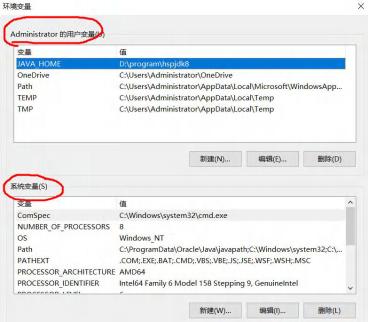


2. 10 安装 JDK 的练习

2.10.1 请同学们安装好 JDK ，配置好 path 环境变量。可以在控制台下正确的执行, (10min)



2.10.2 环境变量配置细节说明:



2.11 Java 快速入门

2.11.1 需求说明

要求开发一个 Hello.java 程序，可以输出 "hello,world!"

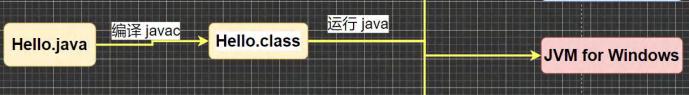
2.11.2 开发步骤

1) 将 Java 代码编写到扩展名为 Hello.java 的文件中。[代码说明]

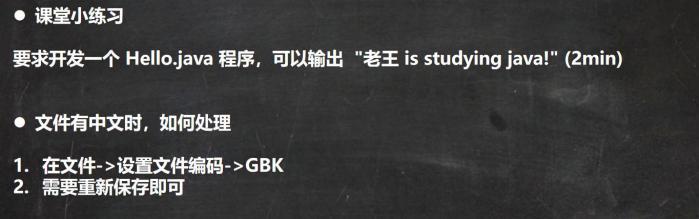
2) 通过 javac 命令对该 java 文件进行编译，生成 .class 文件。

3) 通过 java 命令对生成的 class 文件进行运行。

2.11.3 运行原理示意图



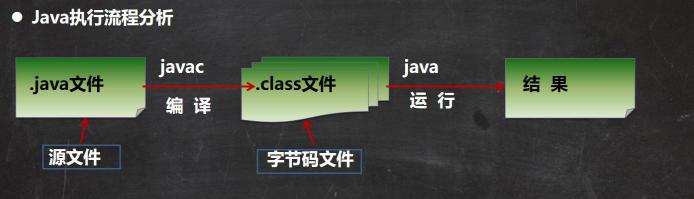
2.12 Java 开发快速入门



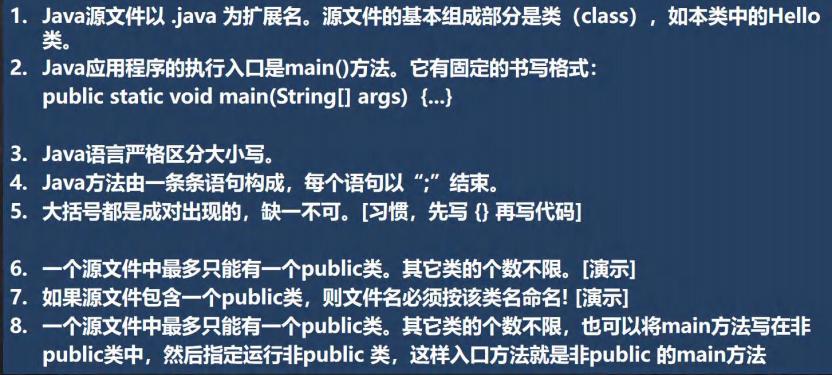
|  |
| --- |
| //这是java 的快速入门， 演示java 的开发步骤  //对代码的相关说明  //1. public class Hello 表示 Hello 是一个类,是一个 public 公有的类  //2. Hello{ } 表示一个类的开始和结束  //3. public static void main(String[] args) 表示一个主方法, 即我们程序的入口 |

|  |
| --- |
| //4. main() {} 表示方法的开始和结束  //5. System.out.println("hello,world~"); 表示输出"hello,world~"到屏幕  //6. ;表示语句结束  public class Hello {  //编写一个 main 方法  public static void main(String[] args) {  System.out.println("韩顺平教育 hello");  }  }  //一个源文件中最多只能有一个 public类。其它类的个数不限。[演示]  //Dog 是一个类  //编译后，每一个类，都对于一个.class  class Dog {  //一个源文件中最多只能有一个 public类。其它类的个数不限，也可以将 main 方法写在非 public类中，  //然后指定运行非 public 类，这样入口方法就是非 public 的 main 方法  public static void main(String[] args) {  System.out.println("hello, 小狗狗~");  }  }  class Tiger { |

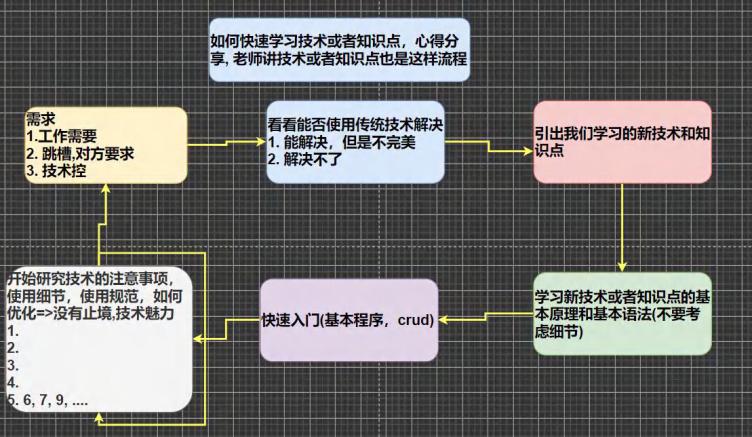
|  |
| --- |
| public static void main(String[] args) {  System.out.println("hello, 小老虎~");  }  } |



2.13 Java 开发注意事项和细节说明



2. 14 老韩聊: 如何快速掌握技术或知识点



2.15 Java 转义字符

2.15.1 Java 常用的转义字符

在控制台，输入tab 键，可以实现命令补全

\t ：一个制表位，实现对齐的功能

\n ：换行符

\\ ：一个\

\" :一个"

\' ：一个'

\r :一个回车 System.out.println("韩顺平教育\r 北京");

2.15.2 说明

应用实例

|  |
| --- |
| //演示转义字符的使用  public class ChangeChar {  //编写一个 main 方法  public static void main(String[] args) {  //\t ：一个制表位，实现对齐的功能  System.out.println("北京\t 天津\t 上海");  // \n ：换行符  System.out.println("jack\nsmith\nmary");  // \\ ：一个\ \\  System.out.println("C:\\Windows\\System32\\cmd.exe");  // \" :一个"  System.out.println("老韩说:\"要好好学习java,有前途\"");  // \' ：一个'  System.out.println("老韩说:\'要好好学习java,有前途\'");  // \r :一个回车 System.out.println("韩顺平教育\r 北京");  // 解读  // 1. 输出 韩顺平教育  // 2. \r 表示回车  System.out.println("韩顺平教育\r 北京"); // 北京平教育  }  } |

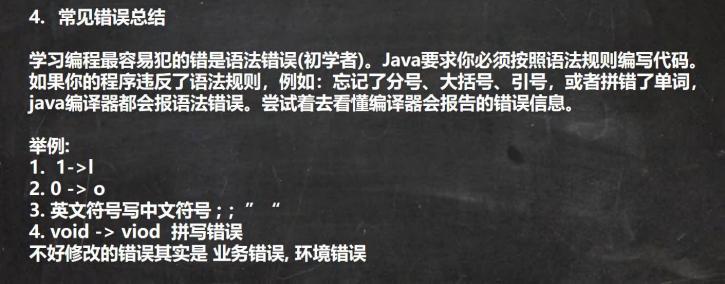
2.15.3 课堂练习

要求：请使用一句输出语句，达到输入如下图形的效果 2min

|  |
| --- |
| public class ChangeCharExer01 {  //编写一个 main 方法  public static void main(String[] args) {  //完成代码  System.out.println("书名\t 作者\t 价格\t 销量\n 三国\t 罗贯中\t120\t1000");  //示意 => 可读性很好  //下面代码完成 两个数相加  //定义变量  int n1 = 10;  int n2 = 30;  //求和  int sum = n1 + n2;  //输出结果  System.out.println("结果=" + sum);  }  } |

2. 16 初学java 易犯错误





2. 17 注释(comment)

2.17.1 介绍：

用于注解说明解释程序的文字就是注释，注释提高了代码的阅读性（可读性）；注释是一个程序员必须要具有的良

好编程习惯。将自己的思想通过注释先整理出来，再用代码去体现。 [举例]

2.17.2 Java 中的注释类型

1) 单行注释 //

2) 多行注释 /\* \*/

3) 文档注释 /\*\* \*/

2.17.3 单行注释：

基本格式

格式： //注释文字

2.17.4 多行注释：

基本格式

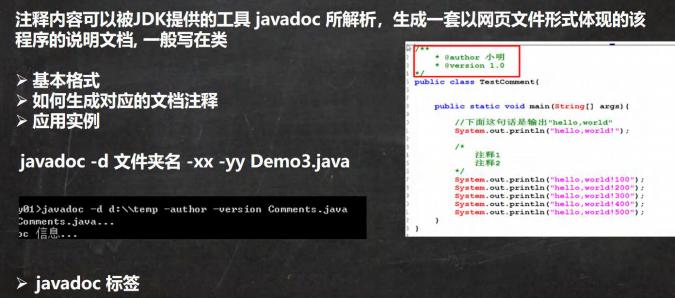
格式： /\* 注释文字 \*/

2.17.5 使用细节

1) 被注释的文字，不会被 JVM（java 虚拟机）解释执行

2) 多行注释里面不允许有多行注释嵌套

2.17.6 文档注释：



2.17.7 代码演示

|  |
| --- |
| //演示注释使用  public class Comment01 {  //编写一个 main 方法  public static void main(String[] args) {  //单行注释  //多行注释  /\* 示意 => 可读性很好  下面代码完成 两个数相加  定义变量 |

|  |
| --- |
| 注释  注释  int n1 = 10;...  System.out.println("ok~~");  \*/  int n1 = 10;  int n2 = 30;  //求和  int sum = n1 + n2;  //输出结果  System.out.println("结果=" + sum);  }  } |
| /\*\*  \* @author 韩顺平  \* @version 1.0  \*/  public class Comment02 {  //编写一个 main 方法  public static void main(String[] args) {  //选中，然后输入 tab 整体右移  //选中，然后输入 shift+tab 整体左移  int n1 = 1 + 2;  int n2 = 5 + 10; |

|  |
| --- |
| int n2 = 1 + 3 \* 34;  }  } |

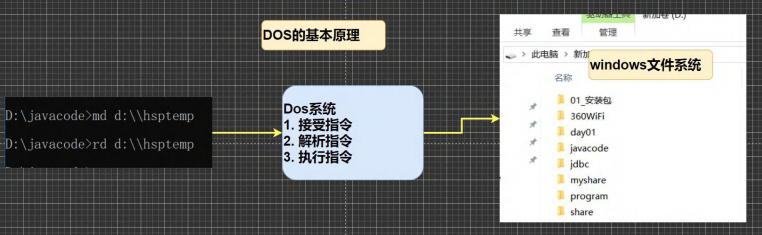
2.18 Java 代码规范



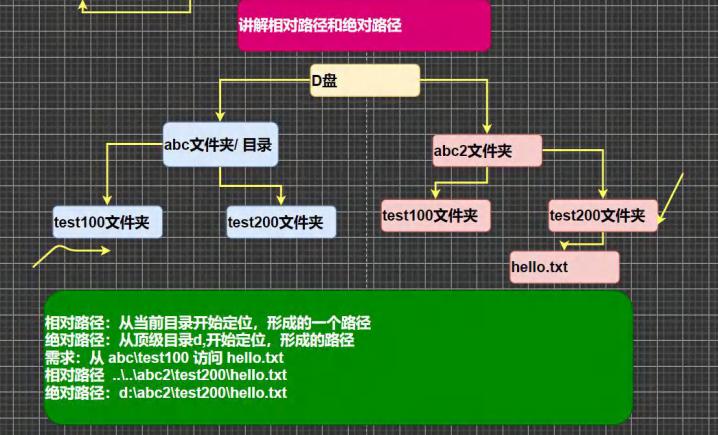
2. 19 DOS 命令(了解)

2.19.1 DOS 介绍

Dos： Disk Operating System 磁盘操作系统, 简单说一下 windows 的目录结构。[**原理图**]



2.19.2 相关的知识补充: **相对路径，** **绝对路径**



2.19.3 常用的 dos 命令

1) 查看当前目录是有什么内容 dir

dir dir d:\abc2\test200

2) 切换到其他盘下：盘符号 cd : change directory

案例演示：切换到 c 盘 cd /D c:

3) 切换到当前盘的其他目录下 (使用相对路径和绝对路径演示), ..\表示上一级目录

案例演示： cd d:\abc2\test200 cd ..\..\abc2\test200

4) 切换到上一级：

案例演示： cd ..

5) 切换到根目录：cd \

案例演示：cd \

6) 查看指定的目录下所有的子级目录 tree

7) 清屏 cls [苍老师]

8) 退出 DOS exit

9) 说明: **因为小伙伴后面使用** **DOS 非常少**，所以对下面的几个指令，老韩给大家演示下, 大家了解即可 (md[创建目

录],rd[删除目录],copy[拷贝文件],del[删除文件],echo[输入内容到文件],type,move[剪切]) => **Linux**,

2.20 本章作业



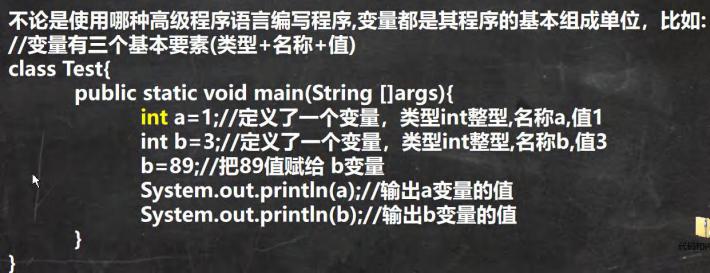
第 3 章 变量

3.1 为什么需要变量

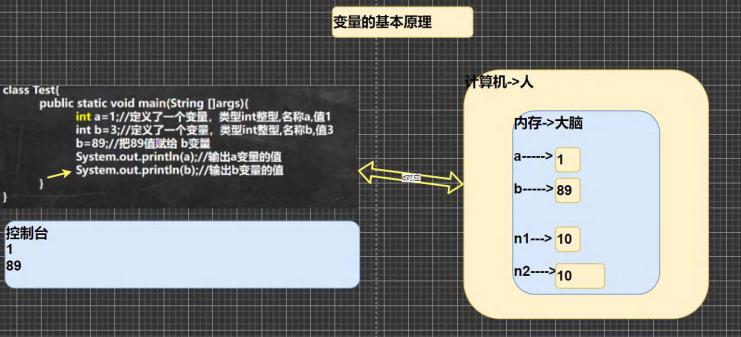
3.1.1 一个程序就是一个世界



3.1.2变量是程序的基本组成单位



3.1.3简单的原理示意图



3.2 变(变化)量(值)的介绍

3.2. 1概念

**变量相当于内存中一个数据存储空间**的表示，你可以把变量看做是一个房间的门牌号，通过**门牌号**我们可以找到房

间，而通过变量名可以访问到变量(值)。

3.2.2变量使用的基本步骤

1) 声明变量

int a;

2) 赋值

a = 60; //应该这么说: 把 60 赋给 a

使用 System.out.println(a);

//也可以一步到位[int a= 60; 通常我们是一步完成]

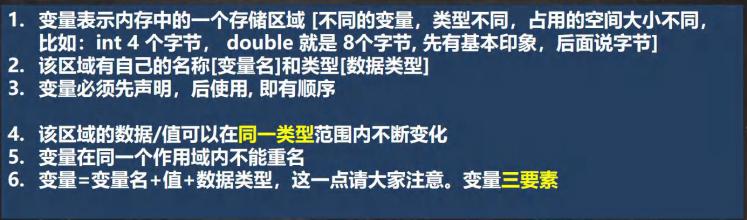
3.3 变量快速入门

变量使用入门案例

看演示并对代码进行说明， **演示记录** **人的信息的代码**

|  |
| --- |
| public class Var02 {  //编写一个 main 方法  public static void main(String[] args) {  //记录人的信息  int age = 30;  double score = 88.9;  char gender = '男';  String name = "king";  //输出信息, 快捷键  System.out.println("人的信息如下:");  System.out.println(name);  System.out.println(age);  System.out.println(score);  System.out.println(gender);  }  } |

3.4 变量使用注意事项



|  |
| --- |
| public class VarDetail {  //编写一个 main 方法  public static void main(String[] args) {  //变量必须先声明，后使用, 即有顺序  int a = 50;//int  System.out.println(a);//50  //该区域的数据/值可以在同一类型范围内不断变化  //a = "jack"; //×  a = 88; //对  System.out.println(a);//88  //变量在同一个作用域内不能重名  //int a = 77;//错误 |

|  |
| --- |
| }  }  class Dog {  public static void main(String[] args) {  int a = 666;//对  }  } |

3.5 程序中 +号的使用



3.6 数据类型

每一种数据都定义了明确的数据类型，在内存中分配了不同大小的内存空间(字节)。



|  |
| --- |
| **上图说明** **[老韩要求小伙伴，背下来!!!!]**  **1. java 数据类型分为两大类** **基本数据类型，** **引用类型**  **2. 基本数据类型有** **8 中** **数值型** **[byte , short , int , long , float ,double] char , boolean**  **3. 引用类型** **[类，接口，** **数组]** |

3.7 整数类型

3.7. 1基本介绍

Java 的整数类型就是用于存放整数值的，比如 12 , 30, 3456 等等

3.7.2案例演示：

byte n1 = 10;

short n2 = 10;

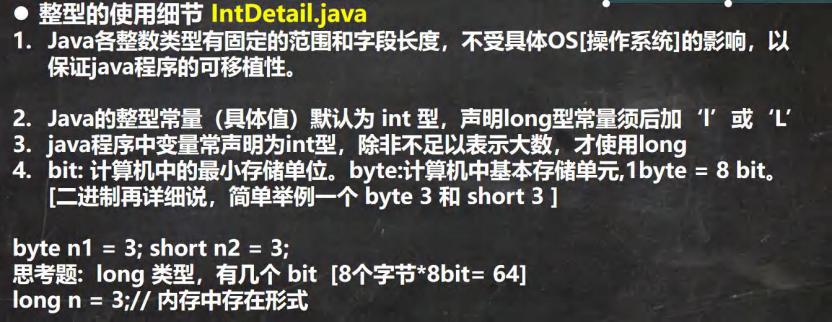
int n3 = 10;//4 个字节

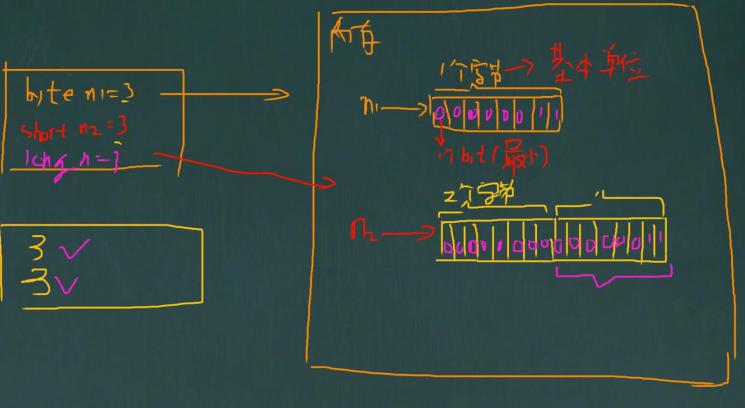
long n4 = 10; //8 个字节

3.7.3整型的类型



3.7.4整型的使用细节 IntDetail.java





|  |
| --- |
| public class IntDetail {  //编写一个 main 方法  public static void main(String[] args) {  //Java 的整型常量（具体值）默认为 int 型，声明 long 型常量须后加‘l’或‘L’  int n1 = 1;//4 个字节  //int n2 = 1L;//对不对?不对  long n3 = 1L;//对 |

|  |
| --- |
| }  } |

3.8 浮点类型

3.8. 1基本介绍

Java 的浮点类型可以表示一个小数，比如 123.4 ，7.8 ，0. 12 等等

3.8.2案例演示

3.8.3浮点型的分类

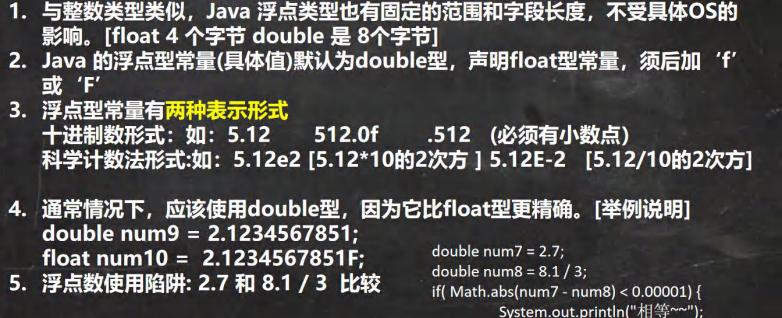


3.8.4说明一下

1) 关于浮点数在机器中存放形式的简单说明,浮点数=符号位+指数位+尾数位

2) 尾数部分可能丢失，造成精度损失(小数都是近似值)。

3.8.5浮点型使用细节 FloatDetail.java



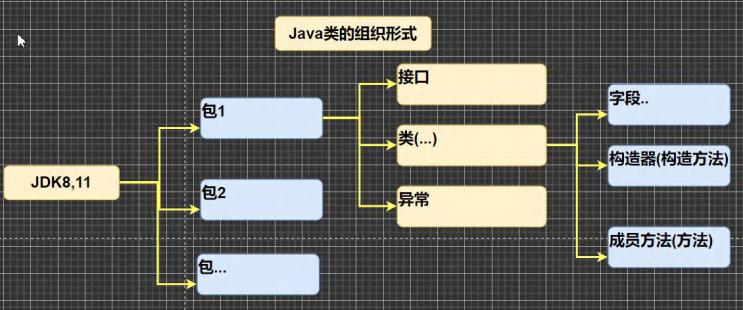
|  |
| --- |
| public class FloatDetail {  //编写一个 main 方法  public static void main(String[] args) {  //Java 的浮点型常量(具体值)默认为 double 型，声明 float 型常量，须后加‘f’或‘F'  //float num1 = 1. 1; //对不对?错误  float num2 = 1. 1F; //对的  double num3 = 1. 1; //对  double num4 = 1. 1f; //对  //十进制数形式：如：5.12 512.0f .512 (必须有小数点）  double num5 = . 123; //等价 0.123  System.out.println(num5); |

|  |
| --- |
| //科学计数法形式:如：5. 12e2 [5.12 \* 10 的 2 次方 ] 5. 12E-2 []  System.out.println(5. 12e2);//512.0  System.out.println(5. 12E-2);//0.0512  //通常情况下，应该使用 double 型，因为它比 float 型更精确。  //[举例说明]double num9 = 2. 1234567851;float num10 = 2. 1234567851F;  double num9 = 2. 1234567851;  float num10 = 2. 1234567851F;  System.out.println(num9);  System.out.println(num10);  //浮点数使用陷阱: 2.7 和 8. 1 / 3 比较  //看看一段代码  double num11 = 2.7;  double num12 = 2.7; //8. 1 / 3; //2.7  System.out.println(num11);//2.7  System.out.println(num12);//接近 2.7 的一个小数，而不是 2.7  //得到一个重要的使用点: 当我们对运算结果是小数的进行相等判断是，要小心  //应该是以两个数的差值的绝对值，在某个精度范围类判断  if( num11 == num12) {  System.out.println("num11 == num12 相等");  }  //正确的写法 , ctrl + / 注释快捷键, 再次输入就取消注释  if(Math.abs(num11 - num12) < 0.000001 ) {  System.out.println("差值非常小，到我的规定精度，认为相等..."); |

|  |
| --- |
| }  // 可以通过java API 来看 下一个视频介绍如何使用 API  System.out.println(Math.abs(num11 - num12));  //细节:如果是直接查询得的的小数或者直接赋值，是可以判断相等  }  } |

3.9 Java API 文档





3. 10 字符类型(char)

3.10.1 基本介绍

字符类型可以表示单个字符,字符类型是 char ，char 是两个字节(可以存放汉字) ，多个字符我们用字符串 String(我们

后面详细讲解 String)

3.10.2 案例演示 Char01.java

#代码

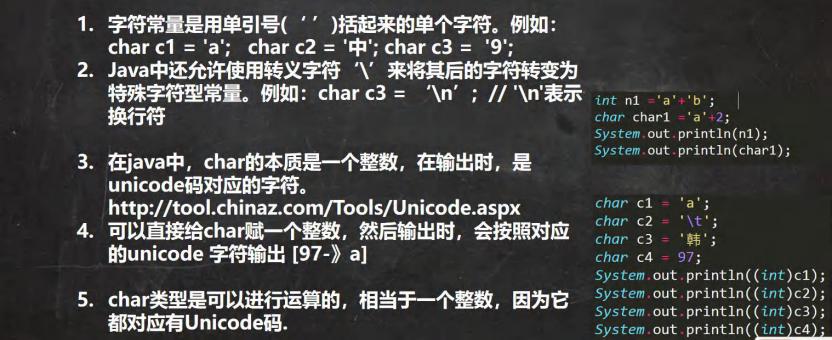
char c1 = 'a';

char c2 = '\t';

char c3 = '韩';

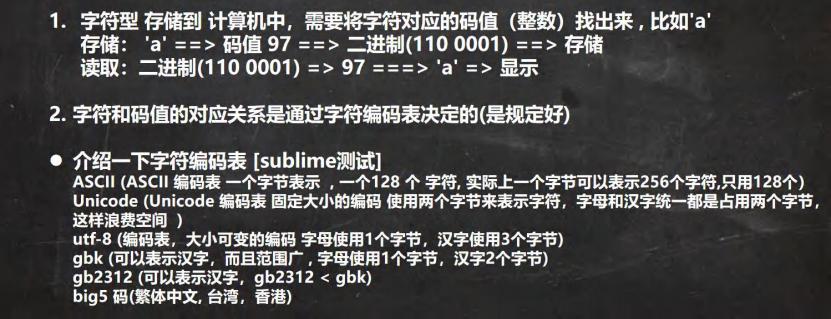
char c4 = 97;

3.10.3 字符类型使用细节

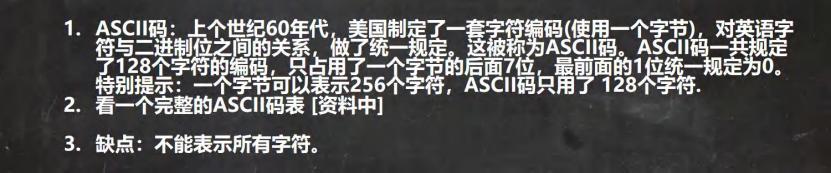


|  |
| --- |
| public class CharDetail {  //编写一个 main 方法  public static void main(String[] args) {  //在java 中，char 的本质是一个整数，在默认输出时，是 unicode 码对应的字符  //要输出对应的数字，可以(int)字符  char c1 = 97;  System.out.println(c1); // a  char c2 = 'a'; //输出'a' 对应的 数字  System.out.println((int)c2);  char c3 = '韩'; |

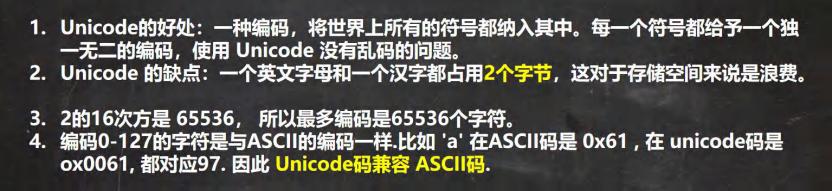
|  |
| --- |
| System.out.println((int)c3);//38889  char c4 = 38889;  System.out.println(c4);//韩  //char 类型是可以进行运算的，相当于一个整数，因为它都对应有 Unicode 码.  System.out.println('a' + 10);//107  //课堂小测试  char c5 = 'b' + 1;//98+1==> 99  System.out.println((int)c5); //99  System.out.println(c5); //99->对应的字符->编码表 ASCII(规定好的)=>c  }  } |



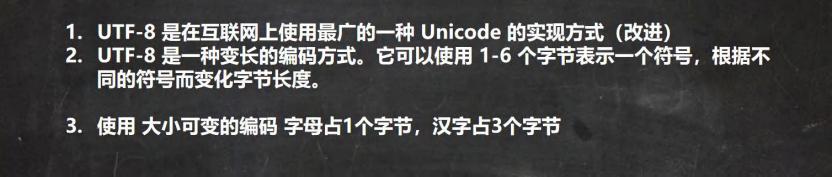
3. 11 ASCII 码介绍(了解)



3.12 Unicode 编码介绍(了解)



3.13 UTF-8 编码介绍(了解)



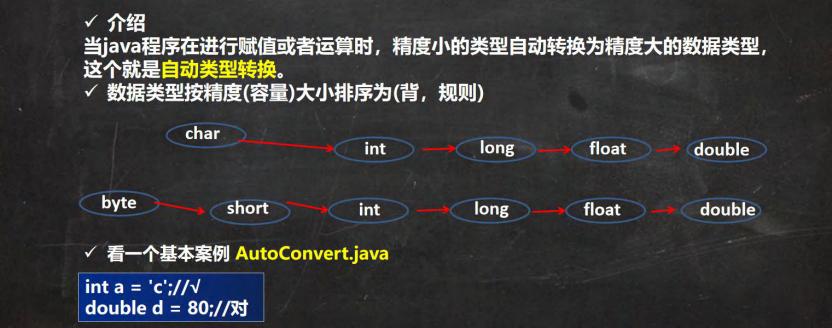
3. 14 布尔类型：boolean



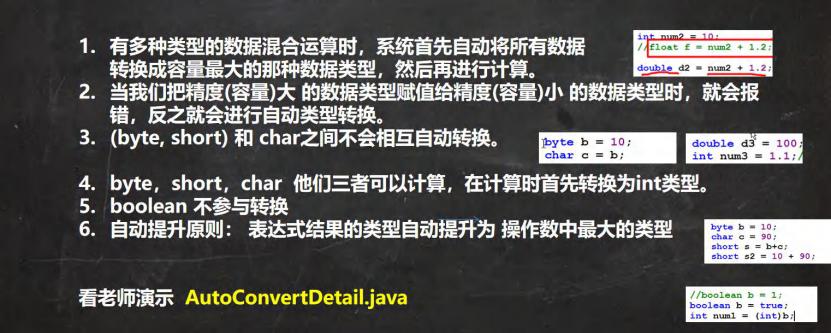
|  |
| --- |
| public class Boolean01 {  //编写一个 main 方法  public static void main(String[] args) {  //演示判断成绩是否通过的案例  //定义一个布尔变量  boolean isPass = true;//  if(isPass == true) {  System.out.println("考试通过，恭喜");  } else {  System.out.println("考试没有通过，下次努力");  }  }  } |

3. 15 基本数据类型转换

3.15.1 自动类型转换



3.15.2 自动类型转换注意和细节



|  |
| --- |
| //自动类型转换细节  public class AutoConvertDetail {  //编写一个 main 方法  public static void main(String[] args) {  //细节 1： 有多种类型的数据混合运算时，  //系统首先自动将所有数据转换成容量最大的那种数据类型，然后再进行计算  int n1 = 10; //ok  //float d1 = n1 + 1. 1;//错误 n1 + 1.1 => 结果类型是 double  //double d1 = n1 + 1. 1;//对 n1 + 1.1 => 结果类型是 double  float d1 = n1 + 1. 1F;//对 n1 + 1.1 => 结果类型是 float  //细节 2: 当我们把精度(容量)大 的数据类型赋值给精度(容量)小 的数据类型时，  //就会报错，反之就会进行自动类型转换。  //  //int n2 = 1. 1;//错误 double -> int  //细节 3: (byte, short) 和 char 之间不会相互自动转换  //当把具体数赋给 byte 时，(1)先判断该数是否在 byte 范围内，如果是就可以  byte b1 = 10; //对 , - 128- 127  // int n2 = 1; //n2 是 int  // byte b2 = n2; //错误，原因： 如果是变量赋值，判断类型  //  // char c1 = b1; //错误， 原因 byte 不能自动转成 char  //  // |

|  |
| --- |
| //细节 4: byte ，short ，char 他们三者可以计算，在计算时首先转换为 int 类型  byte b2 = 1;  byte b3 = 2;  short s1 = 1;  //short s2 = b2 + s1;//错, b2 + s1 => int  int s2 = b2 + s1;//对, b2 + s1 => int  //byte b4 = b2 + b3; //错误: b2 + b3 => int  //  //boolean 不参与转换  boolean pass = true;  //int num100 = pass;// boolean 不参与类型的自动转换  //自动提升原则： 表达式结果的类型自动提升为 操作数中最大的类型  //看一道题  byte b4 = 1;  short s3 = 100;  int num200 = 1;  float num300 = 1. 1F;  double num500 = b4 + s3 + num200 + num300; //float -> double  } |

|  |
| --- |
| } |

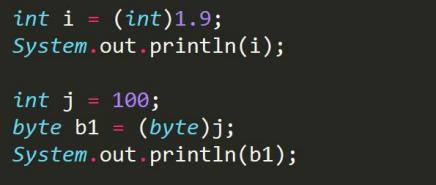
3.15.3 强制类型转换

 介绍

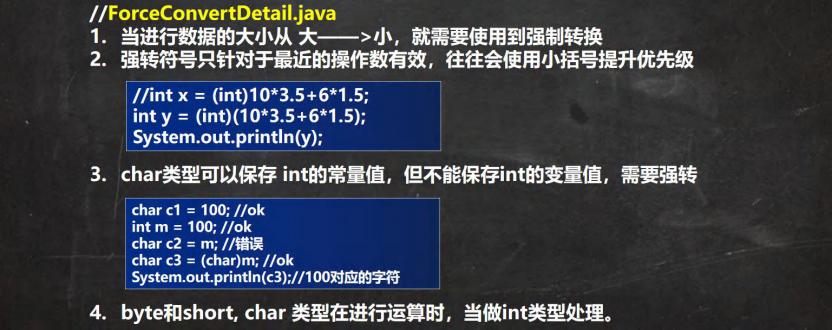
自动类型转换的逆过程，**将容量大的数据类型转换为容量小的数据类型**。使用时要加上强制转换符 ( )，但可能造成

**精度降低或溢出**,格外要注意。

 案例演示 ForceConvert.java

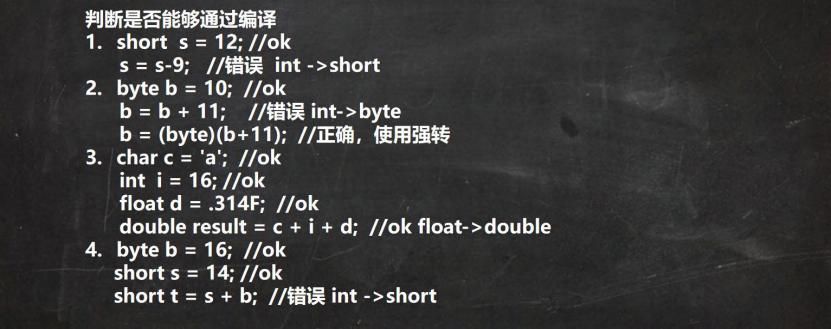


3.15.4 强制类型转换细节说明



|  |
| --- |
| public class ForceConvertDetail {  //编写一个 main 方法  public static void main(String[] args) {  //演示强制类型转换  //强转符号只针对于最近的操作数有效，往往会使用小括号提升优先级  //int x = (int)10\*3.5+6\* 1.5;//编译错误： double -> int  int x = (int)(10\*3.5+6\* 1.5);// (int)44.0 -> 44  System.out.println(x);//44  char c1 = 100; //ok  int m = 100; //ok  //char c2 = m; //错误  char c3 = (char)m; //ok  System.out.println(c3);//100 对应的字符, d 字符  }  } |

3. 16 基本数据类型转换-练习题



3. 17 基本数据类型和 String 类型的转换

3.17.1 介绍和使用



|  |
| --- |
| public class StringToBasic {  //编写一个 main 方法  public static void main(String[] args) {  //基本数据类型->String  int n1 = 100;  float f1 = 1. 1F;  double d1 = 4.5;  boolean b1 = true;  String s1 = n1 + "";  String s2 = f1 + "";  String s3 = d1 + "";  String s4 = b1 + "";  System.out.println(s1 + " " + s2 + " " + s3 + " " + s4);  //String->对应的基本数据类型  String s5 = " 123";  //会在 OOP 讲对象和方法的时候回详细  //解读 使用 基本数据类型对应的包装类，的相应方法，得到基本数据类型  int num1 = Integer.parseInt(s5);  double num2 = Double.parseDouble(s5);  float num3 = Float.parseFloat(s5);  long num4 = Long.parseLong(s5); |



|  |
| --- |
| byte num5 = Byte.parseByte(s5);  boolean b = Boolean.parseBoolean("true");  short num6 = Short.parseShort(s5);  System.out.println("===================");  System.out.println(num1);//123  System.out.println(num2);//123.0  System.out.println(num3);//123.0  System.out.println(num4);//123  System.out.println(num5);//123  System.out.println(num6);//123  System.out.println(b);//true  //**怎么把字符串转成字符** **char -> 含义是指** **把字符串的第一个字符得到**  //解读 s5.charAt(0) 得到 s5 字符串的第一个字符 ' 1'  System.out.println(s5.charAt(0));  }  } |

3.17.2 注意事项

案例演示: StringToBasicDetail.java

1) 在将 String 类型转成 基本数据类型时，要确保string类型能够转成有效的数据，比如 我们可以把 " 123" , 转成一

个整数，但是不能把 "hello" 转成一个整数

2) 如果格式不正确，就会**抛出异常，程序就会终止**， 这个问题在异常处理章节中，会处理

|  |
| --- |
| /\*\*  \* 演示字符串转基本数据类型的细节  \*/  public class StringToBasicDetail {  //编写一个 main 方法  public static void main(String[] args) {  String str = "hello";  //转成 int  int n1 = Integer.parseInt(str);  System.out.println(n1);  }  } |

3. 18 本章作业

