复习：

1. Java的技术结构

J2SE --- 标准版 --- 为一些简单的应用程序来提供解决方案

J2EE --- 企业版 --- 为企业的团队开发提供方案

J2ME --- 微型版 --- 为小型电子设备的嵌入里提供方案

1. Java的跨平台

JVM不是跨平台，但是Java是跨平台

JVM屏蔽不同操作系统之间的差异性

3. JDK、JRE、JVM

4. 入门程序

所有的Java代码写到Java文件中

class文件名和类名对应

javac 编译 java 运行

环境变量

5. 关键字

在Java中被占用有特殊含义的单词 --- 53个 2个保留字---goto, const

6. 标识符

命名规则

驼峰命名法

7. 注释

解释说明程序

// /\*\*/ /\*\*\*/ ---javdoc

8. 计算机常量/字面量---整数、小数、字符、字符串、布尔、空

9. 进制

二进制：0-1, 0b/0B---jdk1.7

八进制:0-7, 0

十进制:0-9

十六进制:0-9,A-F, 0x

其他进制转化为十进制：从低位次开始按位次乘以位次次幂，然后求和

15\*4=120

10. 变量

数据类型 变量名 = 数据;

11. 数据类型

基本类型

数值型

整数型

byte - 1个字节 --- -128~127

short - 2个字节

int - 4个字节 ---整数默认为int -1010~1010

long - 8个字节 --- 结尾需要以L作为标志 -1018~1018

浮点型

float --- 4个字节 -1038~1038 需要以F作为结尾

double --- 8个字节 -10308~10308 小数默认为double

字符型 --- char ---2个字节 --- 0-65535 --- char在内存中存储的时候是以U16编码存储的 --- ‘\u0000’- ‘\uffff’

转义字符： \t \r \n \\ \’ \”

扩展：编码---字符和数字之间的转化规则---无论是哪张码表，默认兼容西欧码表

ISO-8859-1 --- 西欧码表 --- 1个字符1个字节

gb2312 --- 2个字节1个字符 --- gbk --- 收录了常见的简体汉字以及一部分的繁体汉字

Unicode编码体系 --- utf-8 一个字符3个字节 utf-16---一个字符2个字节

布尔型--- boolean ---true/false

引用类型---数组、类、接口

12. 数据类型的转换

自动转换---小转大

强制转换

注意：在计算机中，所有的数据存储的都是补码形式

最高位是一个符号位，如果最高位为0表示是一个正数，如果为1表示是一个负数

int i = 14; --- 正数的原反补三码一致

00000000 00000000 00000000 00001110 --- 原码

00000000 00000000 00000000 00001110 --- 反码

00000000 00000000 00000000 00001110 --- 补码

int i = -14;---反码是在原码的基础上，最高位不变，其余位0<->1了；补码是在反码的基础上+1

10000000 00000000 00000000 00001110 --- 原码

11111111 11111111 11111111 11110001 --- 反码

11111111 11111111 11111111 11110010 --- 补码

注意：将-0规定为这个类型的最小值

byteb b = (byte)200;

13. 运算符

算术：+ - \* / % ++ --

byte/short/char在参与运算的时候会自动提升为int

byte b = 3;

b = (byte)(b + 3);

整数在运算完成之后结果一定是正整数

int i = 4300 / 1000 \* 1000;

不同的数字除以0的结果是不一样的 --- 整数/0 ArithmeticException - 算术异常 非零数字/0.0 非零小数/0 - Infinity - 无穷大 0.0/0 0.0/0.0 0/0.0 - NaN --Not a Number -- 非数字

-8%3 = -2 14%-4 = 2

++/--自增/自减在原基础上每次+1

int i =7;

int j = i++ + ++i;

byte/short/char可以参与++/--，算完之后结果类型依然是原来的类型

byte b = 127;

b++;

赋值：= += -= \*= /= %= &= |= ^= <<= >>= >>>=

除了=以外其他的符号要求变量必须先有值

byte b = 123;

b += 10;

byte b = 3 + 5;

byte b = 5;

b = b + 5; // b不是字面量

关系： == != > < >= <=

逻辑：& | ! ^ && ||

&：两边同真则为真

|：两边同假才为假

^：两边相同则为假

|| 在&&之前会导致&&被整个短路掉

位：& | << >> >>>

利用&来实现判断一个数字的奇偶性

<<:左移几位相当于乘以2的几次方

>>:右移几位相当于除以2的几次方 --- 对于正数补0，负数补的1

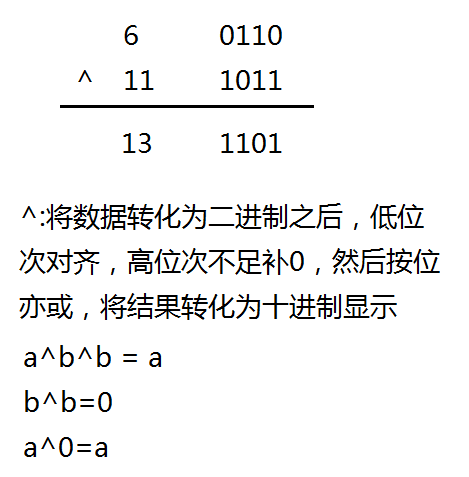
在计算左移和右移的时候，移动（位数%32）

20<<34 = 20 << (34%32) = 80

无符号右移：无论正负一律补0

~:取反。将数据转化为二进制之后，所有位1<->0 --- ~i = - i -

~(-i) = i - 1



int i = 3, j = 7;

交换i和j的值：

1. 利用一个新的变量来记录值 --- 追尾法

int temp = i;

i = j;

j = temp;

2. 加减法

i = i + j;

j = i - j;

i = i - j; -> i + j - i -> j

3. 亦或法

i = i ^ j;

j = i ^ j; -> i ^ j ^ j -> i

i = i ^ j; -> i ^ j ^ i -> j

注意：位运算只能算整数的补码形式

#### 三元运算符

逻辑值? 表达式1 :表达式2;

执行顺序：如果逻辑值为true，则执行表达式1；反之，执行表达式2

int i = 4, j = 6;

int max = i > j ? i : j;

三元表达式是一个表达式，就意味着有一个计算结果 --- 表示用一个变量来记录结果

double d = a > b ? 3.2 : 5;

double d = a > b ? 2.1 : ‘a’;

a > b ? true : 3; --- 错误

a, b, c --- 获取三个整数中的最大值

方法一：int max = a > b ? (a > c ? a : c) : (b > c ? b : c); ---三元表达式的嵌套

方法二：

int max = a > b ? a : b;

max = max > c ? max : c;

练习：

1. >=90-A >=80-B >=70-C >=60-D <60-E --- 输出分数对应的等级

2. 判断某一年是平年还是闰年

闰年:逢100整除400；其余的整除4

#### 从控制台来获取数据

需要在类之外写导包语句:import java.util.Scanner;

Scanner s = new Scanner(System.in);

int i = s.nextInt(); // 获取整数

double d =s.nextDouble(); // 获取小数

String str = s.nextLine(); // 获取字符串

### 运算符的优先级

() ~ ++ -- ! \* / % + - << >> >>> 关系 & | ^ 逻辑 三元 赋值

明日内容：流程控制

判断结构

选择结构

循环结构