复习:

1. java的技术结构

J2SE---基础版,为简单的桌面程序提供解决方案,是两种结构的基础

J2EE---企业版,为企业中的团队开发提供方案

J2ME---微型版,为一些小型电子设备的嵌入提供方案

2. java的跨平台

A. 基于JVM.---JVM不是跨平台

B. 对应不同的操作系统有不同的jvm---只涵盖主流的操作系统

3. 关键字

A. 是被java内部占用有特殊含义的单词---53个

B. goto,const---保留字

C. true/false/null---都可以作为值来使用

4. 标识符

A. 可以由字母(常见语言的基本字符)/数字/\_/$组成

B. 虽然可以使用$,但是在给类命名的时候尽量减少使用$

C. 数字不能开头

D. 不能使用关键字

E. 大小写敏感

5. 注释

A. 单行注释 //

B. 多行注释 /\*\*/

C. 文档注释 /\*\* \*/ --- 文档注释中的文字可以进行提取形成文档 --- jvadoc, 要求这个类必须是一个公共类

6. 计算机常量---字面量---看到值是多少就是多少,不会随着环境改变

整数常量

小数常量

字符常量

字符串常量

布尔常量

空常量

7. 进制---计数方式

二进制：0-1, 满2 进1, 从jdk1.7开始允许在程序中用0b/0B标志二进制

八进制：0-7，满8进1，用0开头

十进制：0-9，满10进1

十六进制:0-9，a-f，满16进1，用0x开头 9+1=a 49+1=4a

十进制转化成其他进制：转化为几进制就除以几，然后用余数倒排

15 \* 4 = 114 --- 几进制？

8. 变量

A. 基本要素：数据类型，变量名，数据

B. 变量在哪儿定义在哪儿使用，变量先定义后使用，变量先给值后使用

9. 数据类型

A. 基本类型

byte--- -128~127

short---2个字节

int---4 整数默认为int

long---8 加上L作为标志 -1018~1018

float --- 4 -1038 ~1038 要求数字以f结尾

double --- 8 小数默认为double -10308~10308 ---注意科学计数法的形式

double d = 3.2e4;

double d2 = 0x3p3; ---表示十六进制的科学计数法 3 \* 23

char --- 2 0-65535 --- utf-16---转义字符

boolean --- true/false

B. 引用类型---数组、类、接口

扩展：将字符对应的数字的规则---编码 , 记录这个规则的形式就称之为编码表---所有的码表默认兼容西欧码表---无论哪一个码表，前256个字符是按照西欧码表来对应的

ISO-8859-1 西欧码表 1个字节1个字符

gb2312 – 国标码 – gbk--- 2个字节1个字符 --- 收录了大部分的简体汉字以及一部分的繁体汉字

Unicode编码体系—收录了常见语言的常见字符：utf-8---3个字节1个字符 utf-16 – 2个字节1个字符

C. 数据类型的转换

byte -> short/char -> int -> long -> float -> double

byte b = ‘a’; --- 可以 --- ‘a’是字面量，能确定这个字面量的编码

float f = 410;

byte b = 00001000; --- 报错

127 --- 01111111 -> 0177

10. 运算符

A. 算术：+ - \* / % ++ --

byte/short/char在运算的时候会自动提升为int

byte b = 5 + 7;--- 两个字面量运算，那么在编译的时候就会去自动的计算其结果，做这句话编译完成之后就是byte b = 12;

double在运算的时候不能保证精确度---在计算机中绝大部分小数是无法精确存储的

整数运算完成之后结果一定是整数

整数/0 – ArithmeticException - 算术异常

非零小数/0 非零数字/0.0 – Infinity

0.0/0 0/0.0 0.0/0.0 – NaN

++、--在前先自增/自减，然后再运算

byte/short/char可以参与自增/自减运算，结果是原来的类型

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

除了=以外，其它符号都要求这个变量得先有值

注意连等运算

int i = 5;

i += i -= i \*= 3; i = -5

i = 5+ (5 –( 5 \* 3));

C. 关系：== != > < >= <=

D. 逻辑：& | ^ ! && ||

E. 位：& | ^ << >> >>> ~

针对的是整数的补码来运算的

正数的原反补三码一致；负数的反码是原码的基础上最高位不变，其余位0变1,1变0，补码是在反码的基础上加1

& --- 奇偶性

交换两个数的值：

方式一：第三方变量---追尾法

int temp = i;

i = j;

j = temp;

方式二：加减法

i = i + j;

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

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

方式三：亦或法

i = i ^ j;

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

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

如何去快速的确定一个数是否是2n？

注意：在计算机中所有的数据都是以补码形式存在的

练习：输入一个数字，其二进制（补码）形式中有几个1？

F. 三元：逻辑值?表达式1:表达式2

G. 运算符的优先级：() ~ ++ -- ! \* / % + - << >> >>> 关系 逻辑 & | ^ 三元 赋值

11. 流程控制

A. 判断结构：if，if-else， if-else if

B. 选择结构：switch-case

switch()的值的类型是int --- 能自动提升为int的都可以使用 --- byte/short/char, jdk1.7开始允许使用String

如果每一个case之后都有break，case的顺序不会影响结果；如果有一个case之后没有break，就会影响结果

int i = 3;

switch(i){

case 3:int j = 0;

case 0: j += 1;

case 1: j += 3;

case 2: j += 5;

default: System.out.println(j);---报错

}

// System.out.println(j); --- 报错

C. 循环结构：while，do-while，for

次数固定或者变化比较规律的时候，可以使用for

循环嵌套

练习：打印九九乘法表

1\*1=1

1\*2=2 2\*2=4

1\*3=3 2\*3=6 3\*3=9

一个循环完成九九乘法表

练习：

1. 判断一个数是否是一个质数---除了1和本身以外不能被别的数整除

n 2~n-1

2. 输入一个数字，打印它的所有的质因数

150 = 2 \* 3 \* 5 \* 5

D. break和continue

break作用于循环结构和选择结构中，表示结束当前的一层结构

continue作用于循环结构中，表示跳出当前的一层循环

12. 数组

A. 存储同一类型的数据的容器，大小固定

B. 定义格式：动态初始化和静态初始化

int[] arr = new int[5]; --- 动态初始化---给定大小但是元素未知

int[] arr = new int[]{2,4,7,1,0}; --- 静态初始化

C. 遍历数组---时间复杂度O(n)，空间复杂度o(1)

for(int i = 0; i < n; i++){

System.out.println(arr[i]);

}

D. 数组的排序--- 冒泡、选择---时间复杂度O(n2),空间复杂度o(1)

Arrays.sort---时间复杂度是O(nlogn)

E. 元素查找---元素有序，二分查找---时间复杂度O(logn)，空间复杂度o(1)

F. 数组的复制---System.arraycopy

G. 数组的反转---头尾交换

扩展：

时间复杂度---在一个程序中，将某段要重复执行的代码的时间认为是单位1，执行这个重复的代码需要多少个单位1就是时间复杂度

空间复杂度---执行一段程序所额外需要耗费的变量占用的空间，如果额外需要几个变量，就认为空间复杂度就是几 n^x logn n^xlogn

注意：空间复杂度和时间复杂度都不考虑系数

13. 二维数组

A. 存储的是一维数组

B. 获取具体元素需要两个下标

练习：杨辉三角

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

arr[i][j] = arr[i - 1][j] + arr[i - 1][j - 1];

输入一个数字，输出对应的行数