一个&、|和两个&&、||有什么区别？

1）& | 位与、位或

&& || 逻辑与、逻辑或

2）位与和位或可以用来计算数值的二进制位数运算

也可以计算true和false的运算

System.out.println(true|false);

逻辑与和逻辑或只能进行true和false的运算。

System.out.println( 10 && 3); // 错误的

3）逻辑运算符/短路运算符

逻辑运算符具有短路功能的。

短路：直接提前结束，不执行后面的判断过程。

当第一个表达式的结果可以决定整个计算结果的时候，

不会执行第二个表达式的判断过程。

&& ： 如果第一个表达式结果为false，则跳过后面所有的判断，直接返回false

|| ： 如果第一个表达式结果为true，则跳过后面所有判断，直接返回true

1.求出1-100之间的偶数之和和奇数之和（使用for循环和while循环）

//求出1-100之间的偶数之和和奇数之和（使用for循环和while循环）

public static void test1() {

int num1 = 0;

int num2 = 0;

/\*

\* //1.求出1-100之间的偶数之和和奇数之和（使用for循环和while循环）

// 1.遍历1-100 2.求出其中的偶数和奇数 3.求出偶数和奇数之和

for(int i=1;i<=100;i++){

if(i%2==0){

num1+=i;

}else{

num2+=i;

}

}

\*

\* \*/

int j = 1;

while(j <= 100) {

if(j % 2 == 0)

num1 += j;

if(j % 2 != 0)

num2 += j;

j++;

}

System.out.println("偶数之和为" + num1);

System.out.println("奇数之和为" + num2);

}

2.创建一个阶乘应用程序

功能：一个数X 的阶乘（通常记作X!）等于X\*(X-1)\*(X-2)\*.....\*1。例如4!等于4×3×2×1=24。

//4!等于4×3×2×1=24。

public static void test2(int x) {

System.out.print(x+"!"+"的阶乘是：");

int num=1;

for(int i=x;i>=1;i--){

num\*=i;

if(i!=1){

System.out.print(i+"\*");

}else{

System.out.print(i);

}

}

System.out.print("="+num);

}

3.题目：有1、2、3、4个数字，能组成多少个互不相同且无重复数字的三位数？都是多少？

程序分析：可填在百位、十位、个位的数字都是1、2、3、4。组成所有的排列后再去掉不满足条件的排列。

public static void test3() {

System.out.print("互不重复的三位数字有：");

int num=0;

for(int i=1;i<=4;i++){ //百位上可取的数字

for(int j=1;j<=4;j++){ //十位上可取的数字

for(int k=1;k<=4;k++){ //个位上可取的数字

//判断互不重复的三位数的条件

if(i!=j && i!=k && j!=k){

num++;

int number= i\*100+j\*10+k;

System.out.print(number+"\t");

}

}

}

}

System.out.println("共有："+num+"个");

}

4.打印出所有的“水仙花数”，所谓“水仙花数”是指一个三位数，其各位

数字立方和等于该数本身。例如：153是一个“水仙花数”，因为

153=1的三次方＋5的三次方＋3的三次方。

//水仙花数

public static void test4() {

int a=0;//百位上的数字

int b=0;//十位上的数字

int c=0;//个位上的数字

int num=0;

for(int i=100;i<1000;i++){

a = i/100;

b = i/10%10;

c = i%10;

if(i==Math.pow(a,3)+Math.pow(b,3)+Math.pow(c,3)){

num++;

System.out.println(i);

}

}

System.out.println("水仙花数共有："+num);

}

Math.pow(a,3);

a\*a\*a

5.打印出四种形式的九九乘法表

提示：

1、System.out.println()的功能为输出+换行

System.out.print()的功能为输出

2、在适当的位置可以使用'\t'进行对齐操作

形式1：

1\*1= 1

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

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

1\*4= 4 2\*4= 8 3\*4=12 4\*4=16

1\*5= 5 2\*5=10 3\*5=15 4\*5=20 5\*5=25

1\*6= 6 2\*6=12 3\*6=18 4\*6=24 5\*6=30 6\*6=36

1\*7= 7 2\*7=14 3\*7=21 4\*7=28 5\*7=35 6\*7=42 7\*7=49

1\*8= 8 2\*8=16 3\*8=24 4\*8=32 5\*8=40 6\*8=48 7\*8=56 8\*8=64

1\*9= 9 2\*9=18 3\*9=27 4\*9=36 5\*9=45 6\*9=54 7\*9=63 8\*9=72 9\*9=81

答：//九九乘法表

//形式一

public static void test5() {

for(int i = 1; i < 10; i++) {

for(int j = 1; j <= i; j++) {

System.out.print(i + "\*" + j + "=" + i\*j + "\t");

}

System.out.println();

}

}

形式2：

1\*9= 9 2\*9=18 3\*9=27 4\*9=36 5\*9=45 6\*9=54 7\*9=63 8\*9=72 9\*9=81

1\*8= 8 2\*8=16 3\*8=24 4\*8=32 5\*8=40 6\*8=48 7\*8=56 8\*8=64

1\*7= 7 2\*7=14 3\*7=21 4\*7=28 5\*7=35 6\*7=42 7\*7=49

1\*6= 6 2\*6=12 3\*6=18 4\*6=24 5\*6=30 6\*6=36

1\*5= 5 2\*5=10 3\*5=15 4\*5=20 5\*5=25

1\*4= 4 2\*4= 8 3\*4=12 4\*4=16

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

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

1\*1= 1

答：//形式二

public static void test6() {

for(int i = 9; i > 0; i-- ) {

for(int j = 1; j <= i; j++) {

System.out.print(j + "\*" + i + "=" + i\*j + "\t");

}

System.out.println();

}

}

形式3：

1\*1= 1

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

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

1\*4= 4 2\*4= 8 3\*4=12 4\*4=16

1\*5= 5 2\*5=10 3\*5=15 4\*5=20 5\*5=25

1\*6= 6 2\*6=12 3\*6=18 4\*6=24 5\*6=30 6\*6=36

1\*7= 7 2\*7=14 3\*7=21 4\*7=28 5\*7=35 6\*7=42 7\*7=49

1\*8= 8 2\*8=16 3\*8=24 4\*8=32 5\*8=40 6\*8=48 7\*8=56 8\*8=64

1\*9= 9 2\*9=18 3\*9=27 4\*9=36 5\*9=45 6\*9=54 7\*9=63 8\*9=72 9\*9=81

答：//形式三

public static void test7() {

for(int i=1;i<=9;i++){

for(int k=1;k<=9-i;k++){

//打印空白格

System.out.print("\t");

}

for(int j=1;j<=i;j++){

if(i\*j>=10){

System.out.print(j+"\*"+i+"="+i\*j+"\t");

}else{

System.out.print(j+"\*"+i+"= "+i\*j+"\t");

}

}

System.out.println();

}

}

形式4：

1\*9= 9 2\*9=18 3\*9=27 4\*9=36 5\*9=45 6\*9=54 7\*9=63 8\*9=72 9\*9=81

1\*8= 8 2\*8=16 3\*8=24 4\*8=32 5\*8=40 6\*8=48 7\*8=56 8\*8=64

1\*7= 7 2\*7=14 3\*7=21 4\*7=28 5\*7=35 6\*7=42 7\*7=49

1\*6= 6 2\*6=12 3\*6=18 4\*6=24 5\*6=30 6\*6=36

1\*5= 5 2\*5=10 3\*5=15 4\*5=20 5\*5=25

1\*4= 4 2\*4= 8 3\*4=12 4\*4=16

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

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

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

1\*1= 1

答：//形式四

public static void test8() {

for(int i=9;i>=0;i--){

for(int k=1;k<=9-i;k++){

//打印空白格

System.out.print("\t");

}

for(int j=1;j<=i;j++){

if(i\*j>=10){

System.out.print(j+"\*"+i+"="+i\*j+"\t");

}else{

System.out.print(j+"\*"+i+"= "+i\*j+"\t");

}

}

System.out.println();

}

}

6.判断101-200之间有多少个素数，并输出所有素数。

只能被1和它本身整除的自然数为素数(质数)

//判断101-200之间有多少个素数，并输出所有素数。

public static void test9() {

int num=0;//统计有多少因数

int num1=0;//统计共有多少素数

for(int i=101;i<=200;i++){

//除以1到它本身

for(int j=2;j<=i/2;j++){

if(i%j==0){

num++;

}

}

//当num等于0的时候就是一个素数

if(num==0){

num1++;

System.out.println(i);

}

//将num值0，进行下一个数字的因数统计

num=0;

}

System.out.println("素数共有："+num1);

}

7.求1000以内的完全数

若一个自然数，恰好与除去它本身以外的一切因数的和相等，这种数叫做完全数。

例如，6=1+2＋3

28=1＋2＋4＋7＋14

496=1+2+4+8＋16+31+62＋124

先计算所选取的整数a(a的取值1~1000)的因数，将各因数累加于m，

若m等于a，则可确认a为完全数

//求1000以内的完全数

public static void test10() {

int temp=0;

int num=0;

//1.遍历2-1000

for(int i=2;i<=1000;i++){

for(int j=1;j<i;j++){

//2.求它的因数

if(i%j==0){

temp+=j;

}

}

if(i==temp){

num++;

System.out.println(i);

}

temp=0;

}

System.out.println("1000以内完全数共有："+num+"个");

}

}