复习：

1. 原反补三码：数据在计算机中存储的是补码。对于正数，原反补三码一致。而对于负数，算出来是原码。在原码的基础上，最高位不变，其余位0<->1;补码是在反码的基础上来+1。规定把-0当作当前类型的最小值。

2. 位运算：只能针对整数的补码形式来进行计算，算出来的结果也是补码。^

3. 三元运算：逻辑值?Code1：Code2; --- 一定能定义一个变量来记录这个结果

int i = 5;

i += i -= i \*= ++i; -> -20

i = 5 + (5 - ( 5 \* 6));

## 流程控制

顺序结构---指代码是从上到下从左到右依次编译执行的

### 分支结构

#### 判断结构

if(判断条件){

Code;

}

如果判断条件成立，则执行Code；反之则不执行

if(判断条件){

Code1;

} else {

Code2;

}

如果判断条件成立，则执行Code1；反之执行Code2.

练习：

1. 获取2/3个数中的较大值

2. 输入一个分数，输出等级

if(判断条件){

Code1;

} else if(条件2){

Code2;

} else if(条件3){

Code3;

}...

练习：

1. 输入一个数字表示月份，输出这个月份对应的季节

3-5-春 6-8-夏 9-11-秋 12,1,2-冬

2. 输入一个数字表示星期，输出这个数字所对应的英文

1-Monday 2-Tuesday 3-Wednesday 4-Thursday 5-Friday 6-Saturday 7-Sunday

#### 选择结构

switch(选项){

case 情况1:Code1;break;

case 情况2:Code2;break;

case 情况3:Code3;break;

...

}

switch的()内需要添加一个选项，这个选项要求是byte/short/char/int四种类型, JDK1.7开始允许使用String（字符串）

练习：

输入一个符号(+ - \* /)，以及两个数字，根据符号运算这两个数字

\*

2

7

14

switch(){

case “+”:...

case “-”:...

}

如果case没有break，代码会从匹配的选项开始一直往下执行，直到遇到了break结束

如果每一个case之后都有break，则case的顺序不影响结果。如果有一个或者多个case之后没有break，则case的顺序影响结果

练习：

输入三个数字，分别表示年月日，输出这一年到这一天一共有多少天

2012 1 28 28

### 循环结构

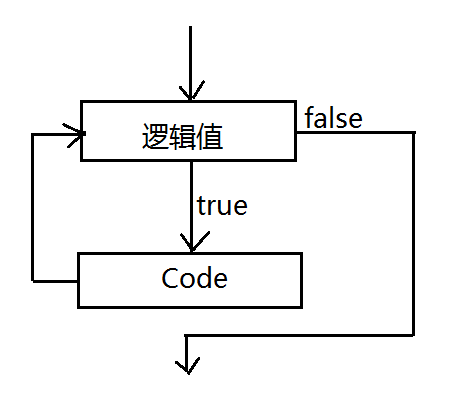
有一些相同/相似的代码需要重复执行，用的就是循环结构

while(逻辑值){

Code;

}

执行顺序：先判断逻辑值是否为true，如果为true，执行Code；执行完成Code，再次判断逻辑值是否成立。直到逻辑值为false的时候循环才结束



练习：

1. 输出200以内所有的7的倍数

2. 输出100以内所有的能被3整除而不能被7整除的数字 --- 相当于先获取100以内3的倍数，然后再判断这个数字是否能被7整除

3. 输入一个数字，求这个数字的各位上的数字的和

75235 -> 7 + 5 + 2 + 3 + 5 = 22

7523 -> 752

4. 输入正整数，然后将这个数字转化为二进制形式(字符串)来展现

6%2=0

3%2=1

1%2=1

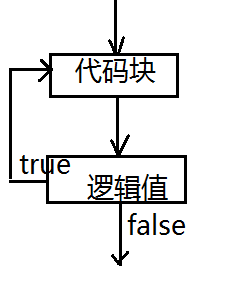
String str = “”;

str = num + str ; -> str = “110”

do{

Code;

}while(逻辑值);



特点：代码块无论如何至少要执行一次。

for循环

for(定义循环变量; 控制条件; 改变循环变量){

Code;

}

练习：

输入一个整数（Java中整数默认为int，并且存储形式是以补码形式存储），统计这个补码中有多少个1

13 --- 000000000 00000000 00000000 00001101 --- 3

-13 --- 11111111 11111111 11111111 11110011 --- 30

练习：

1. 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

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

2. 试着打印

\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*