

判断

判断

```
// 初始化
Scanner in = new Scanner(System.in);
// 读入投币金额
System.out.print("请投币: ");
int amount = in.nextInt();
if ( amount >= 10 )    如果 ( amount 大于等于 10 ), 那么
{
    // 打印车票
    System.out.println("*****");
    System.out.println("*Java城际铁路专线 *");
    System.out.println("* 无指定座位票    *");
    System.out.println("* 票价: 10元      *");
    System.out.println("*****");
    // 计算并打印找零
    System.out.println("找零: " + (amount-10) );
}
```

if语句

if语句这一行结束的时候并没有表示语句结束的“;”，而后面的赋值语句写在if的下一行，并且缩进了，在这一行结束的时候有一个表示语句结束的“;”。这表明这条赋值语句是if语句的一部分，if语句拥有和控制这条赋值语句，决定它是否要被执行。

而继续下面的其他语句。

```
if ( total > amount )
```

```
    total += amount+10;
```

```
final int MINOR = 35;

System.out.print("请输入你的年龄: ");
Scanner in = new Scanner(System.in);

int age = in.nextInt();

System.out.print("你的年龄是"+age);

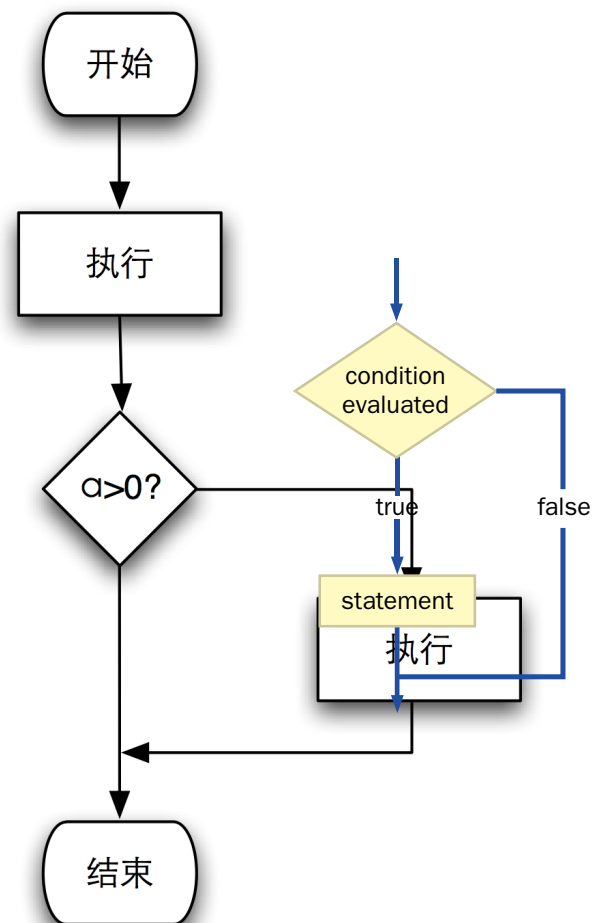
if ( age < MINOR )
{
    System.out.println("年轻是美好的, ");
}

System.out.println("年龄决定了你的精神世界, 好好珍惜吧。");
```

流程图

- 以特定的图形符号加上说明，表示算法的图，称为流程图。
- 圆角矩形表示“开始”与“结束”。
- 矩形表示行动方案、普通工作环节用
- 菱形表示问题判断或判定（审核/审批/评审）环节
- 用平行四边形表示输入输出
- 箭头代表工作流方向

流程图



```
final int MINOR = 35;
```

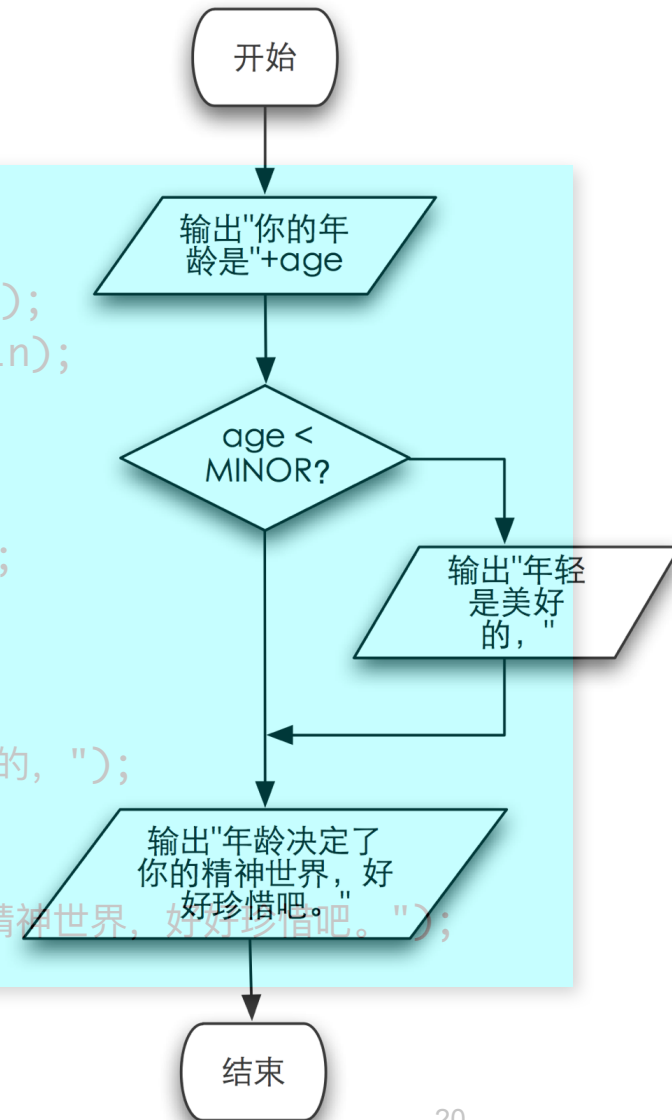
```
System.out.print("请输入你的年龄: ");  
Scanner in = new Scanner(System.in);
```

```
int age = in.nextInt();
```

```
System.out.print("你的年龄是"+age);
```

```
if ( age < MINOR )  
{  
    System.out.println("年轻是美好的, ");  
}
```

```
System.out.println("年龄决定了你的精神世界, 好好珍惜吧。");
```



比较数的大小

```
Scanner in = new Scanner(System.in);  
int x = in.nextInt();  
int y = in.nextInt();  
int max;  
if ( x > y )  
    max = x;  
System.out.println(max);
```

它没有解决 y 大于 x 的问题，当 $x > y$ 的条件不成立时，程序就结束了， max 没有得到值

方案I

```
Scanner in = new Scanner(System.in);  
int x = in.nextInt();  
int y = in.nextInt();  
int max;  
if ( x > y )  
    max = x;  
if ( y > x )  
    max = y;  
System.out.println(max);
```

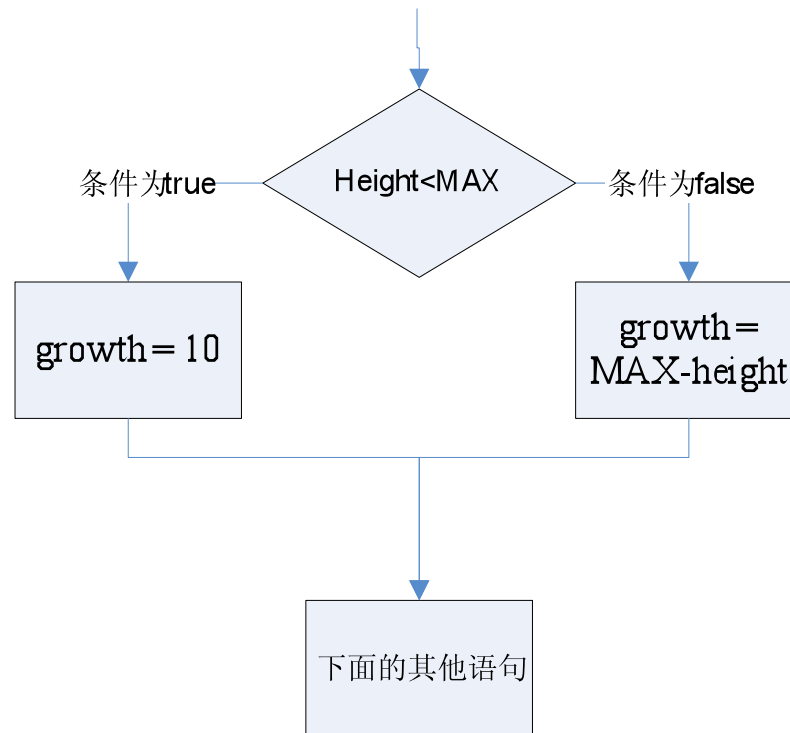
方案2

```
Scanner in = new Scanner(System.in);  
int x = in.nextInt();  
int y = in.nextInt();  
int max = y;  
if ( x > y )  
    max = x;  
System.out.println(max);
```

if-else

```
Scanner in = new Scanner(System.in);  
int x = in.nextInt();  
int y = in.nextInt();  
int max;  
if ( x > y )  
    max = x;  
else  
    max = y;  
System.out.println(max);
```

```
if ( height < MAX )  
    growth = 10;  
else  
    growth = MAX-height;
```



```
final double RATE = 8.25;
final int STANDARD = 40;
double pay = 0.0;
Scanner in = new Scanner(System.in));
System.out.print ("Enter the number of hours
worked: ");
int hours = in.nextInt();
System.out.println ();
if (hours > STANDARD)
    pay = STANDARD * RATE + (hours-STANDARD) * (RATE
* 1.5);
else
    pay = hours * RATE;
System.out.println ("Gross earnings: " + pay);
```

```
final int PASS=60;
Scanner scan = new Scanner(System.in);
System.out.print("请输入成绩: ");
int score = scan.nextInt();
System.out.println("你输入的成绩是"+score);
if ( score < PASS )
    System.out.println("很遗憾, 这个成绩没有及格。");
else
    System.out.println("祝贺你, 这个成绩及格了。");
System.out.println("再见");
```

```
final int PASS=60;
Scanner scan = new Scanner(System.in);
System.out.print("请输入成绩:");
int score = scan.nextInt();
System.out.println("你输入的成绩是"+score);
if ( score < PASS )
    System.out.println("很遗憾，这个成绩没有及格。");
else
{
    System.out.println("祝贺你，这个成绩及格了。");
    System.out.println("再见");
}
```

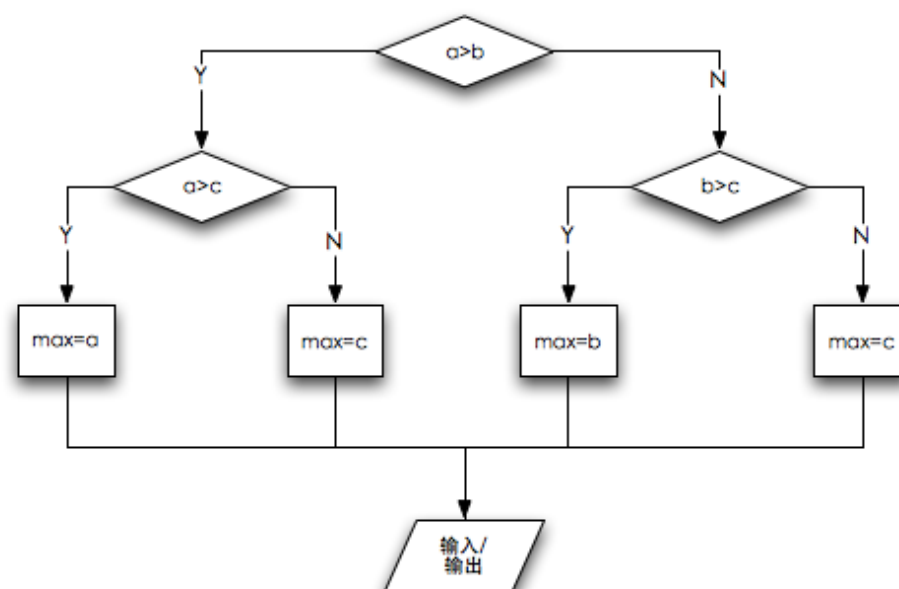
嵌套和级联的if

找三个数中的最大

```
Scanner in = new Scanner(System.in);
int a;
int b;
System.out.print("请输入两个数");
a = in.nextInt();
b = in.nextInt();
int max = 0;
if ( a>b )
{
    max = a;
}
else
{
    max = b;
}
System.out.println(max);
```

- 如果要求输入三个整数，输出其中最大的，应该怎么做？

```
if ( x>y )
{
    if ( x>z )
    {
        max = x;
    }
    else
    {
        max = z;
    }
}
else
{
    if ( y>z )
    {
        max = y;
    }
    else
    {
        max = z;
    }
}
```



嵌套的判断

- 当if的条件满足或者不满足的时候要执行的语句也可以是一条if或if-else语句，这就是嵌套的if语句

```
if ( code == READY )  
    if ( count <20 )  
        System.out.println("一切正常\n");  
else  
    System.out.println("继续等待\n");
```

else的匹配

- else总是和最近的那个if匹配

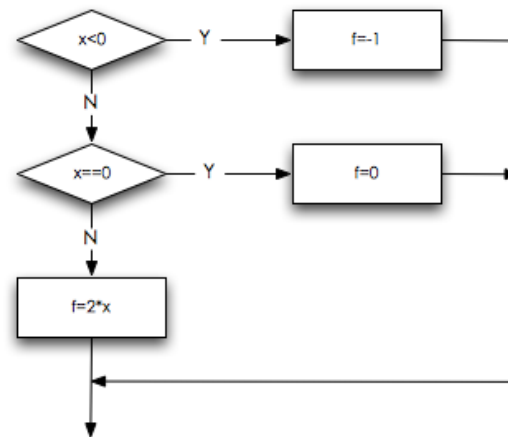
```
if ( code == READY ) {  
    if ( count <20 )  
        System.out.println("一切正常\n");  
} else  
    System.out.println("继续等待\n");
```

tips

- 在if或else后面总是用{}
- 即使只有一条语句的时候

分段函数

```
if ( x < 0 ) {  
    f = -1;  
} else if ( x == 0 ) {  
    f = 0;  
} else {  
    f = 2 * x;  
}
```



级联的if-else if

```
if ( exp1 )  
    st1;  
else if ( exp2 )  
    st2;  
else  
    st3;
```

```
int f;  
if ( x < 0 ) {  
    f = -1;  
} else if ( x == 0 ) {  
    f = 0;  
} else {  
    f = 2 * x;  
}  
System.out.println(
```

```
if ( x < 0 ) {  
    System.out.println(-1);  
} else if ( x == 0 ) {  
    System.out.println(0);  
} else {  
    System.out.println(2 * x);  
}
```

单一出口

if语句常见错误

if语句常见的错误

- 忘了大括号
- if后面的分号
- 错误使用==和=
- 使人困惑的else

忘了大括号

```
if ( age > 60 )  
    salary = salary * 1.2;  
System.out.println(salary);  
  
if ( age > 60 )  
{  
    salary = salary * 1.2;  
    System.out.println(salary);  
}
```

- 永远在if和else后面加上大括号，即使当时后面只有一条语句

if后面的分号

```
if ( age > 60 );  
{  
    salary = salary * 1.2;  
    System.out.println(salay);  
}
```

错误使用==和=

- if只要求()里的值是零或非零

```
if ( a = b )  
{  
    System.out.println("A=B");  
}
```

代码风格

- 在if和else之后必须加上大括号形成语句块；
- 大括号内的语句缩进一个tab的位置；

风格是三观...

```
if ( x < 0 ) {  
    f = -1;  
} else if ( x == 0 ) {  
    f = 0;  
} else {  
    f = 2 * x;  
}
```

```
if ( x < 0 )  
{  
    f = -1;  
} else if ( x == 0 )  
{  
    f = 0;  
} else  
{  
    f = 2 * x;  
}
```

```
if ( x < 0 )  
{  
    f = -1;  
}  
else if ( x == 0 )  
{  
    f = 0;  
}  
else  
{  
    f = 2 * x;  
}
```

- 在if和else之后必须加上大括号形成语句块；
- 大括号内的语句缩进一个tab的位置；