判断

自动售票机

- 自动售票机，选择了终点或线路之后，投入足够的纸币或硬币，就可以自动打印或制作出车票，还会自动找回零钱。

自动售票机

- 自动售票机需要用户做两个操作：选择终点或路线、投入纸币或硬币，而自动售货机则根据用户的输入做出相应的动作：打印出车票并返回找零，或告知用户余额不足以出票。

- 从计算机程序的角度看，这就是意味着程序需要读用户的两个输入，然后进行一些计算和判断，最后输出结果。

简易自动售票机

```
//  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();
//  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) );
```

注释

- 以两个斜杠“//”开头的语句把程序分成了四个部分：

注释（comment）插入在程序代码中，用来向读者提供解释信息。它们对于程序的功能没有任何影响，但是往往能使得程序更容易被人类读者理解。

/* */注释

- 延续数行的注释，要用多行注释的格式来写。多行注释由一对字符序列“/*”开始，而以“*/”结束。
- 也可以用于一行内的注释
  - int ak=47 /* 36*/, y=9;

比较

```
//  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();
System.out.println(amount >= 10);
//  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) );
```

关系运算

- 计算两个值之间的关系，所以叫做关系运算

运算符	意义
= =	相等
!=	不相等
>	大于
>=	大于或等于
<	小于
<=	小于或等于

插入

- 当两个值的关系符合关系运算符的预期时，关系运算的结果为true，否则为false
  - System.out.println(5==3);
  - System.out.println(5>3);
  - System.out.println(5<=3);

优先级

- 所有的关系运算符的优先级比算术运算的低，但是比赋值运算的高
  - 6 > 1
  - 5 == 5
  - 7 >= 3 + 4

优先级

- 判断是否相等的==和!＝的优先级比其他的低，而连续的关系运算是从左到右进行的
  - 5 > 3 == 6 > 4
  - 6 > 5 > 4
  - a == b == true
  - a == b == 6
  - a == b > false
- (a == b) > false

int vs double?

- int a = 5;
- double b = 5.0;
- System.out.println(a==b);

double?

```
double a = 1.0;
double b = 0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1;
System.out.println(a==b);
```

- Math.abs(f1 - f2) < 0.00001