java_ 3

```
内容：
流程控制语句
```

选择结构if

if条件语句分为三种语法格式：

if

if语句是指如果满足某种条件，就进行某种处理执行。

如：“如G27高铁发车时间是11点赶到，发车”。

```
乘客赶到时间为10点
      到上海游玩
```

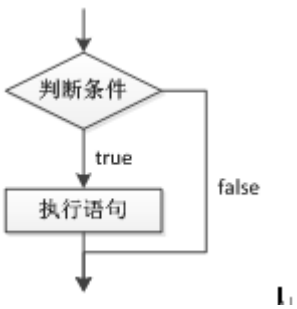
“如果”相当于Java中的关键字if，判断条件为“旅客赶到时间为10点”，需要用()括起来，“到上海游玩 ”则是执行显示语句，放在{}中，可以如这样格式：

```
if（乘客赶到时间为10点）{
   到上海游玩
}
```

描述if语句的用法，在Java中，if语句的具体语法格式如下：

```
if（条件语句){
    执行语句;
    ......
}
```

格式中，判断条件为一个boolean布尔值，当判断条件为true时，才执行{}的语句输出。if语句的执行流程图：



通过代码，了解if语句的用法：

```
    /**
     * 仅仅只是if的条件判定
     */
    public static void getHighSpeed() {
        double startTime = 11.20;//发车时间是11点
        double personTime = 9.40;//乘客赶到高铁站的时间
        if(personTime < startTime){
            System.out.println("张三到上海游玩");
        }
    }
```

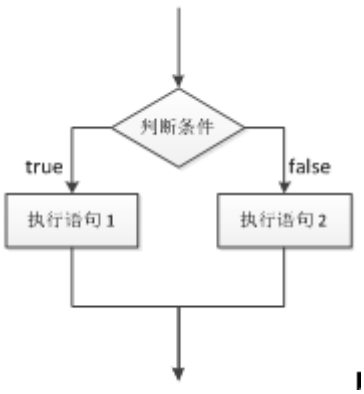
---

if ... else

if...else语句是指如果满足某种条件，就进行某种处理，否则就进行另外一种处理方式。如，判断一个正整数的奇偶，如果该数字能被2整除则是一个偶数，否则该数字就是一个奇数。if...else语句具体语法格式：

```
if (判断条件){
    执行语句1
    ......
}else{
    执行语句2
    ......
}
```

判断条件为Boolean布尔值,当判断条件为true时，执行if后面{}中的执行语句1。当判断条件为false时，执行else后面{}中的执行语句2。if...else语句的执行流程如图：



if...else代码：

```java
    /**
     * if...else(否则)
     */
    public static void getIfElse() {
        double startTime = 11.20;//发车时间是11点
        double personTime = 12.10;//乘客赶到高铁站的时间
        if(personTime < startTime){
            System.out.println("张三到上海游玩");
        }else{
            System.out.println("需要您去改签车票");
        }
    }
```

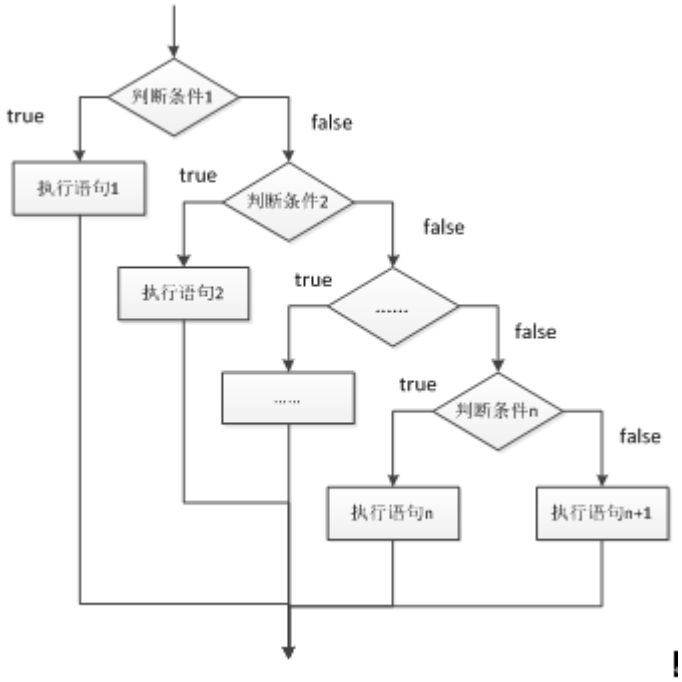
---

if ... else if ... else

if...else　if...else　语句用于对多个条件进行判断，进行多种不同的处理结果。语句具体语法格式如：

```
if (判断条件1) {
    执行语句1
} else if (判断条件2) {
    执行语句2
}
......
else if (判断条件n) {
   执行语句n
} else {
  执行语句n+1
}
```

*:判断条件是一个**boolean**(布尔值)。当判断条件1为true时，执行if后面{}的执行语句1；当判断条件1为false时，继续执行判断条件2，如为true则执行语句2，以此类推，如果所有的判断条件都为false，则意味着所有条件均没有满足都不成立，则执行最后一个else后面{}中的执行语句n+1。if...else if...else语句的执行流程如：



代码：

```
public static void getManyIfElse() {
        double startTime = 11.20;//发车时间是11点
        double personTime = 12.10;//旅客赶到高铁站的时间
        boolean bool = false;//是否改签,false为不可以改签
        if(personTime < startTime){
            System.out.println("张三到上海游玩");
        }else{
            if(bool){
                System.out.println("不可以改签车票");
            }
            if(!bool){//取反
                System.out.println("可以改签车票");
            }

        }
    }
```

工作日编写样例：

```
    /**
     * 输出规则：
     * 周一到周五   工作日
     * 周六到周日   周末
     */
    public static void getDayMethod(int day){
        //int day = 5;
        if(day == 1){
            System.out.println("今天工作日");
        }else if (day == 2) {
            System.out.println("今天工作日");
        }else if (day == 3) {
```

```
            System.out.println("今天工作日");
        }else if (day == 4) {
            System.out.println("今天工作日");
        }else if (day == 5) {
            System.out.println("今天工作日");
        }else if (day == 6) {
            System.out.println("休息日");
        }else if (day == 7) {
            System.out.println("休息日");
        }else {
            System.out.println("不是合法的数字");
        }
    }
```

---

if 与三目运算转换

三元运算符（也叫三目运算符），和if-else语句类似，语法如下：

判断条件？表达式1：表达式2

三元运算符得到一个结果，用于对某个变量进行赋值，当判断条件成立时，运算结果为表达式1的值，否则结果为表达式2的值。

用if...else语句来实现，具体代码如下：

```
/**
 * 三目运算符与if...else
 * @author djy
 * @date   2019年7月4日
 * @version
 */
public class ThreeIf {

    public static void main(String[] args) {
        compNum();
        //System.out.println(mNum);
        //判定0为男，1判定女
        /*int sexNum= 1;
        System.out.println(sexNum == 0 ? "男" : "女");*/
    }

    public static void compNum(){
        int maxNum = 0;
        int num  = 80;
        int num2 = 30;
        /*if(num > num2){
            maxNum = num;
        }else{
            maxNum = num2;
```

```
        }*/
        System.out.println(num > num2 ? (maxNum = num) : (maxNum = num2));
        //return maxNum;
    }

}
```

三目运算符代码变得简洁。

---

switch case 语句

switch（开关） case 语句判断**一个变量**与一**系列值**中某个值是否**相等**，每个值称为一个分支[多条件判定之下使用，switch case代码变得简洁]。

switch case 语法格式

```
switch(expression){//表达式
    case value ://语句
        break; //可选
    case value : //语句
        break; //可选
    //任意数量的case语句
    default : //可选 //语句
}
```

switch case 语句规则

```
- switch 语句中的变量类型可以是： int ，byte、short、char，String 型。
- switch 语句可以拥有多个 case 语句。每个 case 后面跟一个比较的值和冒号。
- case 语句中的值的数据类型必须与定义的变量的数据类型要相同，且只能是常量或者字面常量。
- 当变量的值与 case 语句的值相等时，则执行 case 语句之后的语句，直到 遇到break 语句，跳出 switch 语句方法。
- 当遇到 break 语句时，switch 语句则终止。程序跳转到 switch 语句后面的输出语句。case 语句不是必须要包含 break 语句（最好设置有brea终止）。如果没有 break 语句出现，程序会继续执行下一条 case 语句，直到出现 break 语句。
- switch 语句可以包含一个 default 分支，该分支一般是 switch 语句的最后一个分支（可以在任何位置，尽量放置在最后一个，与最终的else方式一样）。default 在没有 case 语句的值和变量值相等的时候才执行输出。default 分支不需要 break 语句。
```

switch case 执行时，会先进行匹配，匹配成功返回当前 case 的值，再根据是否有 break，判断是否继续输出，或是跳出判断。

switch case 编写例

如，使用数字1~5来表示“工作日”，6至7是休息日，根据某个输入的数字来输出对应中文格式的星期值，格式描述如下：

```
用于表示星期的数字
    如果等于1，则输出“工作日”
    如果等于2，则输出“工作日”
    如果等于3，则输出“工作日”
    如果等于4，则输出“工作日”
    如果等于5，则输出“工作日”
    如果等于6，则输出“休息日”
    如果等于7，则输出“休息日”
```

与前面的学过的 if ... else if ... else 语句也可实现，但这样判断条件较多，代码编写过长，造成代码冗余。可以使用Java中的switch语句达到简洁方式。使用switch关键字描述一个表达式，使用case关键字来描述和表达式结果比较的目标值，当表达式的值和某个目标值匹配时，则执行对应的case下的语句。

switch语句的基本语法格式,用伪代码描述switch语句语法格式如下：

```
switch (表达式){
    case 目标值1:
       执行语句1
       break;
    case 目标值2:
        执行语句2
        break;
    ......
    case 目标值n:
        执行语句n
        break;
    default:
        执行语句n+1
        break;
}
```

代码编写：

```
    public static void showDayWeek(int day) {
        switch(day){
            case 1:
                System.out.println("今天是工作日");
                break;
            case 2:
                System.out.println("今天是工作日");
                break;
            case 3:
                System.out.println("今天是工作日");
                break;
            case 4:
                System.out.println("今天是工作日");
                break;
            case 5:
                System.out.println("今天是工作日");
                break;
            case 6:
                System.out.println("休息日");
```

```
                break;
            case 7:
                System.out.println("休息日");
                break;
            default:
                System.out.println("不是合法数字");
                break;
        }

    }
```

switch语句将表达式的值与每个case中的目标值进行匹配，如果找到了匹配的值，会执行对应case后的语句，如果没找到任何匹配的值，就会执行default后的语句。switch语句中的break关键字将在后面的做具体介绍，此处，只需要知道break的作用是跳出switch语句即可。

switch数字输出中文格式的星期，代码如下：

```
public static void showWeek() {
        int week=6;
        switch(week){
            case 1:
                System.out.println("星期一");
                break;
            case 2:
                System.out.println("星期二");
                break;
            case 3:
                System.out.println("星期三");
                break;
            case 4:
                System.out.println("星期四");
                break;
            case 5:
                System.out.println("星期五");
                break;
            case 6:
                System.out.println("周六");
                break;
            case 7:
                System.out.println("周日");
                break;
            default:
                System.out.println("输入数字不正确！");
                break;
        }

    }
```

代码中，由于变量week的变量值为 6 ，整个switch语句判断的结果满足case 6的条件，因此输出“周六”，程序中的default语句用于处理和前面的case都不匹配的值，如将int week = 6代码替换为int week = 8，再次运行程序，输出“”输入数字不正确！”。

使用switch语句的过程中，如果多个case条件后面的执行语句是一样的，则该执行语句只需书写一次即可，这为一种**简写方式**。如，判断一周中的某一天是否为工作日，当输入的数字为1、2、3、4、5时就视为工作日，否则就视为休息日。

代码如下：

```
    public static void showWeekCode(int day) {
        switch(day){
            case 1:
            case 2:
            case 3:
            case 4:
            case 5:
                System.out.println("今天是工作日");
                break;//跳出switch，中止
            case 6:
            case 7:
                System.out.println("休息日");
                break;
        }
    }
```

*：break 与 continue 的**区别**[使用在for循环]