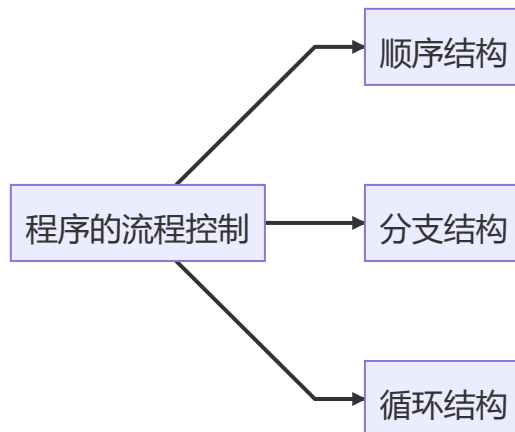


Java基础(3)¹



✧ 分支结构

if分支

接下来我们学习分支结构的第一种形式——**if分支**

if 它的作用，是用于对条件进行判断，判断的结果只可能有两个值 **true** 或者 **false**，然后根据条件判断的结果来决定执行那段代码。

if分支 的格式

if 第一种形式

```
1 // 需求：测量用户体温，发现体温高于37度就报警。
2 double temp = 38;
3 if (temp > 37) {
4     System.out.println("这个人体温不正常，快把他抓走");
5     System.out.println("这个人体温不正常，快把他抓走");
6     System.out.println("这个人体温不正常，快把他抓走");
7 }
8 System.out.println("我是代码的剩余部分");
```

if 第二种形式

```

2 // 需求1: 发红包, 你的钱包余额是99元, 现在要发出90元, 如果钱不够, 则提示: 余额不足。
3 public static void main(String[] args) {
4     // 需求2: 发红包, 你的钱包余额是99元, 现在要发出90元
5     // 如果钱够触发发红包的动作, 如果钱不够, 则提示: 余额不足。
6     System.out.println("我要开始发红包了");
7     double balance = 89;
8     if (balance > 90) {
9         balance -= 90;
10        System.out.println("你的余额是" + balance);
11    } else {
12        System.out.println("余额不足");
13    }
14    System.out.println("红包发送完毕");
15 }

```

if 第三种形式

```

1 // 需求3: 某个公司有一个绩效系统, 根据员工的打分输出对应的绩效级别。[0,60) D [60,80) C [80,90) B [90,100] A
2 public static void main(String[] args) {
3     Scanner sc = new Scanner(System.in);
4     System.out.println("请输入员工打分成绩");
5     double score = sc.nextDouble();
6     // 需求3: 某个公司有一个绩效系统, 根据员工的打分输出对应的绩效级别。
7     [0,60) D [60,80) C [80,90) B [90,100] A
8     if (score ≥ 0 && score < 60) {
9         System.out.println("员工绩效级别: D");
10    } else if (score < 80) {
11        System.out.println("员工绩效级别: C");
12    } else if (score < 90) {
13        System.out.println("员工绩效级别: B");
14    } else if (score ≤ 100) {
15        System.out.println("员工绩效级别: A");
16    } else {
17        System.out.println("你输入的成绩不合理");
18    }
19    System.out.println("绩效评定完毕");
20 }

```

switch分支

switch分支的执行流程

- 1 先执行表达式的值, 再拿着这个值去与 **case** 后的值进行匹配。

- ② 与哪个 `case` 后的值匹配为`true`就执行哪个 `case` 块的代码，遇到 `break` 就跳出 `switch` 分支。
- ③ 如果全部 `case` 后的值与之匹配都是 `false`，则执行 `default` 块的代码。



码

`javac -encoding utf-8 文件名称.java` ----以什么编码格式编译代

```

1  String week = "周三";
2  switch (week){
3      case "周一":
4          System.out.println("埋头苦干, 解决bug");
5          break;
6      case "周二":
7          System.out.println("请求大牛程序员帮忙");
8          break;
9      case "周三":
10         System.out.println("今晚啤酒、龙虾、小烧烤");
11         break;
12     case "周四":
13         System.out.println("主动帮助新来的女程序解决bug");
14         break;
15     case "周五":
16         System.out.println("今晚吃鸡");
17         break;
18     case "周六":
19         System.out.println("与王婆介绍的小芳相亲");
20         break;
21     case "周日":
22         System.out.println("郁郁寡欢、准备上班");
23         break;
24     default:
25         System.out.println("您输入的星期信息不存在~~~");
26 }

```

switch 注意事项

- 表达式类型只能是 `byte`、`short`、`int`、`char`；JDK5开始支持 `枚举`，JDK7开始支持 `String`，不支持 `double`、`float` 等
- `case` 给出的值不允许重复，且只能是字面量，不能是变量。
- 正常使用 `switch` 的时候，不要忘记写 `break`，否则会出现穿透现象。
- `switch`语句匹配的数据类型

可以自己分别用变量a、b放在switch语句中匹配试一试，如果遇到不支持的写法，编译会报错

- **case**后面的值，只能是字面量不能是变量

- **case**穿透现象

当switch语句中没有遇到break，就会直接穿透到下一个case语句执行，直到遇到break为止。

- **case**匹配多个值

当多个case语句想要执行同一段代码时，可以利用

switch分支新特性

箭头表达式

```

1  public static void main(String[] args) {
2      // 先输入年份，在输入一个月份，返回这个月多少天
3      // 31 30 29 28 31 1,3,5,7,8,10,12 30 4, 6 , 9, 11 2 闰年29天 平
   年 28天
4      Scanner sc = new Scanner(System.in);
5      System.out.println("请输入一个年份:");
6      int year = sc.nextInt();
7      System.out.println("请输入一个月份:");
8      int month = sc.nextInt();
9      // int temp = 12;
10     // switch 新特性 → 箭头表达式
11     int days = 0;
12     switch (month) {
13         case 1, 3, 5, 7, 8, 10, 12 → days = 31;
14         case 4, 6, 9, 11 → days = 30;
15         case 2 → {
16             // 判断 year 是否是闰年
17             if (year % 4 == 0 && year % 100 != 0 || year % 400
   ==0) {
18                 days = 29;
19             } else {
20                 days = 28;
21             }
22         }
23         default → days = -1;
24     }
25     if (days == -1) {
26         System.out.println("输入的月份不正确");
27     } else {

```

```

28         System.out.println("这个月" + days + "天");
29     }
30     // System.out.println(days == -1 ? "输入的月份不正确" : "这个月" +
    days + "天");
31 }

```

特性2: yield

```

1  public static void main(String[] args) {
2      // 先输入年份, 在输入一个月份, 返回这个月多少天
3      // 31 30 29 28 31 1,3,5,7,8,10,12 30 4, 6 , 9, 11 2 闰年29
    天 平年 28天
4      Scanner sc = new Scanner(System.in);
5      System.out.println("请输入一个年份: ");
6      int year = sc.nextInt();
7      System.out.println("请输入一个月份: ");
8      int month = sc.nextInt();
9      // int temp = 12;
10     // switch 新特性 yield 在 {} 中返回某个值给变量
11     // int days = 0;
12     int days = switch (month) {
13         // case temp: 错误使用
14         case 1, 3, 5, 7, 8, 10, 12 → 31;
15         case 4, 6, 9, 11 → 30;
16         case 2 → {
17             // 判断 year 是否是闰年
18             if (year % 4 == 0 && year % 100 != 0 || year % 400
    ==0) {
19                 yield 29;
20             } else {
21                 yield 28;
22             }
23         }
24         default → -1;
25     };
26     if (days == -1) {
27         System.out.println("输入的月份不正确");
28     } else {
29         System.out.println("这个月" + days + "天");
30     }
31     // System.out.println(days == -1 ? "输入的月份不正确" : "这个月" +
    days + "天");
32 }

```

if、switch如何选择



如果是对一个范围进行判断，建议使用if分支结构

如果是与一个一个的值比较的时候，建议使用switch分支结构

1. 学习Java第三天 [↩](#)