

JavaSE选择结构

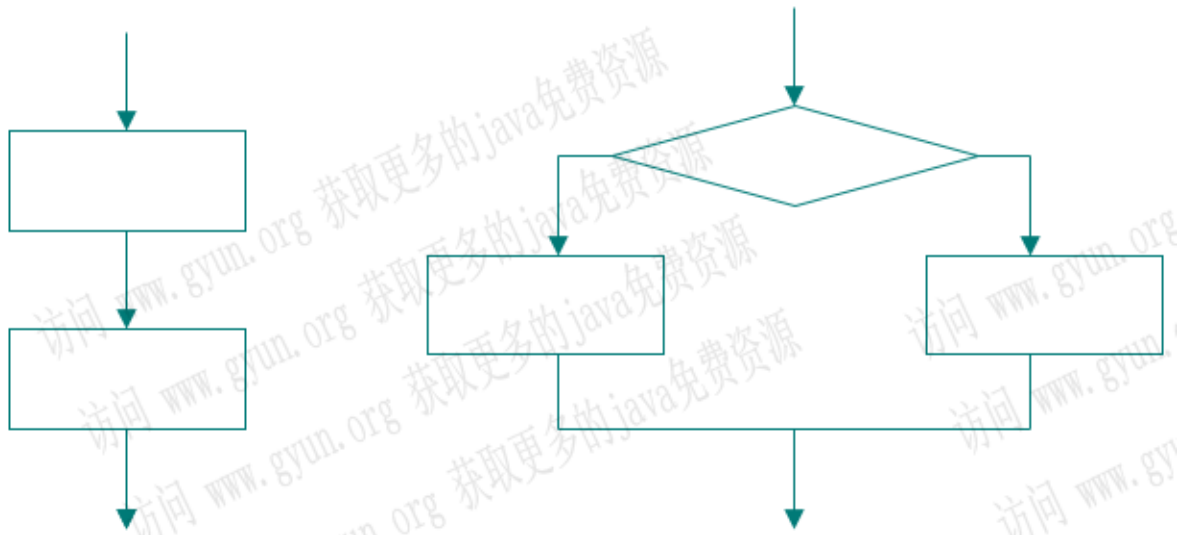
1. 什么流程控制？

- 就是用来控制程序中各语句的执行顺序。

2. 流程控制分以下三类

- 顺序(如左图)
- 选择结构(如右图)
- 循环结构(后一课讲)

如图：



3. 流程图：

逐步解决指定问题的步骤和方法的一种图形化表示方法

图 形	意 义	图 形	意 义
	程序开始或结束		判断和分支
	计算步骤/处理符号		连接符
	输入/输出指令		流程线

4. 选择结构的分类

- if语句

- switch语句

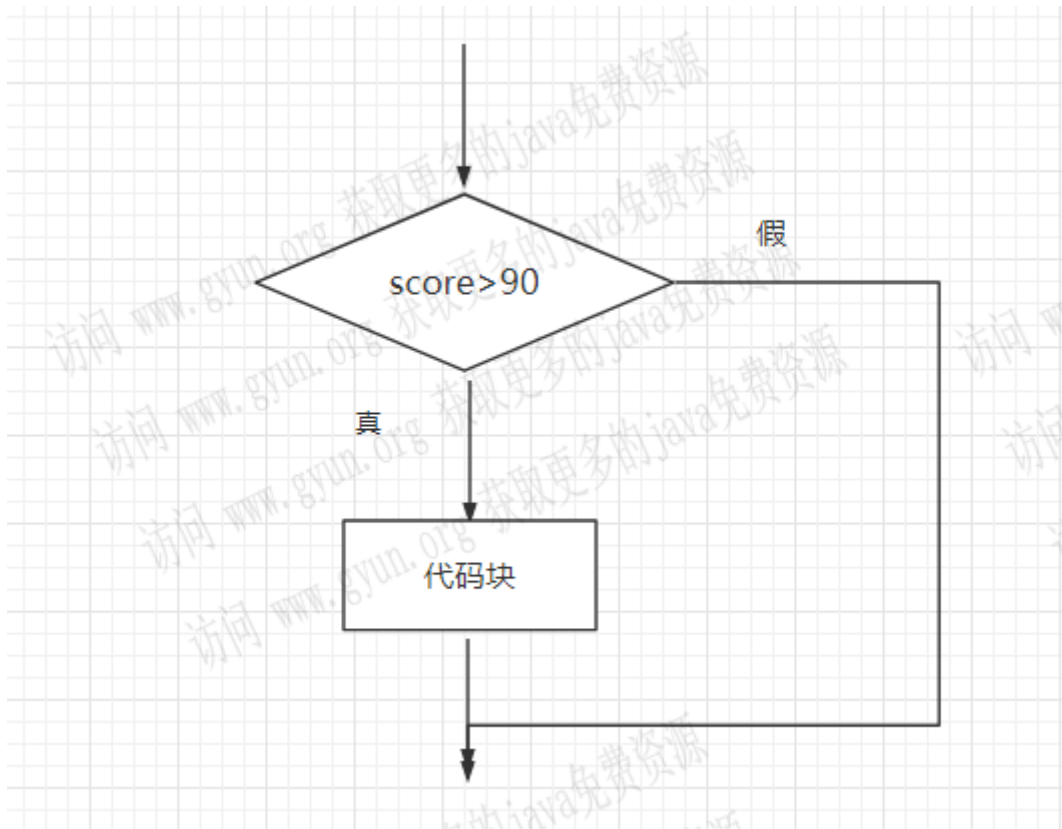
5. if语句四种格式

- 基本if选择结构：可以处理单一或组合条件的情况
- if-else选择结构：可以处理简单的条件分支情况
- 多重if选择结构：可以处理分段的条件分支情况
- 嵌套if选择结构：可以处理复杂的条件分支情况

6. switch选择结构

- 多重分支并且条件判断是等值判断的情况

7. 基本if的流程图



8. 基本if的语法：

```
if ( 条件 ) {  
    //(缩进) 代码块  
}
```

8.0.0.1. if小知识：

- 1 当有一条语句时，可以省略{}，但不推荐。
- 2
- 3 `if`后面括号的条件结果只能为`true`和`false`，如果条件为`true`，则执行代码块，否则不执行。

#`if`的应用演示代码1如下：

```
1 package org.gyun.test;
2
3 /**
4  * @ClassName: Demo1.java
5  * @Description: if的应用
6  * @Author: DongGaoYun
7  * @URL: www.gyun.org
8  * @Email: DongGaoYun@qq.com
9  * @QQ: 1050968899
10 * @Weixin: QingYunJiao
11 * @Date: 2019-9-3 下午4:25:01
12 * @Version: 1.0
13 */
14 public class Demo1 {
15
16     /**
17      * 主方法 实现需求:          *
18      * if 在一条语句的情况下，可以省略 {} ，不推荐省略！强烈要求加 {}          *
19      */
20     public static void main(String[] args) {
21         // 变量声明赋值
22         int num = 10;
23         // 判断
24         if (num > 10)//注意没有加 {} 注意缩进
25             System.out.println(num);
26         // 更新num
27         num++; // 注意:这个一行不用考虑++在前还是后，即num的值就是num+1
28         System.out.println(++num); // 打印是num自加1之后的值，即打印为: 12
29         System.out.println(num++); // 打印是num先赋值时的值，注意:不是自加1后的
30         System.out.println(num); // 打印是num自加1之后的值，即打印为: 13
31     }
32 }
```

9. `if`的应用演示代码2如下：

```
1 package org.gyun.test;
2
3 /**
4  * @ClassName: Demo2.java
5  * @Description: if复杂应用
6  * @Author: DongGaoYun
7  * @URL: www.gyun.org
8  * @Email: DongGaoYun@qq.com
9  * @QQ: 1050968899
10 * @Weixin: QingYunJiao
```

```

11  * @Date: 2019-9-3 下午4:59:36
12  * @Version: 1.0
13  */
14  public class Demo2 {
15      /**
16       * 主方法： 分析需求：
17       */
18      public static void main(String[] args) {
19          // 调用方法
20          // demo1();
21          // demo2();
22          // demo3();
23          // 需求： 如果百米赛跑在10秒内 可以根据条件进入男子和女子组
24          // 语法： if(条件){} 注意：条件的结果是布尔值，即true和false 是true,则执行
代码块，否则不执行！
25          // 声明变量
26          String sex = "男";
27          int s = 91;
28          // if判断
29          if (s < 10) { // 结果为true
30              // 注意java里比较字符串用equals
31              if ("男".equals(sex)) {
32                  System.out.println("进入男子组");
33              } else if ("女".equals(sex)) {
34                  System.out.println("进入女子组");
35              } else {
36                  System.out.println("sex未知");
37              }
38          } else { // false,就执行这一行打印
39              System.out.println("淘汰了");
40          }
41      }
42
43      // 多重if选择结构 使用多重if选择语句处理问题时需要注意顺序，切记不能任意调换。
44      private static void demo3() {
45          // 声明变量
46          int java = 91;
47          // 互斥
48          if (java >= 100) {
49              System.out.println("良好");
50          } else if (java >= 80) {
51              System.out.println("中等");
52          } else if (java >= 60) {
53              System.out.println("及格");
54          } else {
55              System.out.println("差");
56          }
57      }
58
59      // 使用if-else选择结构
60      private static void demo2() {
61          // 声明变量
62          int java = 91;
63          // 互斥
64          if (java > 90) {
65              System.out.println("给予奖励! ");
66          } else {
67              System.out.println("蹲马步");

```

```

68     }
69 }
70
71 /**
72  * 基本的if选择结构
73  *
74  * 使用复杂条件的if选择结构 如:    if (java > 90 && music > 80 || java ==
190 && music > 70) {
75  *
76  * 结合运算符的优先级编写条件**
77  *
78  * 最高的优先级: ( )
79  *
80  * 最低的优先级: =
81  *
82  * 优先级: ! > 算术运算符 > 关系运算符 > && > ||
83  *
84  * 复杂条件使用括号提高可读性
85  *
86  * 需求条件:
87  * 达成其中一个条件, 就给予奖励
88  * 1.条件1 java>90 且music>80
89  * 2.条件2 java==100 且 music>70
90  */
91 //
92 private static void demo1() {
93     // 声明变量
94     int java = 91;
95     int music = 81;
96     // 结果为true,就打印: 给予奖励!
97     if (java > 90 && music > 80 || java == 100 && music > 70) {
98         System.out.println("给予奖励! ");
99     }
100 }
101 }
102

```

10. switch的使用

```

1     switch (表达式) {
2
3         case 1:
4             //执行需求块 如: System.out.println("参加北京大学组织的1个月夏令
营");
5             //不要忘记break
6             break;
7         case 2:
8             //执行需求块 如: System.out.println("奖励外星人笔记本电脑一台 ");
9             //不要忘记break
10            break;
11        case 3:
12            //执行需求块 如: System.out.println("奖励手机一部 ");
13            //不要忘记break
14            break;
15        default:

```

```
16         //执行需求块 如: System.out.println("不给任何奖励 ");
17         //不要忘记break
18         break;
19     }
```

10.0.0.1. switch小知识 :

```
1 使用switch选择结构要注意以下几点:
2      1. 多个case后的常量不能重复
3      2. switch后的小括号中的表达式只允许为:
4          基本数据类型: char byte short int
5          引用数据类型: String enum(枚举, 后面会学)
6      3. switch语句的前后顺序可以进行颠倒.
7
8 使用场景:
9      必须是等值判断.
```

11. switch演示案例 :

```
1  package org.gyun.test;
2
3  import java.util.Collections;
4  import java.util.Scanner;
5
6  /**
7   * MyEclipse 10.7.1工具会报错
8   * 以下使用的工具为: Idea 2018.3
9   * @ClassName: Demo3.java
10  * @Description: switch的应用
11  * @Author: DongGaoYun
12  * @URL: www.gyun.org
13  * @Email: DongGaoYun@qq.com
14  * @QQ: 1050968899
15  * @Weixin: QingYunJiao
16  * @Date: 2019-9-5 下午1:56:14
17  * @Version: 1.0
18  */
19  public class Test1 {
20      /*
21       * 需求:
22       * 如果第一名, 参加麻省理工大学组织的1个月夏令营
23       * 如果第二名, 奖励惠普笔记本电脑一部
24       * 如果第三名, 奖励移动硬盘一个
25       * 否则, 不给任何奖励
26       * 分析需求: //变量 rank 名次
27       */
28      public static void main(String[] args) {
29          // ifMethod();
30          // 声明变量
31          // int rank=1;//写死, 代码没有灵活性了
32          // 用Scanner对象输入数据 三步走: 引包 创建对象 使用
33          Scanner input = new Scanner(System.in);
34          // 用变量接收输入的值 //简洁 好用 复用
35          System.out.print("请输入名次: ");
```

```

36 //int rank = input.nextInt();
37 // switch的使用
38 String rank = input.next();
39 /*
40     使用switch语句要注意以下几点:
41     1.多个case后的常量不能重复
42     2.switch后的小括号中的数据类型只允许为:
43     基本数据类型: char byte short int
44     引用数据类型: String enum(枚举,后面会学)
45     3.switch语句的前后顺序可以进行颠倒。
46 */
47 switch (rank) {
48     default:
49         System.out.println("不给任何奖励 ");
50         //不要忘记break
51         break;
52     case "a":
53         System.out.println("参加麻省理工大学组织的1个月夏令营");
54         //不要忘记break
55         break;
56     case "b":
57         System.out.println("奖励惠普笔记本电脑一部 ");
58         //不要忘记break
59         break;
60     case "c":
61         System.out.println("奖励移动硬盘一个 ");
62         //不要忘记break
63         break;
64 }
65 }
66
67 private static void ifMethod() {
68     // 声明变量
69     // int rank=1;//写死,代码没有灵活性了
70     // Scanner对象 三步走: 引包 创建对象 使用
71     Scanner input = new Scanner(System.in);
72     // 用变量接收输入的值 //简洁 好用 复用
73     System.out.print("请输入名次: ");
74     int rank = input.nextInt();
75     // 多重if
76     if (rank == 1) {
77         System.out.println("参加麻省理工大学组织的1个月夏令营");
78     } else if (rank == 2) {
79         System.out.println("奖励惠普笔记本电脑一部 ");
80     } else if (rank == 3) {
81         System.out.println("奖励移动硬盘一个 ");
82     } else {
83         System.out.println("不给任何奖励");
84     }
85 }
86 }

```

12. break跳出switch或循环小知识：

- 1 使用场景：
- 2 1. 在选择结构`switch`语句中使用
- 3 2. 在循环语句中使用
- 4 3. 其它场景使用没有作用
- 5
- 6 作用：
- 7 结束（终止）`switch`或者循环

13. default的作用：

- 1 和`if`中的`else`的功能一样，只有`switch`中的`case`全都不匹配时，就会选择默认的选项`default`
-

14. switch与多重if的区别

- 相同点：

都是用来处理多分支条件的结构

- 异同点：

switch选择结构

只能处理等值条件判断的情况

多重if选择结构

没有`switch`选择结构的限制，特别适合某个变量处于某个连续区间时的情况
