

04 - 分支结构：if..else 语句

C++ 程序设计基础

SOJ 信息学竞赛教练组

2024 年 5 月 16 日

目录

1 复习回顾

2 双分支结构

3 单分支与多分支

4 分支嵌套

5 总结

问题回顾：两位数判断

编程题

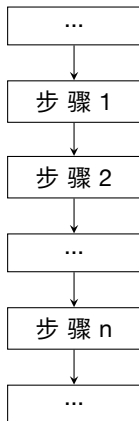
- 编写程序，由用户输入一个整数 n ($0 \leq n \leq 10^9$)，如果该整数是两位数，则输出 1，否则输出 0。
- 样例输入
10
- 样例输出
1

问题回顾：两位数判断

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     bool f = (10 <= n && n <= 99);
9     // 两位数是 10 ~ 99 之间的数
10    // 注意不能写成 10 <= n <= 99 的形式
11    cout << f << endl;
12
13    return 0;
14 }
```

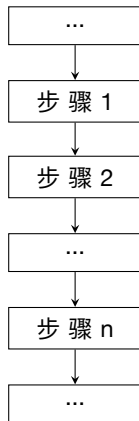
问题回顾：两位数判断

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     bool f = (10 <= n && n <= 99);
9     // 两位数是 10 ~ 99 之间的数
10    // 注意不能写成 10 <= n <= 99 的形式
11    cout << f << endl;
12
13    return 0;
14 }
```



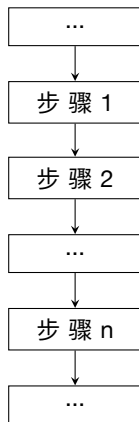
顺序结构

- 程序按照从上到下的顺序逐条执行各个语句



顺序结构

- 程序按照从上到下的顺序逐条执行各个语句
- 程序单一，无法处理复杂多样的问题
 - 成绩分级
 - 电梯运行（接送逻辑）



- 分支结构：可以根据不同情况，执行不同的语句

分支结构

- 分支结构：可以根据不同情况，执行不同的语句
- 主要的分支结构有
 - if..else 语句
 - switch 语句
 - 条件运算符

目录

1 复习回顾

2 双分支结构

3 单分支与多分支

4 分支嵌套

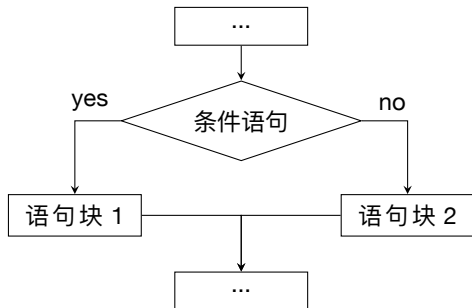
5 总结

双分支结构

- `if..else` 语句：以条件语句作为判断依据，选择执行不同的分支

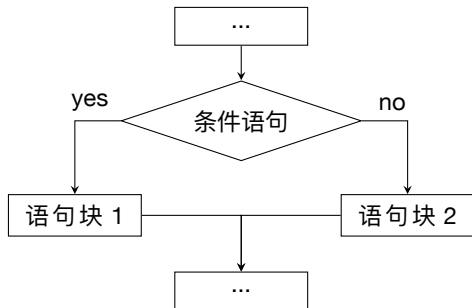
双分支结构

- if..else 语句：以条件语句作为判断依据，选择执行不同的分支



双分支结构

- if..else 语句：以条件语句作为判断依据，选择执行不同的分支



```
1 if (条件语句) {  
2   语句块 1;  
3 } else {  
4   语句块 2;  
5 }
```

if..else 语句

- if..else 语句：以条件语句作为判断依据，选择执行不同的分支
- 条件语句通常是布尔类型的表达式

```
1 if (条件语句) {  
2     语句块 1;  
3 } else {  
4     语句块 2;  
5 }
```

if..else 语句

- if..else 语句：以条件语句作为判断依据，选择执行不同的分支
- 条件语句通常是布尔类型的表达式
 - 条件成立（true）则执行语句块 1

```
1 if (条件语句) {  
2     语句块 1;  
3 } else {  
4     语句块 2;  
5 }
```

if..else 语句

- if..else 语句：以条件语句作为判断依据，选择执行不同的分支
- 条件语句通常是布尔类型的表达式
 - 条件成立（true）则执行语句块 1
 - 条件不成立（false）则执行语句块 2

```
1 if (条件语句) {  
2     语句块 1;  
3 } else {  
4     语句块 2;  
5 }
```


if..else 语句

- if..else 语句：以条件语句作为判断依据，选择执行不同的分支
- 条件语句通常是布尔类型的表达式
 - 条件成立 (true) 则执行语句块 1
 - 条件不成立 (false) 则执行语句块 2
- 当语句块中只有一个语句时，可以省略花括号

```
1 if (条件语句) {  
2     语句块 1;  
3 } else {  
4     语句块 2;  
5 }
```

if..else 语句

- if..else 语句：以条件语句作为判断依据，选择执行不同的分支
- 条件语句也可以是数值，但会被转化为布尔值

```
1 if (条件语句) {  
2     语句块 1;  
3 } else {  
4     语句块 2;  
5 }
```

if..else 语句

- if..else 语句：以条件语句作为判断依据，选择执行不同的分支
- 条件语句也可以是数值，但会被转化为布尔值
 - 数值非 0 值时，会转化成 true

```
1 if (条件语句) {  
2     语句块 1;  
3 } else {  
4     语句块 2;  
5 }
```

if..else 语句

- if..else 语句：以条件语句作为判断依据，选择执行不同的分支
- 条件语句也可以是数值，但会被转化为布尔值
 - 数值非 0 值时，会转化成 true
 - 数值为 0 时，会转化为 false

```
1 if (条件语句) {  
2     语句块 1;  
3 } else {  
4     语句块 2;  
5 }
```

例 4.1：4 的倍数

编程题

- 编写程序，由用户输入一个整数 n ($1 \leq n \leq 10^9$)，如果该整数是 4 的倍数，则输出 Yes，否则输出 No。
- 样例输入
12
- 样例输出
Yes

例 4.1: 4 的倍数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     if (n % 4 == 0) {
9         cout << "Yes" << endl;
10    } else {
11        cout << "No" << endl;
12    }
13
14    return 0;
15 }
```

例 4.2：闰年判断

编程题

- 闰年分为世纪闰年和普通闰年。
普通闰年的年份是 4 的倍数，且不是 100 的倍数；世纪闰年的年份是 400 的倍数。
编写程序，输入一个整数 $year$ ($1000 \leq year \leq 3000$)，表示一个年份，判断该年份是否为闰年，输出对应的判断结果。
- 样例输入
2010
- 样例输出
2010 is not leap year

例 4.2：闰年判断

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int y;
7     cin >> y;
8     bool a = (y % 4 == 0 && y % 100 != 0); // 普通闰年
9     bool b = (y % 400 == 0);             // 世纪闰年
10    if (a || b) {
11        cout << y << " is leap year" << endl;
12    } else {
13        cout << y << " is not leap year" << endl;
14    }
15
16    return 0;
17 }
```


例 4.2：闰年判断

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int y;
7     cin >> y;
8     // 闰年的判断条件:
9     // 年份是 4 的倍数且不是 100 的倍数 或 是 400 的倍数
10    if ((y % 4 == 0 && y % 100 != 0) || y % 400 == 0) {
11        cout << y << " is leap year" << endl;
12    } else {
13        cout << y << " is not leap year" << endl;
14    }
15
16    return 0;
17 }
```

条件运算符

- 条件运算符也是双分支结构的一种形式

条件运算符

- 条件运算符也是双分支结构的一种形式
 - 条件语句? 表达式 1: 表达式 2

条件运算符

- 条件运算符也是双分支结构的一种形式
 - 条件语句？ 表达式 1: 表达式 2
- 依照条件语句是否成立，取不同的结果

条件运算符

- 条件运算符也是双分支结构的一种形式
 - 条件语句？ 表达式 1: 表达式 2
- 依照条件语句是否成立，取不同的结果
 - 条件成立 (true) 则取表达式 1 的结果

条件运算符

- 条件运算符也是双分支结构的一种形式
 - 条件语句？ 表达式 1: 表达式 2
- 依照条件语句是否成立，取不同的结果
 - 条件成立 (true) 则取表达式 1 的结果
 - 条件不成立 (false) 则取表达式 2 的结果

条件运算符

- 条件运算符也是双分支结构的一种形式
 - 条件语句？ 表达式 1: 表达式 2
- 依照条件语句是否成立，取不同的结果
 - 条件成立 (true) 则取表达式 1 的结果
 - 条件不成立 (false) 则取表达式 2 的结果
 - ```
int a = 3, b = 5;
cout << (a < b? a: b) << endl; // 输出 3
```

# 条件运算符

- 条件运算符也是双分支结构的一种形式
  - 条件语句？ 表达式 1: 表达式 2
- 依照条件语句是否成立，取不同的结果
  - 条件成立 (true) 则取表达式 1 的结果
  - 条件不成立 (false) 则取表达式 2 的结果
  - ```
int a = 3, b = 5;  
cout << (a < b? a: b) << endl; // 输出 3
```
- 注意，应尽量保证两个表达式运算结果的数据类型是一致的

填空题

1. 阅读程序写结果

```
1 int main() {  
2     int a = 0, b = 1;  
3     if (a == b) {  
4         cout << "true" << endl;  
5     } else {  
6         cout << "false" << endl;  
7     }  
8  
9     return 0;  
10 }
```

输出:

填空题

1. 阅读程序写结果

```
1 int main() {  
2     int a = 0, b = 1;  
3     if (a == b) {  
4         cout << "true" << endl;  
5     } else {  
6         cout << "false" << endl;  
7     }  
8  
9     return 0;  
10 }
```

输出: false

填空题

2. 阅读程序写结果

```
1 int main() {  
2     int a = 0, b = 1;  
3     if (a = b) {  
4         cout << "true" << endl;  
5     } else {  
6         cout << "false" << endl;  
7     }  
8  
9     return 0;  
10 }
```

输出:

填空题

2. 阅读程序写结果

```
1 int main() {  
2     int a = 0, b = 1;  
3     if (a = b) {  
4         cout << "true" << endl;  
5     } else {  
6         cout << "false" << endl;  
7     }  
8  
9     return 0;  
10 }
```

输出: true

填空题

3. 阅读程序写结果

```
1 int main() {  
2     int a = 0, b = 1;  
3     if (a + b) {  
4         cout << "true" << endl;  
5     } else {  
6         cout << "false" << endl;  
7     }  
8  
9     return 0;  
10 }
```

输出:

填空题

3. 阅读程序写结果

```
1 int main() {  
2     int a = 0, b = 1;  
3     if (a + b) {  
4         cout << "true" << endl;  
5     } else {  
6         cout << "false" << endl;  
7     }  
8  
9     return 0;  
10 }
```

输出: true

目录

- 1 复习回顾
- 2 双分支结构
- 3 单分支与多分支**
- 4 分支嵌套
- 5 总结

单分支结构

- 单分支结构没有 else 部分

```
1 if (条件语句) {  
2     语句块;  
3 }
```


单分支结构

- 单分支结构没有 else 部分
 - 条件语句成立时，执行语句块

```
1 if (条件语句) {  
2     语句块;  
3 }
```

单分支结构

- 单分支结构没有 else 部分
 - 条件语句成立时，执行语句块
 - 条件语句不成立时，跳过语句块

```
1 if (条件语句) {  
2     语句块;  
3 }
```

填空题

1. 阅读程序写结果

```
1 int days = 365, y;  
2 cin >> y;  
3  
4 if ((y % 4 == 0 && y % 100 != 0) || y % 400 == 0) {  
5     days++;  
6 }  
7  
8 cout << days << endl;
```

输入：2020 输出：

输入：1900 输出：

填空题

1. 阅读程序写结果

```
1 int days = 365, y;  
2 cin >> y;  
3  
4 if ((y % 4 == 0 && y % 100 != 0) || y % 400 == 0) {  
5     days++;  
6 }  
7  
8 cout << days << endl;
```

输入：2020 输出：366

输入：1900 输出：

填空题

1. 阅读程序写结果

```
1 int days = 365, y;  
2 cin >> y;  
3  
4 if ((y % 4 == 0 && y % 100 != 0) || y % 400 == 0) {  
5     days++;  
6 }  
7  
8 cout << days << endl;
```

输入：2020 输出：366

输入：1900 输出：365

多分支结构

- 多分支结构会依次判断条件语句

```
1 if (条件语句 1) {  
2     语句块 1;  
3 } else if (条件语句 2) {  
4     语句块 2;  
5 } else if (条件语句 ...) {  
6     语句块 ...;  
7 } else {  
8     最后的语句块;  
9 }
```

多分支结构

- 多分支结构会依次判断条件语句
 - 当前条件成立时，执行对应的语句块，并结束判断

```
1 if (条件语句 1) {  
2     语句块 1;  
3 } else if (条件语句 2) {  
4     语句块 2;  
5 } else if (条件语句 ...) {  
6     语句块 ...;  
7 } else {  
8     最后的语句块;  
9 }
```

多分支结构

- 多分支结构会依次判断条件语句
 - 当前条件成立时，执行对应的语句块，并结束判断
- 因此至多只会执行一个语句块

```
1 if (条件语句 1) {  
2     语句块 1;  
3 } else if (条件语句 2) {  
4     语句块 2;  
5 } else if (条件语句 ...) {  
6     语句块 ...;  
7 } else {  
8     最后的语句块;  
9 }
```


多分支结构

- 多分支结构会依次判断条件语句
 - 当前条件成立时，执行对应的语句块，并结束判断
- 因此至多只会执行一个语句块
- 注意，最后的 else 不用写条件

```
1 if (条件语句 1) {  
2     语句块 1;  
3 } else if (条件语句 2) {  
4     语句块 2;  
5 } else if (条件语句 ...) {  
6     语句块 ...;  
7 } else {  
8     最后的语句块;  
9 }
```

例 4.3：成绩分级

编程题

- 编写程序，输入一个整数 $score$ ($0 \leq score \leq 100$)，表示成绩，输出对应的等级。

- 样例输入

90

- 样例输出

A

分数	等级
90 ~ 100	A
80 ~ 89	B
60 ~ 79	C
0 ~ 59	F

例 4.3：成绩分级

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int s;
7     cin >> s;
8     if (s >= 90 && s <= 100) {
9         cout << "A" << endl;
10    } else if (s >= 80 && s <= 89) {
11        cout << "B" << endl;
12    } else if (s >= 60 && s <= 79) {
13        cout << "C" << endl;
14    } else if (s >= 0 && s <= 59) {
15        cout << "F" << endl;
16    }
17
18    return 0;
19 }
```

例 4.3：成绩分级

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int s;
7     cin >> s;
8     if (s >= 90) {
9         cout << "A" << endl;
10    } else if (s >= 80) {
11        cout << "B" << endl;
12    } else if (s >= 60) {
13        cout << "C" << endl;
14    } else {
15        cout << "F" << endl;
16    }
17
18    return 0;
19 }
```

目录

- 1 复习回顾
- 2 双分支结构
- 3 单分支与多分支
- 4 分支嵌套**
- 5 总结

多分支与多个分支

- 多分支结构条件不能同时成立，最多只会执行一个分支

```
1 if (条件语句 1) {  
2     语句块 1;  
3 } else if (条件语句 2) {  
4     语句块 2;  
5 } else if (条件语句 ...) {  
6     语句块 ...;  
7 } else {  
8     最后的语句块;  
9 }
```

多分支与多个分支

- 多分支结构条件不能同时成立，最多只会执行一个分支
- 多个分支结构条件可能同时成立，符合条件就执行对应的分支，可执行多个分支

```
1 if (条件语句 1) {  
2   语句块 1;  
3 } else if (条件语句 2) {  
4   语句块 2;  
5 } else if (条件语句 ...) {  
6   语句块 ...;  
7 } else {  
8   最后的语句块;  
9 }
```

```
1 if (条件语句 1) {  
2   语句块 1;  
3 }  
4 if (条件语句 2) {  
5   语句块 2;  
6 }  
7 if (条件语句 3) {  
8   语句块 3;  
9 }
```

多分支与多个分支

- 当 *score* 为 95 时，分别会输出什么？

```
1 // 多分支
2 if (s >= 90) {
3     cout << "A" << endl;
4 } else if (s >= 80) {
5     cout << "B" << endl;
6 } else if (s >= 70) {
7     cout << "C" << endl;
8 }
```

```
1 // 多个分支
2 if (s >= 90) {
3     cout << "A" << endl;
4 }
5 if (s >= 80) {
6     cout << "B" << endl;
7 }
8 if (s >= 70) {
9     cout << "C" << endl;
10 }
```


分支嵌套

- 在分支结构的语句块中，使用分支结构，这就形成了分支嵌套

```
1  if (条件语句 1) {  
2    if (条件语句 2) {  
3      语句块 1;  
4    } else {  
5      语句块 2;  
6    }  
7  } else {  
8    if (条件语句 3) {  
9      语句块 3;  
10   } else {  
11     语句块 4;  
12   }  
13 }
```

分支嵌套

- 在分支结构的语句块中，使用分支结构，这就形成了分支嵌套
- 注意，应尽量不省略大括号，以明确各层次的嵌套关系

```
1  if (条件语句 1) {  
2    if (条件语句 2) {  
3      语句块 1;  
4    } else {  
5      语句块 2;  
6    }  
7  } else {  
8    if (条件语句 3) {  
9      语句块 3;  
10   } else {  
11     语句块 4;  
12   }  
13 }
```

例 4.4：数字判断

编程题

- 输入一个整数 n ，按以下规则输出对应描述 n 的语句。
 - 当 n 是正数时，输出 n 是奇数 odd 还是偶数 even
 - 当 n 是 0 时，输出 zero
 - 当 n 是负数时，输出 negative
- 样例输入
5
- 样例输出
odd

例 4.4：数字判断

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     if (n > 0) {
9         if (n % 2 == 0) {
10             cout << "even" << endl;
11         } else {
12             cout << "odd" << endl;
13         }
14     } else if (n == 0) {
15         cout << "zero" << endl;
16     } else {
17         cout << "negative" << endl;
18     }
19
20     return 0;
```

例 4.5：求三个数的最大值

编程题

- 编写程序，由用户输入三个整数 a, b, c ($0 \leq a, b, c \leq 1000$)，输出这三个数中的最大值。
- 样例输入
4 29 16
- 样例输出
29

例 4.5：求三个数的最大值

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int a, b, c;
7     cin >> a >> b >> c;
8     if (a >= b && a >= c) {           // a 最大的情况
9         cout << a << endl;
10    } else if (b >= a && b >= c) {      // b 最大的情况
11        cout << b << endl;
12    } else {                           // c 最大的情况
13        cout << c << endl;
14    }
15
16    return 0;
17 }
```

目录

- 1 复习回顾
- 2 双分支结构
- 3 单分支与多分支
- 4 分支嵌套
- 5 总结**

总结

- 双分支结构
- 单分支结构
- 多分支结构
- 分支嵌套

Thank you!