

C++ 程序设计 I

徐东

xu.dong.sh@outlook.com

信息与计算科学

数学系

上海师范大学

2018 年 8 月 23 日

内容

- 1 if 语句
- 2 switch 语句
- 3 条件表达式

双向选择语句

- *if - else* 语句

```
if(条件){
```

在条件**成立**的情况下，需要执行的（若干）语句；

```
}
```

```
else{
```

在条件**不成立**的情况下，需要执行的（若干）语句；

```
}
```

双向选择语句

- *if - else* 语句

```
if(条件){
```

在条件**成立**的情况下，需要执行的（若干）语句；

```
}
```

```
else{
```

在条件**不成立**的情况下，需要执行的（若干）语句；

```
}
```

- 二选一

双向选择语句

- *if - else* 语句

```
if(条件){
```

在条件**成立**的情况下，需要执行的（若干）语句；

```
}
```

```
else{
```

在条件**不成立**的情况下，需要执行的（若干）语句；

```
}
```

- 二选一
- 关键字 *else* 后不跟条件

双向选择语句

- *else* 语句块不能单独出现
- 悬挂 *else* 问题
 - 通过大括号 `{}` 明确 *if* 和 *else* 之间的配对关系
 - 默认, *else* 语句块和离它最近的单向 *if* 语句配对形成 *if-else* 语句。

多向选择语句

- *if - else if* 语句

```
if(条件 1){
```

在条件 1 成立的情况下，需要执行的（若干）语句；

```
}
```

```
else if(条件 2){
```

在条件 2 成立的情况下，需要执行的（若干）语句；

```
}
```

```
.....//可能存在的其他 else if 语句块
```

```
else{
```

在所有条件都不成立的情况下，需要执行的（若干）语句；

```
}
```

多向选择语句

- *if - else if* 语句

```
if(条件 1){
```

在条件 1 成立的情况下，需要执行的（若干）语句；

```
}
```

```
else if(条件 2){
```

在条件 2 成立的情况下，需要执行的（若干）语句；

```
}
```

```
.....//可能存在的其他 else if 语句块
```

```
else{
```

在所有条件都不成立的情况下，需要执行的（若干）语句；

```
}
```


多向选择语句

- *if - else if* 语句
 - 多选一
 - 可以放置多个 *else if* 语句块
 - *else if* 关键字后必须跟条件
 - 可以省略 *else* 语句块
 - *else* 语句块后没有条件
- 注意
 - 条件出现的次序对结果的影响
 - 条件判断不应该留有“缝隙”

选择语句

- *if* 语句 唯一的选择
- *if - else* 语句 二选一
- *if - else if* 语句 多选一

选择语句

- *if* 语句 唯一的选择
- *if - else* 语句 二选一
- *if - else if* 语句 多选一
- 问题

选择语句

- *if* 语句 唯一的选择
- *if - else* 语句 二选一
- *if - else if* 语句 多选一
- 问题
 - 成绩百分制的转换问题
 - 折扣问题
 - ...

选择语句

- *if* 语句 唯一的选择
- *if - else* 语句 二选一
- *if - else if* 语句 多选一
- 问题
 - 成绩百分制的转换问题
 - 折扣问题
 - ...
- 注意：条件出现的顺序和“缝隙”

switch 语句

```
1    double a = 0.0, b = 0.0;
2    char op;
3    cin >> a >> op >> b;
4    double result = 0.0;
5
6    if(op == '+'){
7        result = a+b;
8    }
9    else if(op == '-'){
10        result = a-b;
11    }
12    else if(op == '*'){
13        result = a*b;
14    }
15    else if(op == '/'){
16        result = a/b;
17    }
18    else{
19        ;
20    }
21
```

switch 语句

● 简单的字符比较

```
if(op == '+'){  
    result = a+b;  
}  
else if(op == '-'){  
    result = a-b;  
}  
else if(op == '*'){  
    result = a*b;  
}  
else if(op == '/'){  
    result = a/b;
```

switch 语句

```
if(op == '+'){  
    result = a+b;  
}  
else if(op == '-'){  
    result = a-b;  
}  
else if(op == '*'){  
    result = a*b;  
}  
else if(op == '/'){  
    result = a/b;
```

- 简单的字符比较

- 比较字符的 ASCII 码值

switch 语句

```
if(op == '+'){  
    result = a+b;  
}  
else if(op == '-'){  
    result = a-b;  
}  
else if(op == '*'){  
    result = a*b;  
}  
else if(op == '/'){  
    result = a/b;
```

- 简单的字符比较
 - 比较字符的 ASCII 码值
- 更简单的形式
 - *switch* 语句

switch 语句形式

```
switch(表达式)
{
    case 常量1 : 语句段1; [break;]
    case 常量2 : 语句段2; [break;]
    .....
    case 常量n : 语句段n; [break;]

    default    : 语句段x; [break;]
}
```

说明

- `switch(表达式)`
 - `switch` 表达式的值只能是字符或整数值
- `case` 常量
 - `case` 常量必须是字符常量或整数常量
 - `case` 常量只是代码执行的一个可能起始点而不是条件比较
- `break` 语句
 - 提前结束整个 `switch` 语句的执行 (跳过 `switch` 语句块中后续剩余部分的代码)
- `default` 类似于 `else` 的行为

说明

- 每个 case 常量必须唯一
- default 标签可选
- 多个 case 常量可以指向同一组语句
- case 和 default 标签可以按任意顺序排列

switch 语句的执行流程

- ① 计算 switch 表达式的值
- ② 将计算结果依次与 case 常量比较
- ③ 如果发现相等的 case 常量，那么立即执行该常量冒号后的语句，直到 switch 语句结束或碰到 break 语句为止。
- ④ 如果不与任何 case 常量匹配，那么执行 default : 后的语句。

代码

```
1  double a = 0.0, b = 0.0;
2  char op;
3  cin >> a >> op >> b;
4  double result = 0.0;
5
6  switch(op){
7      case '+': result = a + b;
8              break;
9      case '-': result = a - b; break;
10     case '*': result = a * b; break;
11     case '/': result = a / b; break;
12     default : ;
13             break;
14 }
15 cout << a << op << b << "u=u" << result << endl;
```

双向赋值

```
int a = 0, b = 0, x = 0;  
cin >> a >> b;  
if(a > b){  
    x = a + b;  
}  
else{  
    x = a - b;  
}
```

双向赋值

```
int a = 0, b = 0, x = 0;  
cin >> a >> b;  
if(a > b){  
    x = a + b;  
}  
else{  
    x = a - b;  
}
```


双向赋值

```
int a = 0, b = 0, x = 0;  
cin >> a >> b;  
if(a > b){  
    x = a + b;  
}  
else{  
    x = a - b;  
}
```

- 更简形式：条件表达式

条件表达式

- 条件表达式

条件 ? 表达式 1 : 表达式 2

条件表达式

- 条件表达式

条件 ? 表达式 1 : 表达式 2

- 条件成立 执行表达式 1

条件表达式

- 条件表达式

条件 ? 表达式 1 : 表达式 2

- 条件成立 执行表达式 1
- 条件不成立 执行表达式 2

条件表达式

- 条件表达式

条件 ? 表达式 1 : 表达式 2

- 条件成立 执行表达式 1

- 条件不成立 执行表达式 2

- ? : 条件运算符（三元运算符）

双向赋值

```
1  if(a > b){  
2      x = a + b;  
3  }  
4  else{  
5      x = a - b;  
6  }
```

- 等价的条件表达式

双向赋值

```
1  if(a > b){  
2      x = a + b;  
3  }  
4  else{  
5      x = a - b;  
6  }
```

- 等价的条件表达式

$$x = (a > b) ? a + b : a - b ;$$

Q & A