Module03-04
C++ 语言基础：语句

语言基础 - 语句

- 语句 (Statements)
  - 表达式、声明作为语句
  - 复合语句
  - 选择语句
  - 循环语句
  - 控制语句
  - 注释

语句 - 表达式和声明作为语句

- 表达式和声明语句
  - 表达式和声明可以单独构成一个完整语句

语句 - 复合语句

- 复合语句
  - 也叫语句块，是在 {} 内，有 0 到多个语句组成

语句 - 选择语句

- if 语句
  - 语句的格式
    - `if (condition) statement`
    - `if (condition) statement else statement`
  - 示例:

```
if(n < 5 && 2 < n) {
    if (k > 8)
        cout << n + k << endl;
    else
        cout << n - k << endl;
} else {
    cout << "n >= 5 or n <= 2" << endl;
}
```

语句 - 选择语句

- if 语句（续）
  - 关于 condition
    - condition 表达式的值：任何非 0 值转换成 true，其它为 false
    - 等于 == 和赋值 = 的混淆

```
if (m = 7)   // 原本是想表达m == 7
```

语句 - 选择语句

- switch 语句
  - 语句的格式
    - `switch (condition) statement`
  - 示例：

```
enum Color { RED, GREEN, BLUE };

void func(const Color& c) {
    switch (c) {
    case RED: cout << "Color RED" << endl; break;
    case GREEN: cout << "Color GREEN" << endl; break;
    case BLUE: cout << "Color BLUE" << endl; break;
    default: cout << "Unknown Color: " << c << endl; break;
    }
}
```

语句 - 选择语句

- switch 语句（续）
  - 关于 condition
    - condition 只能是具有整型结果的表达式，整型：bool、 char、 int、 short、 long 以及各自的无符号形式、枚举常量
  - 关于 case 语句中的 break
    - 如果 case 语句中没有 break 语句（以及其它控制语句），则表示该 case 分支执行后继续下一个 case 语句，这种情况叫“直落”
    - 最后一个 case(或 default) 语句不需 break，但也可以加上 break

语句 - 循环语句

- while 语句
  - 语句的格式
    - `while(condition) statement`
  - 示例

```
int n = 12;
while (--n > 0)
    cout << n << endl;
```

语句 - 循环语句

- for 语句
  - 语句的格式
    - for(init; condition; iterate-expr) statement
  - 示例：

```
vector<int> vec;
.....
for (size_t i = 0; i < vec.size(); ++i)
    cout << vec[i] << ' ';
cout << endl;
```

  - for 语句中 init 部分
    - init 部分声明的（一个或多个）对象的作用域直到 for 语句结束

语句 - 循环语句

- do .. while 语句
  - 语句的格式
    - do statement while (condition);
  - 示例：

```
do {
     int a = m / n;
} while (n > 9);
```

  - do 语句的不安全性
    - 由于 do 语句在没有条件测试之前进入循环体执行一次，造成潜在的不安全，如示例中如果第一次 n 的值为 0 ？
    - 建议尽量不要使用 do..while 语句！

语句 - 循环语句

- 无限循环
  - `for(;;)`
  - `while(true)`

语句 - 控制语句

- break
  - break 只能出现在循环语句或 switch 语句体内
  - 用于结束循环或结束选择 (switch)
- continue
  - continue 只能出现在循环语句体内
  - continue 会忽略其后未执行的代码，继续下一轮迭代
- goto
  - 语句的格式：

```
goto label;
label: statement
```

语句 - 注释

- 通行注释
  - 通行注释的形式：
    - // comment text
- 块注释
  - 块注释的形式
    - /* comment text */
  - 注意避免嵌套的块注释

表达式 - Bjarne's Advices

- Bjarne's Advices
  - 避免使用 goto
  - 避免使用 do..while