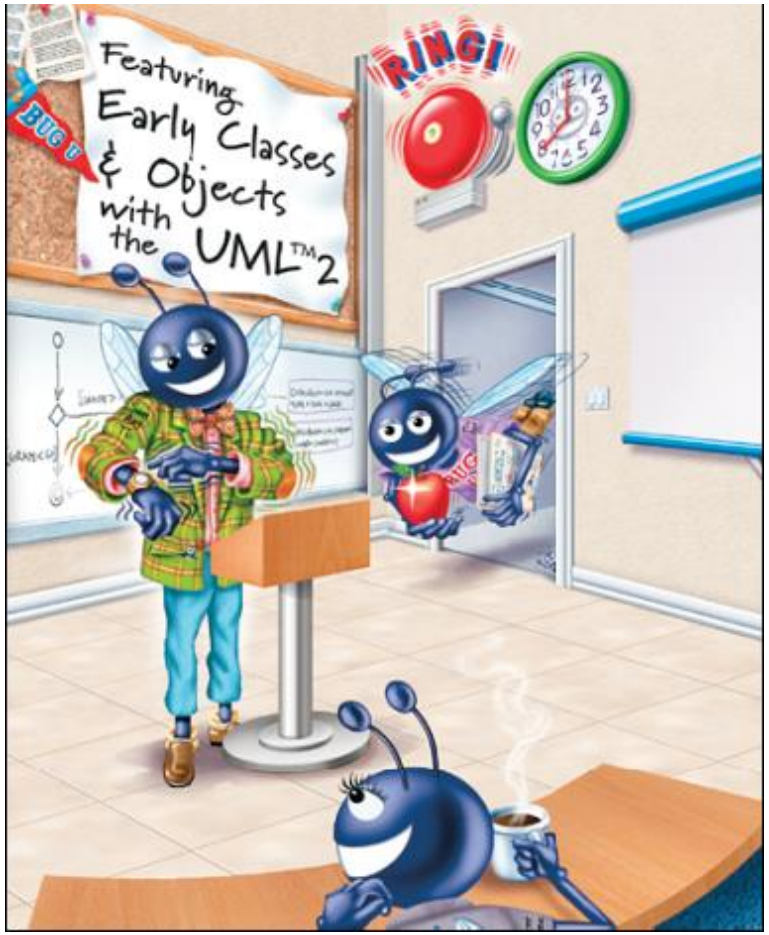
C++程序设计



上节课内容回顾

1. 如何定义类
2. 如何调用成员函数
3. 构造函数
4. 实现与接口分离

第四讲 控制语句

学习目标：

- 解决问题的基本方法
- if、if...else、while
- 计数器控制的循环和标记控制的循环
- 使用 for 循环语句和 do..while 循环
- 使用 switch 选择语句
- 使用 break 和 continue 改变控制流程

1. Control Structures

- **Bohm and Jacopini: all programs written in terms of 3 control structures**
  - **Sequence structure (顺序结构)**
  - **Selection structures (选择结构)**
    if, **if/else**, and **switch**
  - **Repetition structures (循环结构)**
    while, **do/while**, and **for**

1. Control Structures

- 单入口/单出口控制语句
  - 三种类型的控制语句：顺序、选择、循环
  - 两种组合方式
    - 堆叠
      - 前一个的出口作为下一个的入口
    - 嵌套

2. if Selection Statement

- if (grade >= 60)
  cout << "Passed";
- C++专门提供了 bool 数据类型，这种数据类型的取值不是 true 就是 false 。true 和 false 是C++的两个关键字。

3. if…else Double-Selection Statement

- **C++ code**
  - ```
    if ( grade >= 60 )
      cout << "Passed";
    else
      cout << "Failed";
    ```
    ```

3. if…else Double-Selection Statement

- Ternary conditional operator (?:) （三元算子）
  - 三个参数 (condition, value if true, value if false)
- 代码:
  - cout << (grade >= 60 ? “Passed” : “Failed”);

Condition　　Value if true　　Value if false

3. if…else Double-Selection Statement

- **Nested if…else statements**
  - ◈

```
if ( studentGrade >= 90 )
   cout << "A";
else if (studentGrade >= 80 )
   cout << "B";
else if (studentGrade >= 70 )
   cout << "C";
else if ( studentGrade >= 60 )
   cout << "D";
else
   cout << "F";
```

3. if…else Double-Selection Statement

性能提示：与罗列大量 if 单选结构相比，嵌套的 if/else 结构的速度会快许多。这是由于在后一种情况下，只要碰到满足条件的第一个表达式，整个结构便会终止并退出，不必遍历所有表达式。

3. if…else Double-Selection Statement

- **Dangling-else problem**
  - Example
    - ```
      if ( x > 5 )
        if ( y > 5 )
          cout << "x and y are > 5";
      else
        cout << "x is <= 5";
      ```
  - Compiler interprets as
    - ```
      if ( x > 5 )
        if ( y > 5 )
          cout << "x and y are > 5";
        else
          cout << "x is <= 5";
      ```

4. while Repetition Statement

- **Repetition statement**
  - **Example**
    - int product = 3;

```
while ( product <= 100 )
   product = 3 * product;
```

5. Counter-Controlled Repetition

- 直到计数器达到指定值时循环结束。
- 循环的次数已知

5. Counter-Controlled Repetition

- **Example**

十名学生进行一个测验，成绩已知，求平均成绩。

5. Counter-Controlled Repetition

- 没有初始化的变量
  - 包含”垃圾“或未定义的值
- 整数除法的截断误差
  - 商的小数部分被丢弃

5. Counter-Controlled Repetition

常见编程错误：假如不初始化counter或total，会造成不正确的程序结果。这属于逻辑错误。大多数变量最初都会包含垃圾值。

常见编程错误：如果在循环结束后的计算中使用计数器控制变量，经常会造成一种“相差1”的错误。

6. Sentinel-Controlled Repetition

- 如果问题变为：
  - 开发一个求平均成绩的程序来处理任意数量的成绩
  - 不知道学生的数量，程序如何知道何时结束？

6. Sentinel-Controlled Repetition

- Sentinel-controlled repetition（标记控制的循环）
  - 也称为”不确定“的循环
  - 使用标记值
    - 指示数据输入的结束
    - 当标记值输入时循环结束
    - 标记值的选择不能与正常输入相混淆

```
// GradeBook.cpp
#include <iostream>
……
using std::fixed; // ensures that decimal point is displayed

#include <iomanip> // para
using std::setprecision; //

// include definition of class Grade
#include "GradeBook.h“
……
```

fixed forces output to print in fixed point format (not scientific notation) and forces trailing zeros and decimal point to print

steprecision stream manipulator (in header **<iomanip>**) sets numeric output precision

```
void GradeBook::determineClassAverage()
{
   int total; // sum of grades entered by user
   int gradeCounter; // number of grades entered
   int grade; // grade value
   double average; // number with decimal point for average

   // initialization phase
   total = 0; // initialize total
   gradeCounter = 0; // initialize loop counter
```

```
cout << "Enter grade or -1 to quit: ";
cin >> grade; // input grade or sentinel value

// loop until sentinel value read from user
while ( grade != -1 ) // while grade is not -1
{
   total = total + grade; // add grade to total
   gradeCounter = gradeCounter + 1; // increment counter

   // prompt for input and read next grade from user
   cout << "Enter grade or -1 to quit: ";
   cin >> grade; // input grade or sentinel value
} // end while
```

```
   if ( gradeCounter != 0 ) // if user entered at least one grade...
   {
      // calculate average of all grades entered
      average = static_cast< double >( total ) / gradeCounter;

      // display total and average (with two digits of precision)
      cout << "\nTotal of all " << gradeCounter << " grades entered is "
         << total << endl;
      cout << "Class average is " << setprecision( 2 ) << fixed << average
         << endl;
   } // end if
   else // no grades were entered, so output appropriate message
      cout << "No grades were entered" << endl;
} // end function determineClassAverage
```

6. Sentinel-Controlled Repetition

良好编程习惯：每次利用键盘输入数据时都提醒用户。提醒时，应指出输入数据所采用的形式，以及任何特殊值。

常见编程错误：使用浮点数时，不可假定它们肯定能精确地表示，否则会导致不确切的结果。在大多数计算机上，浮点数都是约数。

6. Sentinel-Controlled Repetition

- Unary cast operator (一元类型转换运算符)
  - 创建一个暂时的不同数据类型的拷贝
    - Example: static_cast< double > (total)
      - 创建一个 total 的浮点类型的拷贝
  - 显示转换
- Promotion (提升)
  - 将一种数据类型 (如：int) 转换为另外一种数据类型 (如：double) 来执行计算
  - 隐式转换

6. Sentinel-Controlled Repetition

- 格式化浮点数
  - setprecision：参数化的流操作运算符
    - 声明显示的精度
    - 缺省精度为6位数字
  - fixed：非参数化的流操作运算符
    - 指示浮点数以定点小数方式显示
      - 与之对应的是科学计数法 (3.1×10^3)
  - showpoint：强制显示小数点

7. Increment and Decrement Operators

- If c = 5, then
  - cout << ++c;
    - c is changed to 6
    - Then prints out 6
  - cout << c++;
    - Prints out 5 (cout is executed before the increment)
    - c then becomes 6

7. Increment and Decrement Operators

- When variable is not in an expression
  - Preincrementing and postincrementing have same effect
    - Example
      - ++c;
        cout << c;

        and

        c++;
        cout << c;

        are the same

8. for 循环

```
for ( int counter = 1; counter <= 10; counter++ )
    cout << counter << " ";
```

8. for 循环

- General form of the for statement
  - for (*initialization*; *loopContinuationCondition*; *increment*)
    statement;
- Can usually be rewritten as:
  - *initialization*;
    while (*loopContinuationCondition*)

```
{
    statement;
    increment;
}
```

9. do…while 循环

```
.....
int counter = 1; // initialize counter

do
{
   cout << counter << " "; // display counter
   counter++; // increment counter
} while ( counter <= 10 ); // end do...while
.......
```

10. switch 多路选择

```
while ( ( grade = cin.get() ) != EOF )
{
   // determine which grade was entered
   switch ( grade ) // switch statement nested in while
   {
      case 'A': // grade was uppercase A
      case 'a': // or lowercase a
         aCount++; // increment aCount
      break; // necessary to exit switch
      ......
```

```
      case '\n': // ignore newlines,
      case '\t': // tabs,
      case ' ': // and spaces in input
        break; // exit switch

      default: // catch all other characters
        cout << "Incorrect letter grade entered."
          << " Enter a new grade." << endl;
        break; // optional; will exit switch anyway
    } // end switch
  } // end while
```

10. switch 多路选择

- **Reading character input**
  - **Function cin.get()**
    - **从键盘读入一个字符**
  - **EOF**
    - **<*ctrl*> *d* in UNIX/Linux**
    - **<*ctrl*> *z* in Windows**

10. switch 多路选择

The cctype Character Functions

Function Name	Return Value
`isalnum()`	This function returns `true` if the argument is alphanumeric (that is, a letter or a digit).
`isalpha()`	This function returns `true` if the argument is alphabetic.
`isblank()`	This function returns `true` if the argument is a space or a horizontal tab.
`iscntrl()`	This function returns `true` if the argument is a control character.
`isdigit()`	This function returns `true` if the argument is a decimal digit (0–9).
`isgraph()`	This function returns `true` if the argument is any printing character other than a space.
`islower()`	This function returns `true` if the argument is a lowercase letter.

10. switch 多路选择

The cctype Character Functions

`isprint()`	This function returns `true` if the argument is any printing character, including a space.
`ispunct()`	This function returns `true` if the argument is a punctuation character.
`isspace()`	This function returns `true` if the argument is a standard white-space character (that is, a space, formfeed, newline, carriage return, horizontal tab, vertical tab).
`isupper()`	This function returns `true` if the argument is an uppercase letter.

10. switch 多路选择

The cctype Character Functions

Function Name	Return Value
`isxdigit()`	This function returns `true` if the argument is a hexadecimal digit character (that is, 0–9, a–f, or A–F).
`tolower()`	If the argument is an uppercase character, `tolower()` returns the lowercase version of that character; otherwise, it returns the argument unaltered.
`toupper()`	If the argument is a lowercase character, `toupper()` returns the uppercase version of that character; otherwise, it returns the argument unaltered.

11. break and continue

```
while (cin.get(ch))
{
    statement1
    if (ch == '\n')
        break;
    statement2
}
statement3
```

break skips rest of loop and goes to following statement

11. break and continue

```
while (cin.get(ch))
{
    statement1
    if (ch == '\n')
        continue;
    statement2
}
statement3
```

continue skips rest of loop body and starts a new cycle

12. Structured Programming Summary

- Structured programming
  - 生产便于理解、测试、调试和维护的程序
- Rules for structured programming
  - 使用单入口/单出口的控制结构
  - Rules：stacking, nesting

思考题： Vintage Gum Vending Machine

用户插入硬币，在四个tab中选择一个，获取口香糖:

1. 每包口香糖 75 cents
2. 机器只接收 25 美分（quarter）
3. 不找零，一次只分发一包口香糖，多余的钱归机器
4. 如果用户选择的口香糖已经无货，可以提示，但不退钱

思考题： Vintage Gum Vending Machine

s - report the machine status

d - drop in a quarter

1 - pull the 1st tab

2 - pull the 2nd tab

3 - pull the 3rd tab

4 - pull the 4th tab

r - restock the machine

q - quit

思考题： Vintage Gum Vending Machine

示例:

```
> s
    1: 5 packs of Beemans
    2: 7 packs of Dentyne
    3: 1 packs of Chiclets
    4: 6 packs of Carefree
    There is $24.50 in the machine
```

思考题： Vintage Gum Vending Machine

示例:

```
> d
    ching
> d
    ching
> 3
   (nothing happens)
> d
    ching
> 3
    A pack of Chiclets slides into view.
```

思考题： Vintage Gum Vending Machine

示例:

```
> d
    ching
> d
    ching
> d
    ching
> 3
    You hear mechanical clanking, but no gum appears.
```

思考题： Vintage Gum Vending Machine

示例:

```
> 1
    (nothing happens)
 > s
     1: 5 packs of Beemans
     2: 7 packs of Dentyne
     3: 0 packs of Chiclets
     4: 6 packs of Carefree
     There is $26.00 in the machine
> r
    A grouchy-looking attendant shows up, opens the back, fiddles
around a bit, closes it, and leaves.
```

思考题： Vintage Gum Vending Machine

示例:

```
> s
    1: 10 packs of Beemans
    2: 10 packs of Dentyne
    3: 10 packs of Chiclets
    4: 10 packs of Carefree
    There is $0.00 in the machine
> q
   So long!
```