

第一章 初识 C++ 程序

目录

1 编写一个 C++ 程序

2 认识类

3 编译与调试程序

学习目标

- ① 掌握 C++ 程序的基本组成、了解类的概念；
- ② 学会独立上机编写、调试以及运行一个简单的 C++ 程序。

1.1 编写一个 C++ 程序

一个空的 main 函数

```
/*    一个空的main函数
    返回一个整型值
*/
int main(){    //程序从main函数开始执行
    return 0; /*返回一个整型值*/
}
```

1.1 编写一个 C++ 程序

一个空的 main 函数

```
/*    一个空的main函数
    返回一个整型值
*/
int main(){    //程序从main函数开始执行
    return 0; /*返回一个整型值*/
}
```

注释

- main() 是主函数，也是入口函数
- 函数包括四部分：返回值类型、函数名、形参列表和函数体
- int (整型类型)，即为 main 函数返回值类型
- C++ 有两种注释方法
 - 双斜线 (//) 注释单行语句，以换行符结束
 - 界定符 (/* */) 注释多行语句，以/* 开始，到 */结束

1.1 编写一个 C++ 程序

例 1.2

已知圆柱体的底面半径和高分别为 6cm 和 12cm，求圆柱体的体积？

1.1 编写一个 C++ 程序

例 1.2

已知圆柱体的底面半径和高分别为 6cm 和 12cm，求圆柱体的体积？

数学解法

解：设半径为 `radius`，高为 `height`，体积为 `volume`

由已知可得：`radius=6cm`，`height=12cm`

`volumn= π *radius2*height=3.14*6*6*12=1356.48cm3`。

1.1 编写一个 C++ 程序

代码清单 1.2, 例 1.2

```
#include <iostream>

int main() {
    // 定义三个double类型对象, 存放半径、高和体积的值
    double radius, height, volume;
    // 屏幕终端显示Please input radius and height:
    std::cout << "please_input_radius_and_height: ";
    // 从键盘输入6.5 12回车
    std::cin >> radius >> height;
    // 计算圆柱体体积, 并把结果存放到对象volume中
    volume = 3.14*radius*radius*height;
    // 屏幕输出 the volume is 1591.98
    std::cout << "the_volume_is_" << volume;
    return 0;
}
```


1.1 编写一个 C++ 程序

代码清单 1.2, 例 1.2

```
#include <iostream>

int main() {
    // 定义三个double类型对象, 存放半径、高和体积的值
    double radius, height, volume;
    // 屏幕终端显示Please input radius and height:
    std::cout << "please input radius and height: ";
    // 从键盘输入6.5 12回车
    std::cin >> radius >> height;
    // 计算圆柱体体积, 并把结果存放到对象volume中
    volume = 3.14*radius*radius*height;
    // 屏幕输出 the volume is 1591.98
    std::cout << "the volume is " << volume;
    return 0;
}
```

注释

- `iostream` 为输入输出流库, 通过 `cin` 和 `cout` 语句来实现读写操作
- “`std::`” 表明 `cin`、`cout` 定义在 `std` 的命名空间, “`::`” 为作用域运算符
- `radius`、`height` 和 `volume` 均为 `double` 类型的对象

1.2 认识类

类 (class) = 数据结构 (data structure) + 操作 (algorithm)

类 (class)

- 核心思想是定义一种数据结构 (data structure) 以及与数据结构相关联的一组操作，并把它们封装在一起，形成一个类类型(class type)。
- 属于用户自定义类型，具有抽象 (abstract) 和封装 (encapsulation) 的属性，是面向对象程序设计 (object-oriented programming, OOP) 的基础。

1.2 认识类

下面用面向对象的方法来求解前面的求圆柱体体积的问题

1.2 认识类

下面用面向对象的方法来求解前面的求圆柱体体积的问题

代码清单 1.3, 例 1.3

```
#include<iostream>
using namespace std; //使用标准命名空间
class Cylinder { //定义一个名为Cylinder的类类型
    double m_radius, m_height;
public:
    double volume() { //计算圆柱体的体积
        return 3.14*m_radius*m_radius*m_height;
    }
    Cylinder(double i=0, double h=0) :m_radius(i)
        , m_height(h){} //初始化半径和高的操作
};

int main() {
    //定义并初始化对象object
    Cylinder object(1.0, 1.0);
    //调用Cylinder类的volume函数
    double vol=object.volume();
    cout << vol << endl;
}
```

1.2 认识类

下面用面向对象的方法来求解前面的求圆柱体体积的问题

代码清单 1.3, 例 1.3

```
#include<iostream>
using namespace std; //使用标准命名空间
class Cylinder { //定义一个名为Cylinder的类类型
    double m_radius, m_height;
public:
    double volume() { //计算圆柱体的体积
        return 3.14*m_radius*m_radius*m_height;
    }
    Cylinder(double i=0, double h=0) :m_radius(i)
        , m_height(h){} //初始化半径和高的操作
};
int main() {
    //定义并初始化对象object
    Cylinder object(1.0, 1.0);
    //调用Cylinder类的volume函数
    double vol=object.volume();
    cout << vol << endl;
}
```

注释

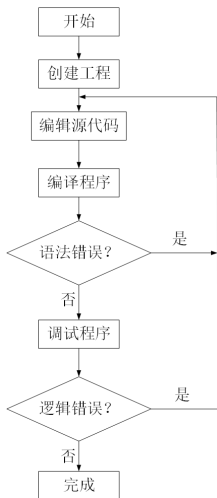
- Cylinder 为自定义的类类型, 其定义了一个含有半径和高的数据结构以及与之关联的操作。
- 此处 cout 没有 std:: 前缀是因为通过 using namespace std; 提前声明了使用标准命名空间。

1.3 编译与调试程序

C++ 程序编译、调试和执行步骤

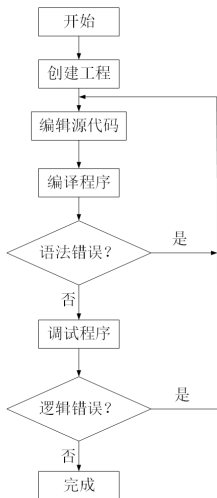
1.3 编译与调试程序

C++ 程序编译、调试和执行步骤



1.3 编译与调试程序

C++ 程序编译、调试和执行步骤



说明

- 1 C++ 程序工程的创建
推荐使用 **Visual Studio** 编译器
- 2 添加空的源文件 (*.cpp),
如 main.cpp
- 3 编写源代码
- 4 编译, 编译器会指出具体的语法错误
- 5 改正语法错误
- 6 调试程序 (找出逻辑错误)
- 7 运行程序

1.3 编译与调试程序

Visual Studio 几个常用快捷键

F5	执行程序
F7	编译源文件
F9	添加断点
F10	单步执行一行代码
Ctrl+F5	执行但不调试

1.3 编译与调试程序

Visual Studio 几个常用快捷键

F5	执行程序
F7	编译源文件
F9	添加断点
F10	单步执行一行代码
Ctrl+F5	执行但不调试

建议

- 遵循“编辑-编译-调试”的原则
- 养成调试程序的好习惯

本章结束