

第一讲 面向对象编程与 C++

面向对象编程概念

- 面向对象的核心概念是类和对象
- 类是对现实世界的抽象，包括表示静态属性的数据和对数据的操作，对象是类的实例化。对象间通过消息传递相互通信，来模拟现实世界中不同实体间的联系。在面向对象编程中，对象是组成程序的基本模块。

面向对象与面向过程

- 面向对象(Object Oriented)是把构成问题事务分解成各个对象，建立对象的目的不是为了完成一个步骤，而是为了描述某个事物在整个解决问题的步骤中的行为。
- 面向过程(Procedure Oriented)就是分析出解决问题所需要的步骤，然后用函数把这些步骤一步一步实现，使用的时候一个一个依次调用。

面向对象编程的目的

- 现实世界本身是由对象所组成
- 对象来描述问题比用功能来描述问题更自然、更完整、更准确
- 软件所要解决的各种现实问题本身就是由各种对象所组成，而且相对于功能的变化，对象更稳定

C++的特点

- 与 C 兼容
- 语法灵活，功能强大
- 面向对象语言
- 适合大型系统的开发
- 多样化的设计风格

命名空间

NameSpace（命名空间）是 C++中的一种机制，可以在不同的空间内使用相同名字类名或函数名，用来控制不同类库相同命名冲突的问题。

```
namespace 命名空间名称
{
    // 命名空间成员
}
```

标准命名空间 std:

使用方式

整体声明

```
#include <iostream>
using namespace std;
int main( )
{
    cout << "Hello C++!"<<endl;
    return 0;
}
```

单个对象临时声明

```
# include <iostream>
int main ( )
{
    std::cout<<"Hello C++!"<<std::endl;
    return 0;
}
```

单个对象使用前进行声明

```
# include <iostream>
using std::cout;
using std::endl;
int main()
{
    cout<<"Hello C++!"<<endl;
    return 0;
}
```

编译预处理

C++提供的预处理功能主要有宏定义命令、文件包含命令、条件编译命令3种，主要处理#开始的预编译指令，如宏定义(#define)、文件包含(#include)、条件编译(#ifdef)等。这些命令以符号“#”开头，而且末尾不包含分号。

宏定义

不带参数

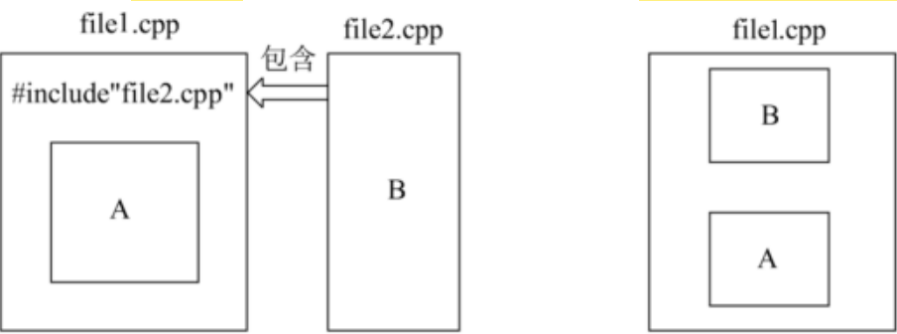
```
#include <iostream>
#define PI 3.1415926535879732384626
using namespace std;
int main(int argc, char const *argv[])
{
    cout << PI << endl;
    return 0;
}
// 结果: 3.14159
```

带参数

```
#include <iostream>
#define PI 3.1415926535879732384626
#define S(r) PI*r*r
using namespace std;
int main(int argc, char const *argv[])
{
    double r;
    cin >> r;
    cout << S(r) << endl;
    return 0;
}
// 结果: 3
//      28.2743
```

文件包含

文件包含include命令的作用是将一个文件的**全部内容包含到本文件中**，有如下两种形式：



- #include <文件名称>
- #include "[路径]文件名"

条件编译

一般情况下，源程序中所有的行都应参加正式编译。但有时候**希望在一定条件下编译某些行**，在其它情况不编译这些行。这时候就需要在预处理阶段作**条件编译**处理。

常见的条件编译有如下几种使用方法：

ifdef、else和endif #ifdef 标识符 程序段1 #else 程序段2 #endif	ifndef、else和endif #ifndef 标识符 程序段1 #else 程序段2 #endif	if、elif和else #if 常量表达式1 程序段1 #elif 常量表达式2 程序段2 #elif 常量表达式3 程序段3 #else 程序段n+1 #endif
--	--	---

一些说明

- C++程序中用到的特殊符号，如<、>、"、/、*、{、}等一律为**半角英文符号**，**不是全角中文符号**
- C++由于面向对象的概念，输入输出将不再使用 **scanf** 和 **printf**，而是使用输入输出流，其中 **cin** 是输入流，负责接受键盘的输入，**cout** 是输出流，负责输出到屏幕或外设等
- **setw(int n)** 用来控制**输出间隔**。setw(16) 表示其后面紧跟的输出占 16 个位置，不足的用空格填充。