移动平台应用软件开发 C/C++/JAVA基础 引入库的方法

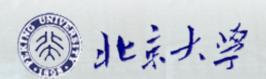
主讲: 秘齐勋

zhangqx@ss.pku.edu.cn

《移动平台应用软件开发》课程建设小组

北京大学

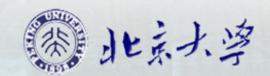
二零一五年



引入库的方法

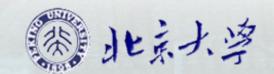
· C/C++通过与处理指令 include来包含 库文件

• JAVA通过import方式引入

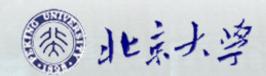


C++ 中的namespace

- 在C++中,名称(name)可以是符号常量、 变量、宏、函数、结构、枚举、类和对象 等等。
- 为了避免在大规模程序的设计中,以及在程序员使用各种各样的C++库时,这些标识符的命名发生冲突,标准C++引入了关键字namespace(命名空间/名字空间/名称空间/名域),可以更好地控制标识符的作用域。



- #include<iostream> // 这个是1998年标准化以后的标准头文件
- #include<iostream.h> // 这个是标准化以前的头文件
- 更本质上的区别就是iostream把标准C++ 库的组件放在一个名位std的namespace 里面。而相对的iostream.h则将这些标准 组件放在全局空间里



动手实践

· 使用<iostream>时,引入std::有以下方法:

```
using namespace std;
cout<<x;</pre>
```

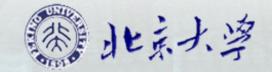
```
using std::cout;
cout<<x;
```

```
std::cout<<x;
```

```
#include <iostream.h>
int main()
{
    cout<<''hello\n'';
    return 0;
}</pre>
```

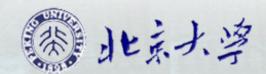
原写法

编译方法: #g++ hello.cpp -o hello



JAVA中的import

- import package1[.package2...].classname;
 - ▶package 为包名, classname 为类名
 - ▶import 只能导入包所包含的类,而不能导入包。
 - ➤一般不导入单独的类,而是导入包下所有的类 ,例如 import java.util.*;
 - ➤Java 编译器默认为所有的 Java 程序导入了 JDK 的 java.lang 包中所有的类(import java.lang.*;),其中定义了一些常用类,如 System、String、Object、Math 等,因此我们可以直接使用这些类而不必显式导入。但是使用其他类必须先导入。



实验

· 分别写一个C语言/C++语言/JAVA语言 的Hello World程序



Q&A

本讲结束!

