

开始学习 C++

唐浩

信息与计算科学 3200102118

2022 年 7 月 7 日

目录

进入 C++	1
C++ 语句	1
1 进入 C++	1
1.1 main() 函数	1
1.2 C++ 注释	2
1.3 C++ 预处理器和 iostream 文件	2
1.4 头文件名	2
1.5 名称空间	3
1.6 使用 cout 进行 C++ 输出	3
1.7 C++ 源代码格式化	3
2 C++ 语句	3
2.1 声明语句和变量	4

1 进入 C++

先展示一个简要的 C++ 程序如下：

```
// myfirst.cpp -- displays a message

#include<iostream>    //a preprocessor directive
using namespace std; //make definitions visible

int main(int argc, char *argv[]) //function header
{
```

```
    cout << "Come up and C++ me some time." << endl; //message
    cout << "You won't regret it!" << endl;
    return 0; //terminate main()
}
```

本人使用的是 **linux** 系统 (**ubuntu 20.04**), 采用的是 **g++** 编译器, 故输入命令: '**g++ -o test myfirst.cpp**', 编译器运行之后将生成一个可执行文件 '**test**', 继续在终端 (emacs shell) 输入 '**./test**', 即可得到输出数据:

```
Come up and C++ me some time.
You won't regret it!
```

1.1 main() 函数

每个 C++ 程序都必须有一个 **main()** 函数, 其基本结构为:

```
int main(int argc, char*argv[])
{
    statements
    return 0;
}
```

在运行 C++ 程序时, 通常都是从 **main()** 函数开始执行, 若无 **main()**, 则程序不完整, 编译器将报错。

1.2 C++ 注释

C++ 的注释以双斜杠 (**//**) 打头。一个优秀的程序是不能没有注释的, 注释将给读者带来一个良好的说明体验。注释是对读者的注释, 编译器不会识别注释。

C++ 生成的注释从 **//** 开始, 到行尾结束, 可以单独位于一行, 也可以和代码位于同一行。

1.3 C++ 预处理器和 iostream 文件

若程序要使用 C++ 输入或输出工具, 则需要提供这样的两行代码:

```
#include<iostream>
using namespace std; //如果不使用, 则在cout, endl等前需要加上 std::
```

#include<iostream> 将导致[预处理器](#)将 **iostream** 文件的内容添加到程序中。这是一种典型的预处理操作: 在源代码被编译之前, 替换或添加文本。**iostream** 中的 **io** 指的是输入和输出。C++ 的输入/输出方案涉及 **iostream** 文件中的多

个定义。为了使用 `cout` 来显示消息，程序需要这些定义。`#include` 编译指令会使 `iostream` 文件的内容随源代码文件的内容一起发给编译器。

注意：使用 `cin` 和 `cout` 进行输入和输出的程序必须包含文件 `iostream`。

1.4 头文件名

像 `iostream` 这样的文件叫做包含文件 (include file) —— 由于它们被包含在其他文件中；也叫头文件 (header file) —— 由于它们被包含在文件起始处。C++ 编译器自带了很多头文件，每个头文件都支持一组特定的工具。C++ 依旧可以使用 C 的头文件类型: `xxx.h`，但是 C++ 的头文件其实是没有扩展名 `h` 的。例如，C++ 版本的 `math.h` 为 `cmath` (有些 C 头文件被转换成 C++ 头文件，去掉了 `.h`，加上了前缀 `c`)。

1.5 名称空间

```
using namespace xxx;
```

如果存在两个不同的函数，其名字的都一样的，可以通过将其分别装入不同的名称空间，使编译器能准确了解用户所需的是哪个函数

1.6 使用 `cout` 进行 C++ 输出

```
cout << "string" << endl;
```

`cout` 将双引号中的内容打印出来，由于其不会自动移到下一行，故行末加上 `endl`，`endl` 是一个特殊的 C++ 符号，表示一个重要的概念：重启一行。

C++ 还提供另外一种在输出中指示换行的旧式方法：C 语言符号 `\n`：

```
cout << "What's next?\n";
```

1.7 C++ 源代码格式化

在 C++ 中，分号标示了语句的结尾。因此在 C++ 中，回车的作用就和空格或制表符相同。也就是说，在 C++ 中，通常可以在能够使用回车的地方使用空格，反之亦然。

2 C++ 语句

```
//carrots.cpp
```

```
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
```

```
{
    int carrots;

    carrots = 25;
    cout << ``I have ";
    cout << carrots;
    cout << `` carrots." << endl;
    carrots = carrots - 1;
    cout << ``Crunch, crunch. Now I have " << carrots << `` carrots." << endl;
    return 0;
}
```

2.1 声明语句和变量

在 C++ 中，是使用声明语句来指出存储类型并提供位置标签的。例如: `int carrots;`

它提供了两项信息：需要的内存以及该内存单元的名称。`int` 表示整数。变量必须要声明。