

1.7 编译预处理指令

- 编译预处理是指在对源程序正式进行**编译之前**，**编译预处理程序**（precompiler）根据源程序中的编译预处理指令**对源程序进行预先的处理**。
- 所有的编译预处理指令都以符号“#”开始。编译预处理指令扩展了C++编程环境的范围，改善了程序的组织和管理。
- 编译预处理指令主要包括：
 - #include文件包含指令
 - #define宏定义指令
 - 条件编译指令

#define宏定义指令

宏定义就是定义一个代表一个字符串的标识符，该标识符称为宏（macro）。

```
1  #define<标识符> [<字符串>]
2  #define<标识符> (<参数表>) (<字符串>)
```

编译时，precompiler将源程序中的宏用被定义的字符串替换，此替代过程称为宏替换或宏展开。

- 取消宏定义：

```
1  #undef<标识符>
```

- 建议：不用**
- 符号常量的宏可用const取代
- 带参数的宏可用内联函数取代

```
#define PI 3.141592654
#define MAXLENGTH 100
```

```
#define MAX(a, b) ((a)>(b) ? (a):(b))

void eg2_37() {
    cout << MAX(10, 11) << endl;
}
```

#include文件包含指令

- #include文件包含指令指示编译器将一个源文件嵌入到当前源文件中该指令处。#include包含指令有两种使用形式：

- `#include<文件名> //用于系统文件`
 - precompiler在编译器自带的或外部库的头文件中搜索
 - 要嵌入的文件，它们一般是系统提供的公共头文件，存放在系统目录中的Include子目录下。
 - `#include "文件名" //用于用户自定义文件`
- precompiler先在当前子目录搜索要嵌入的文件，如果没有找到文件，再去搜索编译器自带的或外部库的头文件。
 - 系统定义的用<文件名>自己定义的用"文件名" 。

条件预编译指令

条件编译预处理是指按照指定的条件决定 **哪些源程序段将被编译** ,哪些源程序段将不被编译.条件预编译的三种形式:

```
1  #if...[#else]...endif
2  #ifdef...[#else]..#endif
3  #ifndef...[#else]...#endif
```

主要作用：**管理测试代码与生产代码**

例子：条件预编译

```
1  #include<bits/stdc++.h>
2  using namespace std;
3
4
5  #ifdef LETTER
6      #define LETTER 0
7  #endif
8  void eg_38_1()
9  {
10     char c1, a_A = 'a' - 'A';
11     const char* str = "LANGUAGE";
12     int i = 0;
13
14     while ((c1=*(str+i)) != '\0')
15     {
16         i++;
17     #if LETTER
18         if (c1 >= 'a' && c1 <= 'z') c1 -= a_A;
19     #else
20         if (c1 >= 'A' && c1 <= 'Z') c1 += a_A;
21     #endif
22         cout << c1;
23     }
24     cout << endl;
25 }
26
27 int main()
28 {
29     eg_38_1();
30     return 0;
31 }
```

工程参考

```
1  #ifdef DEBUG1
2  void eg2_39()
3  {
4      int vDebug = 50, vRelease = 6;
5  #ifdef DEBUG
6      cout << "Debug Version: " << vDebug << endl;
7  #else
8      cout << "Release Version: " << vRelease << endl;
9  #endif
10 }
```

管理测试代码与生产代码:

- 程序调试好后,只需删除宏定义即可得到正式版
- 减少编译后程序的大小

例子: 条件预编译

```
1  // eg.hpp
2  #ifndef eg_hpp //若未定义宏eg_hpp, 下面的代码参与编译
3  #define eg_hpp //定义宏eg_hpp
4
5  void eg2_31();
6  void eg2_32();
7  void eg2_33();
8  void eg2_34();
9
10
11 #endif /* eg_hpp */
```

防止头文件被重复包含。也可将include换为import