

20190108

C++语法规则很多，要落实下来，得通过多敲代码来理解*看N遍不如写一次；在写代码的过程中，会碰到其它你不曾碰到过的编译问题,切记程序是调试出来的；再就是通过练习，把敲代码的速度提升上来，熟悉键盘，培养写代码的感觉；熟练掌握单例模式的用法，做到随手写出来

☑1、编写一个类，实现简单的栈。栈中有以下操作：

元素入栈 void push(int); 元素出栈 void pop(); 读出栈顶元素 int top(); 判断栈空 bool empty(); 判断栈满 bool full(); 如果栈溢出，程序终止。栈的数据成员由存放 10个整型数据的数组构成。先后做如下操作：

创建栈

将10入栈

将12入栈

将14入栈

读出并输出栈顶元素

出栈

读出并输出栈顶元素

结果

```
kyle@ubuntu:Stack$g++ *.cc  
kyle@ubuntu:Stack$./a.out
```

创建一个空栈

把10入栈

10已入栈

把12入栈

12已入栈

• 把14入栈

14已入栈

栈顶元素为:14

出栈一次

栈顶元素为:12

出栈一次

栈顶元素为:10

出栈一次

调用了析构函数

- 实现代码位于Stack文件夹

疑问：

- Stack.cc 和 TestStack.cc文件在调用Stack头文件时候插件报错，说not found Stack.h这个头文件，但是g++编译是可以通过的，不知道问什么

```
1Stack.cc 2TestStack.cc 3Stack.h >
1  ///
2  /// @file    TestStack.cc
3  /// @author   kyle(kyleyang58@gmail.com)
4  /// @date    2019-01-12 18:26:33
5  ///
>> 6 #include "Stack.h"
7 #include <iostream>
8 using std::cout;
9 using std::endl;
10
11 int main(void)
12 {
13     cout << "创建一个空栈"<< endl;
14     Stack stack ;
15     stack.push(10);
16     stack.push(12);
17     stack.push(14);
18     while (!stack.empty())
19     {
20         cout<< "栈顶元素为:" << stack.top() <<endl;
21         stack.pop();
22     }
23     stack.pop();
24     while (!stack.empty())
```

- 已解决：原因是在.h文件初始化的过程中与private：中元素声明的顺序不一致导致的

2、编写一个类，实现简单的队列。队列中有以下操作：

元素入队 void push(int);

元素出队 void pop();

读取队头元素 int front();

读取队尾元素 int back();

判断队列是否为空 bool emty();

判断队列是否已满 bool full();

```

kyle@ubuntu:Queue$g++ *.cc
kyle@ubuntu:Queue$./a.out
Build a Queue
Queue is empty.
Queue is not full
push 1 in Queue
Queue is not full
push 2 in Queue
push 3 in Queue
push 4 in Queue
push 5 in Queue
push 6 in Queue
push 7 in Queue
push 8 in Queue
push 9 in Queue
Queue is full, can not push anymore.
Queue is full, can not push anymore.
_rear : 9
rear element is:9
front element is:1
rear element is:9
push 10 in Queue
_rear : 0
rear element is:10
pop Queue
2 3 4 5 6 7 8 9 10
Queue is empty.
Queue is not full
>> ~Queue()

```

实现代码位于Queue文件夹

3、用C++实现一个双向链表

```

kyle@ubuntu:Linklist$ls
Doublelink_list.cc test.cc
kyle@ubuntu:Linklist$g++ Doublelink_list.cc
kyle@ubuntu:Linklist$./a.out
连续插入6个数字：
此时链表为： 0 1 2 3 4 5
删除头元素：
此时链表为： 1 2 3 4 5
删除尾元素：
此时链表为： 1 2 3 4
删除3
此时链表为： 1 2 4
kyle@ubuntu:Linklist$

```

