

极客班 C++ 系统工程师 STL 与泛型编程作业题

说明：

1. 答案提交截止日期为2015年10月11日 23:30。
2. 请将答案提交到云课堂对应测验与作业单元。
 - 请以自己昵称与注册邮箱对应的文件夹里创建文件夹“STL与泛型编程”
 - 请在作业文档中注明题目和对应的代码文件夹（部分题目所需的关键代码可以写在作业文档里），方便老师批改。

STL与泛型编程

1. 给定一个 vector : `v1 = [0, 0, 30, 20, 0, 0, 0, 0, 10, 0]`，希望通过 `not_equal_to` 算法找到不为零的元素，并复制到另一个 vector: `v2`
2. 为以下 Programmer 对象提供一个基于 Id 并且升序的仿函数 `ProgrammerIdGreater`，使得 Programmer 对象可以在 set 中以 Id 排序存放
 - (1) 将 `Programmer(1, L"Scott Meyers")`、`Programmer(2, L"Martin Fowler")`、`Programmer(3, L"Bill Gates")`、`Programmer(4, L"P.J. Plaught")`、`Programmer(5, L"Stanley B. Lippman")`、`Programmer(6, L"Andrei Alexandrescu")` 插入到一个 set 中；
 - (2) 通过 `for_each` 遍历 set，并且使用 Programmer 对象的 `Print` 方法打印出对象，结果应该如下所示；

```
[6] : Andrei Alexandrescu
[5] : Stanley B. Lippman
[4] : P.J. Plaught
[3] : Bill Gates
[2] : Martin Fowler
[1] : Scott Meyers
```

- (3) 在 set 中查找 Id 为 3、名称为 “Bill Gates” 的对象；
- (4) 如果(2)中找到目标对象，则将其名称改成 “David Vandevoorde” ，
再通过 for_each 重新打印 set，查看是否真的修改成功了；
- (5) 重新定义一个 set(命名为 set2)，将 Programmer 在 set2 中排序方式改为通过 Name 来排序，为此还需重新定义一个仿函数用于比较 Name，
请写出该仿函数，名字可能类似 `ProgrammerNameComparer`；
- (6) 通过 for_each 重打印 set2，验证 set2 中的元素是否是按照名字来排序的。

```
struct Programmer
{
    Programmer(const int id, const std::wstring name)
        : Id(id), Name(name)
    {
    }

    void Print() const
    {
        std::wcout << L "[" << Id << L "] : " << Name << std::endl;
    }

    int Id;
    std::wstring Name;
};

struct ProgrammerIdGreater { ... }
```