

第九周作业

请自行完成教材上面第8章的课后练习。

一、简答题

1. 什么是泛型(Generics)程序设计？泛型程序设计为我们提供了什么方便？在定义和使用模板的时候有什么问题？一般我们怎么解决这些问题？（0.5分）
2. STL主要包含哪几部分？根据自己的理解，请对各部分将其简要分类并给出典型的代表。如算法模板：排序sort、查找find、求和accumulate等（还有很多类..）（1分）

二、选择填空题

1. 下列类模板定义正确的是____。(0.5分)

A)

```
1 | template <class T1, class T2>
2 | class A {
3 |     T1 b;
4 |     int fun(int a){ return T1 + T2;}
5 | };
```

B)

```
1 | template <class T1, class T2>
2 | class A {
3 |     int T2;
4 |     T1 fun(T2 a){ return a + T2;}
5 | };
```

C)

```
1 | template <class T1, class T2>
2 | class A {
```

```

3 | public:
4 |     T2 b, T1 a;
5 |     A<T1>(){}
6 |     T1 fun(){ return a;}
7 | };

```

D)

```

1 | template <class T1, class T2>
2 | class A {
3 |     T2 b;
4 |     T1 fun(double a){
5 |         b = (T2)a;
6 |         return (T1)a;
7 |     }
8 | }

```

2. 下面程序的输出结果是:

```

A::Print:1
B::Print:2
B::Print:3

```

填空使程序完整。(1分)

```

1 | template <class T>
2 | void PrintAll(const T & c){
3 |     typename T::const_iterator i;
4 |     for (i = c.begin(); i != c.end(); ++i) {
5 |         _____(1)_____
6 |     }
7 | }
8 | class A {
9 | protected:
10 |     int nVal;
11 | public:
12 |     A(int i):nVal(i){}
13 |     virtual void Print(){
14 |         cout << "A::Print:" << nVal <<endl;
15 |     }
16 | };

```

```

17 class B: public A{
18 public:
19     B(int i):A(i){}
20     void Print(){
21         cout << "B::Print:" << nVal << endl;
22     }
23 };
24 int main() {
25     _____(2)_____;
26     v.push_back(new A(1));
27     v.push_back(new B(2));
28     v.push_back(new B(3));
29     PrintAll(v);
30     return 0;
31 }

```

3. 下面程序的输出结果是:

John,Mary,Jack,Tom,

填空使程序完整。(1分)

```

1  #include <list>
2  #include <iostream>
3  #include <string>
4  using namespace std;
5  template <class T>
6  class MyClass {
7      list<T> array;
8      public:
9          MyClass(T * begin,int n):array(n){
10              copy(begin, begin + n, array.begin());
11          }
12          void List() {
13              _____(1)_____;
14              for (i = array.rbegin();i != array.rend(); ++i) {
15                  cout << *i << ",";
16              }
17          }
18 };
19 int main(){
20     string array[4] = {"Tom","Jack","Mary","John"};

```

```

21 | -----(2)-----;
22 | obj.List();
23 | return 0;
24 | }

```

三、编程题

1. 自行实现模板函数 `accumulate2()`，要求具有和 `accumulate()` 同样的功能，功能参考课本附录E。（2分）

- 实现要求：

- 实现 `accumulate2(InIt first, InIt last, T val)`；
- 实现 `accumulate2(InIt first, InIt last, T val, BinOp op)`；

- 提交要求：

- 提交实现的两个函数模板，命名为 `accumulate2.cpp`。

备注：

`InIt` 为输入迭代器，具体可参考ppt和教材。

2. 实现习题6.8中第17题集合类 `IntSet` 的功能，其元素类型不再限制为 `int` 型，而是支持各种类型的类模板，类名改为 `Set`，相应的 `IntSet` 初始化函数改为相应的 `Set` 初始化，实现的接口“大致”不变（参照实例化程序）。（2分）

- 实例化：

```

1 | int a[] = {2,1,4,5,6,8};
2 | int b[] = {3,2,4,9};
3 | double d[] = {2.0,3.0};
4 | Set<int> s1;
5 | Set<int> s2(a,6);
6 | Set<int> s3(b,4);
7 | s1.insert(1);
8 | cout << (s1.is_empty()) << endl;
9 | s1.insert(5);
10 | cout << (s1.is_element(5)) << endl;
11 | s1.insert(6);
12 | cout << (s1.size()) << endl;

```

```

13 s1.display();
14 cout << (s1.is_subset(s2)) << endl;
15 cout << s1.remove(2) << endl;
16 (s2.union2(s3)).display();
17 (s2.intersection2(s3)).display();
18 (s2.difference2(s3)).display();
19 Set<double> s4(d,2);
20 Set<double> s5;
21 s5.insert(2.0);
22 s5.insert(3.0);
23 s5.display();
24 cout << s5.is_equal(s4) << endl;

```

◦ 实例化结果:

```

0
1
3
1,5,6,
1
0
1,2,3,4,5,6,8,9,
2,4,
1,5,6,8,
2,3,
1

```

◦ 提交说明:

- 将类的定义和实现的代码都放在IntSet.h中。

3. Date是日期类，以char *存储日期（格式为：1997/07/01）。小灰想用Date存储一些有意义的日期，将其放在另一个类meaningful_Date中，请你重构meaningful_Date类并完善Date类。Date类仅有一个私有成员变量，如下：（2分）

```

1 class Date{
2     char *date;
3     // TODO: 补全类的定义 给出相应实现
4 };

```

◦ 实现要求:

- 所有有意义的日期放在meaningful_Date中；
- 每个有意义的日期都对应一个有意义的事件，可以通过事件检索到"日期"，如myDate["birthday"]就对应着出生日期，myDate["first travel with GF"]对应着第一次和女朋友旅行的日期；
- 能提供对重要"日期"增删查改的接口；
- 重要事件不能重复；
- 尽量利用STL实现；
- 可以发挥聪明才智，增加更多功能；

◦ 提交说明：

- 基本的设计说明放在readme.pdf中；
- 类的定义都放在一个文件meaningful_Date.h中，类的实现都放在meaningful_Date.cpp中。