高级程序设计 2020 春

实验报告

指导老师：陈家骏 黄书剑 姓名：王晨渊, 学号：181220057

1 概念题

1.1 C++ 的 STL 提供了哪几种模板？分别简述它们的作用

提供了三种模板：容器模板、算法模板和迭代器模板。容器用于存储数据，迭代器用于访问容器中的元素，算法用于操作容器中的元素。

1.2 列举 STL 中能快速定位（访问）任意位置的容器，并说明它们的内部数据结构。

1.2.1 vector

动态数组

1.2.2 deque

分段的连续空间

1.2.3 map,multimap

某种二叉树

1.2.4 set,multiset

某种二叉树

1.3 简述 C++ 中自定义操作条件（谓词）的概念及作用。

参数为元素类型，返回值为 bool 的函数或函数对象。作为 STL 中算法的操作条件。

2 编程题

2.1 代码补全

```cpp
#include <iostream>
#include <string>
using std::string;
class StrOperation
{
private:
    string str;
public:
    StrOperation(string s);
    bool judgePalindrome();
    void insertStr(int i,string s);
    void replaceStr(int be,int en,string s);
};

StrOperation::StrOperation(string s)
{
    str = s;
}
bool StrOperation::judgePalindrome()
{
    string::const_iterator be = str.begin(),ed = str.end();
    ed--;
    while(be!=ed&&be!=ed+1)
    {
        if(*be!=*ed)
        {
            return false;
        }
        be++;
        ed--;
    }
    return true;
}
void StrOperation::insertStr(int i,string s)
{
    str.insert(i,s);
}
void StrOperation::replaceStr(int be,int en,string s)
{
```

```
40          str.replace(be,en,s);
41      }
```

2.2 动物管理

```
1   #include <iostream>
2   #include <string>
3   #include <map>
4   using std::endl;
5   using std::cout;
6   using std::map;
7   using std::string;
8   class System
9   {
10  public:
11      System();
12      void record(string name, int num);
13      void del(string name);
14      int animal_num(string name);
15      int _kinds();
16      int _total();
17  private:
18      map<string,int> animals;
19      int kinds_of_animals;
20      int total;
21  };
22
23  System::System()
24  {
25      kinds_of_animals=0;
26      total=0;
27  }
28  void System::record(string name, int num)
29  {
30      map<string,int>::iterator it = animals.find(name);
31      cout<<"Record successfully!";
32      if(it!=animals.end())
33      {
34          animals[name] += num;
```

```
35              cout<<"Existed before."<<endl;
36          }
37          else
38          {
39              animals[name] = num;
40              cout<<"Not Existed before."<<endl;
41              kinds_of_animals++;
42          }
43          total+=num;
44      }
45      void System::del(string name)
46      {
47          map<string,int>::iterator it = animals.find(name);
48          if(it!=animals.end())
49          {
50              total -= animals[name];
51              kinds_of_animals--;
52              animals.erase(name);
53              cout<<"Delete successfully"<<endl;
54          }
55          else
56          {
57              cout<<"Deletion failed!Not Existed"<<endl;
58          }
59          // if(animals.erase(name)) cout<<"Delete Successfully"<<endl;
60          // else cout<<"Delete Failed! Animal not exists."<<endl;
61      }
62
63      int System::animal_num(string name)
64      {
65          map<string, int>::const_iterator it = animals.find(name);
66          if(it!=animals.end())
67          {
68              return animals[name];
69          }
70          else
71          {
72              return 0;
73          }
```

```
74      }
75
76      int System::_kinds()
77      {
78          return kinds_of_animals;
79      }
80      int System::_total()
81      {
82          return total;
83      }
```

2.3 图书管理

实践发现，使用 deque 速度远快于 vector，deque 通过测试需要 0.003071 秒，而 vector 需要 0.007055 秒。

```
1       #include <iostream>
2       #include <string>
3       #include <list>
4       #include <stack>
5       #include <map>
6       #include <vector>
7       #include <cassert>
8       #include <algorithm>
9       #include <cstdlib>
10      #include <ctime>
11      using namespace std;
12      #define ADT deque
13      class Book
14      {
15      public:
16          Book(string name, string author,int year, int ID, int num);
17          int get_num() const;
18          int get_ID() const;
19          int get_year() const;
20          void display() const;
21          void display_year() const;
22          const string& get_author() const;
23      private:
24          string _name,_author;
25          int _year;
```

```
26          int _ID;
27          int _num;
28      };
29      Book::Book(string name, string author,int year, int ID, int num)
30      {
31          _name = name;
32          _author = author;
33          _year = year;
34          _ID = ID;
35          _num = num;
36      }
37      int Book::get_num() const
38      {
39          return _num;
40      }
41      int Book::get_ID() const
42      {
43          return _ID;
44      }
45
46      const string& Book::get_author() const
47      {
48          return _author;
49      }
50      void Book::display() const
51      {
52          cout<<"author:"<<_author<<" book name:"<<_name<<" num:"<<_num;
53          cout<<" year:"<<_year<<" ID:"<<_ID<<endl;
54      }
55      int Book::get_year() const
56      {
57          return _year;
58      }
59      void Book::display_year() const
60      {
61          cout<<"Year:"<<_year<<endl;
62      }
63
64      class MatchAuthor
```

```
65      {
66      public:
67          MatchAuthor(const string& author)
68          {
69              _author = author;
70          }
71          void operator()( Book& bk)
72          {
73              if(bk.get_author()==_author)
74              {
75                  bk.display();
76              }
77          }
78      private:
79          string _author;
80      };
81
82      bool cmp_year(Book& bk1,Book bk2)
83      {
84          return bk1.get_year()>bk2.get_year();
85      }
86
87      void display(Book& bk)
88      {
89          bk.display();
90      }
91
92      class Machine
93      {
94      public:
95          Machine();
96          void addBook(int num,const string& name,const string& author,int year);
97          void deleteBook(int ID);
98          int _getID();
99          void display_with_year();
100         void find(const string& author);
101         void _freeID(int id);
102     private:
103         ADT<Book> books;
```

```
104         stack<int> available_IDs;// 书被删除后重新加入需要新的ID
105         int max_id;
106     };
107     Machine::Machine()
108     {
109         max_id=-1;
110     }
111     void Machine::deleteBook(int ID)
112     {
113         ADT<Book>::const_iterator it = books.begin();
114         while(it!=books.end())
115         {
116             if((it->get_ID())==ID)
117             {
118                 books.erase(it);
119                 _freeID(ID);
120                 return;
121             }
122             it++;
123         }
124         cout<<"No this book!"<<endl;
125     }
126
127     void Machine::find(const string& author)
128     {
129         for_each(books.begin(),books.end(),MatchAuthor(author));
130     }
131
132     void Machine::display_with_year()
133     {
134         sort(books.begin(),books.end(),cmp_year);
135         for_each(books.begin(),books.end(),display);
136     }
137
138     void Machine::addBook(int num,const string& name,const string& author,int year)
139     {
140         books.push_back((Book){name,author,year,_getID(),num});
141     }
142
```

```
143
144     int Machine::_getID()
145     {
146         if(available_IDs.empty())
147         {
148             max_id++;
149             return max_id;
150         }
151         int tmp = available_IDs.top();
152         available_IDs.pop();
153         return tmp;
154     }
155     void Machine::_freeID(int id)
156     {
157         available_IDs.push(id);
158     }
159
160
161
162
163     int main()
164     {
165         Machine my_library;
166         clock_t st=clock();
167         for(int i=0;i<120;i++)
168         {
169             my_library.addBook(i,"test","Dad",i+100);
170             my_library.addBook(i,"see","wang",123);
171             my_library.addBook(i,"Pig","Chen",11);
172         }
173         for(int i=0;i<50;i++)
174         {
175             my_library.deleteBook(i);
176         }
177         for(int i=0;i<10;i++)
178         {
179             my_library.addBook(i,"test2","Kitty",10);
180             my_library.addBook(i,"see2","wang",123);
181             my_library.addBook(i,"Pig2","Chen",11);
```

```
182             }
183                 for(int i=100;i<120;i++)
184             {
185                 my_library.deleteBook(i);
186             }
187             my_library.find("Chen");
188             my_library.display_with_year();
189             cout<<"Time consumption:"<<(double)(clock()-st)/CLOCKS_PER_SEC<<endl;
190         }
```