程 序 设 计 A 实 验 指 导

实验UNIT 09 模板和群体数据

《程序设计》课程组

武 汉 大 学 计 算 机 学 院

第9讲上机实验

实验目的：

1. 了解结点类的声明和实现，学习其使用方法；
2. 了解链表类的声明和实现，学习其使用方法；
3. 了解栈类的声明和实现，学习其使用方法；
4. 了解队列类的声明和实现，学习其使用方法；
5. 掌握对数组元素排序的方法；
6. 掌握对数组元素查找的方法。

第9讲上机实验

实验任务：

课堂练习：将插入排序实现为函数模板

编程练习：

1. 编写程序Node.h实现教材例9-5的结点类，并编写程序lab9_1.cpp实现链表类的基本操作。
2. 编写程序Link.h实现教材例9-6的链表类，在测试程序lab9_2.cpp中声明两个整型链表A和B，分别插入5个元素，然后把B中的元素加入到A的尾部。
3. 编写程序queue.h用链表类实现队列（或栈）类，并编写测试程序lab9_3.cpp中声明一个整型队列（或栈）对象，插入5个元素，压入队列（或栈），并依次取出并显示出来。
4. 将直接插入排序、直接选择排序、冒泡排序、顺序查找排序函数封装到第9章的数组类中，作为成员函数，实现并测试这个类。

现在开始课堂练习！

将插入排序时限为函数模板。

第9讲上机练习

```
#include <iostream>
#include <vector>
#include <ctime>
using namespace std;
template<typename T, class compare = std::less<T>>
void insertSort(vector<T>& nums, compare cmp = std::less<T>());
template<typename T>
void printVector(vector<T>& nums);
void generateIntVector(vector<int>& nums, int n);

class Base {
        int x;
public:
        Base() :x(-1) {}
        explicit Base(int m_x) :x(m_x) {}
        friend bool operator< (const Base& b1, const Base& b2);
        friend class cmp1;
        friend ostream& operator<<(ostream& os, const Base& b);
        int getX() { return x; }
};
```

第9讲上机练习

```
bool operator<(const Base & b1, const Base & b2){
        return b1.x < b2.x;
}
ostream& operator<<(ostream& os, const Base& b) {
        os << b.x;
        return os;
}

struct cmp1 {
        bool operator()(Base& b1, Base& b2) {
                return b1.x < b2.x;
        }
};
```

第9讲上机练习

```
int main(){
        srand((int)time(0));
        vector<int> nums1;
        generateIntVector(nums1, 10);
        cout << "Initial value of nums1: ";
        printVector<int>(nums1);
        insertSort<int>(nums1,greater<int>());

        cout << "Sorted value of nums1: ";
        printVector<int>(nums1);
        vector<Base> baseArr;
        for (int i = 0; i < 15; i++) {
                int tmpVal = rand() % 100;
                Base tmpBase(tmpVal);
                baseArr.push_back(tmpBase);
        }
        cout<<"Initial value of baseArr: ";
        printVector<Base>(baseArr);
        insertSort<Base>(baseArr, [](Base b1, Base b2)mutable -> bool {return
        b1.getX() < b2.getX(); });
        cout << "Sorted value of baseArr: ";
        printVector<Base>(baseArr);
        return 0;
}
```

第9讲上机练习

```
template<typename T>
void printVector(vector<T>& nums) {
        if (nums.empty()) {
                cout << "The vector is empty!" << endl;
                return;
        }
        cout << nums[0];
        for (int i = 1; i < nums.size(); i++)
                cout << " " << nums[i];
        cout << endl;
}

void generateIntVector(vector<int>& nums, int n) {
        int tmp = 0;
        for (int i = 0; i < n; i++) {
                tmp = rand() % 100;
                nums.push_back(tmp);
        }
}
```

第9讲上机练习

```
template<typename T, class compare>
void insertSort(vector<T>& nums, compare cmp) {
        int n = nums.size();
         i = 1, j = 1;
        T tmp;
        for (; i < n; i++) {
                tmp = nums[i];
                for (j = i - 1; j >= 0 && cmp(tmp, nums[j]); j--)
                        nums[j + 1] = nums[j];
                }
                nums[j + 1] = tmp;
        }
}
```

Wuhan University

本次课堂练习结束！

第9讲上机任务

编程练习：

1. 编写程序Node.h实现教材例9-5的结点类，并编写程序lab9_1.cpp实现链表类的基本操作。
2. 编写程序Link.h实现教材例9-6的链表类，在测试程序lab9_2.cpp中声明两个整型链表A和B，分别插入5个元素，然后把B中的元素加入到A的尾部。
3. 编写程序queue.h用链表类实现队列（或栈）类，并编写测试程序lab9_3.cpp中声明一个整型队列（或栈）对象，插入5个元素，压入队列（或栈），并依次取出并显示出来。
4. 将直接插入排序、直接选择排序、冒泡排序、顺序查找排序函数封装到第9章的数组类中，作为成员函数，实现并测试这个类。

第9讲上机练习

- 实验步骤提示：

1. 参照例9-5中结点类Node的声明（9-5.h），给出其实现。在测试程序中从键盘4输入10个整数，用这些整数值作为结点数据，生成一个链表，按顺序输出链表中结点的数值。然后从键盘输入一个待查找整数，在链表中查找该整数，若找到则删除该整数所在的结点（如果出现多次，全部删除），然后输出删除结点以后的链表。在程序结束之前清空链表。
2. 参照例9-6中链表类LinkedList的声明（9-6.h），给出其实现，注意合理使用Node类的成员函数。在测试程序中声明整数链表A和B，分别插入5个元素，使用循环语句显示链表中的元素，然后把B中的元素加入A的尾部，再显示出来。

第9讲上机练习

- 实验步骤提示：

3. 队列类的特点是其元素参照顺序为先入先出（FIFO），用任务2中的链表类实现队列类，用链表类的成员函数实现队列类的成员函数，在测试程序中2声明一个整型队列对象，观察队列类中元素先入先出的特点。

4. 编写程序array1.h声明并实现数组类，其中包含成员函数void insertSort()实现直接插入排序；成员函数void selectSort()实现直接选择排序；成员函数void BubbleSort()实现冒泡排序；成员函数int seqSearch(T Key)实现顺序查找；这些函数操作的数据就是数组类的数据成员alist。在测试程序lb9-4.cpp中声明数组类的对象，测试这些成员函数。

本 讲 结 束