#ifndef SEQLIST\_H\_INCLUDED

#define SEQLIST\_H\_INCLUDED

const int MaxSize=100;

template<class DataType>

class SeqList

{

public:

SeqList(){length=0;}

SeqList(DataType a[], int n);

~SeqList(){};

int Length(){return length;}

DataType Get(int i);

int Locate(DataType x);

void Insert(int i,DataType x);

DataType Delete(int i);

int Empty( );

void PrintList( );

private:

DataType data[MaxSize];

int length;

};

template <typename DataType>

SeqList<DataType>::SeqList(DataType a[], int n)

{

if (n>MaxSize) throw "参数非法";

for (int i = 0; i< n; i++)

data[i] = a[i];

length = n;

}

template <typename DataType>

int SeqList<DataType>::Locate(DataType x)

{

for (int i = 0;i < length; i++)

if (data[i] == x) return i+1; //返回其序号i+1

return 0; //退出循环，说明查找失败

}

template <typename DataType>

void SeqList<DataType>::Insert(int i,DataType x)

{

if (length == MaxSize) throw "上溢";

if (i < 1 || i > length + 1) throw "插入位置错误";

for (int j = length; j >= i; j--)

data[j] = data[j-1]; //第j个元素存在数组下标为j-1处

data[i-1] = x;

length++;

}

template <typename DataType>

DataType SeqList<DataType>::Delete(int i)

{

if (length == 0) throw "下溢";

if (i < 1||i > length) throw "位置错误";

int x = data[i-1];

for (int j = i; j <length; j++)

data[j-1] = data[j]; //第j个元素存在数组下标为j-1处

length--;

return x;

}

template <class DataType>

DataType SeqList<DataType>::Get(int i)

{

if(i<1 && i>length) throw "查找位置非法";

else

return data[i-1];

}

template <typename DataType>

void SeqList<DataType>::PrintList()

{

for (int i=length-1;i>=0;i--)

cout<<data[i]<<"";

cout<< endl;

#include <iostream>

using namespace std;

#include "SeqList.h"

int main()

{

int r[5]={1,2,3,4,5};

SeqList<int> L(r,5);

cout<< "执行插入操作前数据为:"<<endl;

L.PrintList();

。。。。

}