**IntArray**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BaiTap2

{

public class IntArray

{

private int[] array;

public int[] Array

{

get { return array; }

set { array = value; }

}

public IntArray()

{

array = new int[0];

}

public IntArray(int k)

{

if (k <= 0 || k > 1000000)

{

Console.WriteLine("Vui long nhap so khac");

}

array = new int[k];

Random random = new Random();

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

{

array[i] = random.Next(1, 201);

}

}

public IntArray(int[] a)

{

array = new int[a.Length];

a.CopyTo(array, 0);

}

public IntArray(IntArray obj)

{

array = new int[obj.Array.Length];

obj.Array.CopyTo(array, 0);

}

public void Input()

{

Console.WriteLine("Nhap so phan tu cho mang: ");

int n=int.Parse(Console.ReadLine());

array=new int[n];

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

{

Console.Write($"Nhập giá trị cho phần tử thứ {i + 1}: ");

array[i] = int.Parse(Console.ReadLine());

}

}

public void Output()

{

Console.WriteLine("Các phần tử trong mảng:");

for (int i = 0; i < array.Length; i++)

{

Console.WriteLine($"Phần tử thứ {i + 1}: {array[i]}");

}

}

public int SequentialSearch(int x)

{

for (int i = 0; i < array.Length; i++)

{

if (array[i] == x)

{

return i;

}

}

return -1;

}

public int BinarySearch(int x)

{

int left = 0;

int right = array.Length - 1;

while (left <= right)

{

int mid = (left + right) / 2;

if (array[mid] == x)

{

return mid;

}

else if (array[mid] < x)

{

left = mid + 1;

}

else

{

right = mid - 1;

}

}

return -1;

}

public void BurbleSort()

{

for (int i = 0; i <= array.Length - 1; i++)

{

for (int j = 0; j < array.Length - 1; j++)

{

if (array[j] > array[j + 1])

{

int temp = array[j];

array[j] = array[j + 1];

array[j + 1] = temp;

}

}

}

}

}

}

**Program**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BaiTap2

{

internal class Program

{

static void TestConstructor1()

{

IntArray obj = new IntArray(20);

Console.WriteLine("Gia tri mang phat sinh: ");

obj.Output();

}

static void TestConstructor2()

{

int[] a = { 4, 7, 9, 10, 20, 8, 3, 17, 10, 6 };

IntArray obj = new IntArray(a);

Console.WriteLine("Gia tri mang: ");

obj.Output();

}

static void TestConstructor3()

{

IntArray obj1=new IntArray();

obj1.Input();

Console.WriteLine("Gia tri mang: ");

obj1.Output();

IntArray obj2 = new IntArray(obj1);

Console.WriteLine("Gia tri mang copy: ");

obj2.Output();

}

static void TestTimTuanTu()

{

int k, x, kq;

Console.Write("Nhap so luong phan tu: ");

int.TryParse(Console.ReadLine(), out k); ;

IntArray obj=new IntArray(k);

Console.WriteLine("Cac phan tu: ");

obj.Output();

Console.WriteLine("Gia tri can tim x= ");

int.TryParse(Console.ReadLine(), out x);

kq = obj.SequentialSearch(x);

if (kq == -1)

Console.WriteLine($"Khong ton tai {x}");

else

Console.WriteLine($"Co {x} tai vi tri {kq}");

}

static void TestTimNhiPhan()

{

int k, x, kq;

Console.Write("Nhap so luong phan tu: ");

int.TryParse(Console.ReadLine(), out k);

IntArray obj = new IntArray(k);

obj.BurbleSort();

Console.WriteLine("Cac phan tu: ");

obj.Output();

Console.WriteLine("Gia tri can tim x= ");

int.TryParse(Console.ReadLine(), out x);

kq = obj.BinarySearch(x);

if (kq == -1) Console.WriteLine($"Khong ton tai {x}");

else Console.WriteLine($"Ton tai {x} tai vi tri {kq}");

}

static void Main(string[] args)

{

//TestConstructor1();

//TestConstructor2();

//TestConstructor3();

//TestTimTuanTu();

TestTimNhiPhan();

Console.ReadKey();

}

}

}

**MangSinhVien**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BaiTap1

{

internal class MangSinhVien:SinhVien

{

private SinhVien[] danhSachSinhVien;

private int soLuongSinhVien;

public MangSinhVien(int n)

{

danhSachSinhVien = new SinhVien[n];

soLuongSinhVien = 0;

}

public void NhapDanhSach()

{

for (int i = 0; i < danhSachSinhVien.Length; i++)

{

Console.WriteLine($"Nhập thông tin cho sinh viên thứ {i + 1}:");

SinhVien sv = new SinhVien();

do

{

Console.Write("Mã số sinh viên: ");

sv.maso = Console.ReadLine();

}

while (TonTai(sv.maso, i) == true) ;

Console.Write("Họ và tên: ");

sv.hoten = Console.ReadLine();

Console.Write("Chuyên ngành: ");

sv.chuyennganh = Console.ReadLine();

Console.Write("Năm sinh: ");

sv.namsinh = int.Parse(Console.ReadLine());

Console.Write("Điểm trung bình tích lũy: ");

sv.diemtb = float.Parse(Console.ReadLine());

danhSachSinhVien[i] = sv;

}

soLuongSinhVien = danhSachSinhVien.Length;

}

public void XuatDanhSach()

{

Console.WriteLine("\nDanh sách sinh viên:");

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

{

Console.WriteLine($"Sinh viên thứ {i + 1}:");

Console.WriteLine("Mã số sinh viên: " + danhSachSinhVien[i].maso);

Console.WriteLine("Họ và tên: " + danhSachSinhVien[i].hoten);

Console.WriteLine("Chuyên ngành: " + danhSachSinhVien[i].chuyennganh);

Console.WriteLine("Năm sinh: " + danhSachSinhVien[i].namsinh);

Console.WriteLine("Điểm trung bình tích lũy: " + danhSachSinhVien[i].diemtb);

Console.WriteLine();

}

}

public bool TonTai( string msx,int vt)

{

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

{

if (danhSachSinhVien[i].maso.CompareTo(msx) == 0) ;

return true;

}

return false;

}

}

}