using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BT2

{

public class IntArray

{

private int[] arr;

public int[] Arr

{

get { return arr; }

set { arr = value; }

}

public IntArray()

{

arr = new int[0];

}

public IntArray(int k)

{

Random rand = new Random();

arr = Enumerable.Range(1, k).Select(\_ => rand.Next(1, 201)).ToArray();

}

public IntArray(int[] a)

{

arr = new int[a.Length];

Array.Copy(a, arr, a.Length);

}

public IntArray(IntArray obj)

{

arr = new int[obj.arr.Length];

Array.Copy(obj.arr, arr, obj.arr.Length);

}

public void Input()

{

Console.Write("Enter the size of the array: ");

int n = Convert.ToInt32(Console.ReadLine());

arr = new int[n];

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

{

Console.Write($"Enter element {i + 1}: ");

arr[i] = Convert.ToInt32(Console.ReadLine());

}

}

public void Output()

{

foreach (var item in arr)

Console.Write($"{item} ");

Console.WriteLine();

}

public int SequentialSearch(int x)

{

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

if (arr[i] == x)

return i;

return -1;

}

public int BinarySearch(int x)

{

int left = 0, right = arr.Length - 1;

while (left <= right)

{

int mid = left + (right - left) / 2;

if (arr[mid] == x)

return mid;

if (arr[mid] < x)

left = mid + 1;

else

right = mid - 1;

}

return -1;

}

}

class Program

{

static void Main(string[] args)

{

Console.Write("Enter the size of the array: ");

int k = Convert.ToInt32(Console.ReadLine());

IntArray objA = new IntArray(k);

Console.WriteLine("The random values in objA are: ");

objA.Output();

Console.Write("Enter the value to find: ");

int x = Convert.ToInt32(Console.ReadLine());

int index = objA.SequentialSearch(x);

if (index != -1)

Console.WriteLine($"Found {x} at index {index} in objA.");

else

Console.WriteLine($"{x} is not found in objA.");

IntArray objB = new IntArray();

Console.WriteLine("Enter values for objB in ascending order: ");

objB.Input();

Console.Write("Enter the value to find: ");

x = Convert.ToInt32(Console.ReadLine());

index = objB.BinarySearch(x);

if (index != -1)

Console.WriteLine($"Found {x} at index {index} in objB.");

else

Console.WriteLine($"{x} is not found in objB.");

Console.Write("Nhap so luong sinh vien: ");

int n = Convert.ToInt32(Console.ReadLine());

Sinhvien[] MangSinhVien = new Sinhvien[n];

Console.WriteLine("Nhap thong tin cho cac sinh vien:");

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

{

MangSinhVien[i] = new Sinhvien();

MangSinhVien[i].nhap(MangSinhVien.Take(i).ToArray());

}

Console.WriteLine();

Console.WriteLine("Danh sach sinh vien:");

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

{

MangSinhVien[i].xuat();

}

TestSV();

}

static void TestSV()

{

Sinhvien svA = new Sinhvien();

Console.WriteLine("Nhập thông tin cho svA:");

svA.nhap(new Sinhvien[0]);

Console.WriteLine("Thông tin svA:");

svA.xuat();

Sinhvien svB = new Sinhvien("18DH001", "Lam Thanh Ngoc", "CNPM", 2000, 7.6f, "");

Console.WriteLine("Thông tin svB:");

svB.xuat();

Sinhvien svC = new Sinhvien();

Console.WriteLine("Nhập họ tên và điểm trung bình tích lũy để cập nhật thông tin cho svC:");

svC.nhap(new Sinhvien[0]);

Console.WriteLine("Thông tin svC:");

svC.xuat();

Console.WriteLine("Thông tin svB sau khi cập nhật thông tin cho svC:");

svB.xuat();

}

}

class Sinhvien

{

private string maSo;

private string hoTen;

private string chuyenNganh;

private int namSinh;

private float diemTB;

private string loai;

public Sinhvien()

{

}

public Sinhvien(string maSo, string hoTen, string chuyenNganh, int namSinh, float diemTB, string loai)

{

this.maSo = maSo;

this.hoTen = hoTen;

this.chuyenNganh = chuyenNganh;

this.namSinh = namSinh;

this.diemTB = diemTB;

this.loai = loai;

}

public void xepLoai()

{

this.loai = this.diemTB < 5 ? "Kem" : (this.diemTB < 7 ? "Trung binh" : (this.diemTB < 8 ? "Kha" : "Gioi"));

}

public bool KTtuoi()

{

int currentyear = DateTime.Now.Year;

int age = currentyear - this.namSinh;

return age > 17 && age <= 70;

}

public void nhap(Sinhvien[] existingStudents)

{

bool isUnique;

do

{

Console.Write("Nhap ma so sinh vien: ");

maSo = Console.ReadLine();

isUnique = !existingStudents.Any(s => s.maSo == maSo);

if (!isUnique)

{

Console.WriteLine("Ma so sinh vien da ton tai. Vui long nhap lai.");

}

} while (!isUnique);

// Rest of the code...

}

public void xuat()

{

Console.WriteLine();

Console.WriteLine($"Ma so sinh vien: {maSo}");

Console.WriteLine($"Ho ten sinh vien: {hoTen}");

Console.WriteLine($"Chuyen nganh: {chuyenNganh}");

Console.WriteLine($"Nam sinh: {namSinh}");

Console.WriteLine($"Diem trung binh: {diemTB}");

Console.WriteLine($"Xep loai: {loai}");

Console.WriteLine("");

}

// Rest of the class...

}

}