using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace Delegates

{

//public class Student

//{

// public Student(string name, string surname, DateTime birthdate)

// {

// Name = name;

// Surname = surname;

// Birthdate = birthdate;

// }

// public string Name { get; set; }

// public string Surname { get; set; }

// public DateTime Birthdate { get; set; }

// public override string ToString()

// {

// return Name+" "+Surname+" " + Birthdate.ToLongDateString();

// }

//}

//public class Program

//{

// static bool onlyWinter(Student s)

// {

// return Array.IndexOf(new int[3] { 12, 1, 2 }, s.Birthdate.Month) > -1;

// }

// static bool onlySpring(Student s)

// {

// return Array.IndexOf(new int[3] { 3,4,5 }, s.Birthdate.Month) > -1;

// }

// static void Display(Student s)

// {

// Console.WriteLine(s.Name);

// Console.WriteLine(s.Surname);

// Console.WriteLine();

// }

// static void Main(string[] args)

// {

// List<Student> students = new List<Student>

// {

// new Student("Ilkin","Suleymanoc",new DateTime(2005,10,23)),

// new Student("Ayxan","Ahmedzade",new DateTime(2006,2,23)),

// new Student("Huseyn","Abbasov",new DateTime(2005,4,23)),

// new Student("Mehemmed","Bayramov",new DateTime(2004,5,2)),

// new Student("Coshqun","Gulmemmedli",new DateTime(2003,04,24)),

// new Student("Ali","Ahmedov",new DateTime(2001,10,3)),

// new Student("Omer","Cavanshirli",new DateTime(1995,7,26)),

// new Student("Nurlan","Shirinov",new DateTime(1998,2,16)),

// };

// //Predicate<Student> onlyWinterStudents = new Predicate<Student>(onlyWinter);

// //var result = Array.FindAll(students.ToArray(),onlyWinterStudents);

// //foreach (var item in result)

// //{

// // Console.WriteLine(item);

// //}

// //Predicate<Student> onlySpringStudents = new Predicate<Student>(onlySpring);

// //var result = Array.FindAll(students.ToArray(), onlySpringStudents);

// //foreach (var item in result)

// //{

// // Console.WriteLine(item);

// //}

// //Predicate<Student>pred = (s) =>

// // {

// // return Array.IndexOf(new int[] { 3, 4, 5 }, s.Birthdate.Month) > -1;

// // };

// //var result = Array.FindAll(students.ToArray(), (s) =>

// //{

// // return Array.IndexOf(new int[] { 3,4,5}, s.Birthdate.Month) > -1;

// //});

// ////Func<Student, string> myfunc = (s) =>

// ////{

// //// return s.Name + " " + s.Surname;

// ////};

// ////var specialResult = result.Select(myfunc).ToList();

// //var specialResult = result.Select((s) =>

// //{

// // return s.Name + " " + s.Surname;

// //}).ToList();

// //foreach (var item in specialResult)

// //{

// // Console.WriteLine(item);

// //}

// //foreach (var item in result)

// //{

// // Console.WriteLine(item);

// //}

// // students.ForEach(Display);

// //var result = Array.FindAll(students.ToArray(), (s) =>

// //{

// // return Array.IndexOf(new int[] { 3, 4, 5 }, s.Birthdate.Month) > -1;

// //});

// //var studentsResult = result.Select((s) =>

// //{

// // return s.Name + " " + s.Surname;

// //}).ToList();

// //studentsResult.ForEach((s) =>

// //{

// // Console.WriteLine(s);

// //});

// // Debtor fullname,birthdate,debt,email,hasLate,bank

// //1. ilin birinci rubunde dogulan ve email sonu gmail.com

// //2. borcu 1000-1500 arasi olanlari gosterin

// //3. gecikmesi olan ve yaz aylarinda dogulanlari gosterin

// //4. gecikmesi olmayan ve borcu 2500 den yuxari olanlari gostermek

// //5. Kapital bank olan ve borcu 30000 den yuxari olanlari gosterin

// }

//}

public delegate void Notify();

class ProcessBusinessLogic

{

public event Notify ProcessCompleted;

public void StartProcess()

{

Console.WriteLine("Process starting");

Thread.Sleep(2000);

OnProcessCompleted();

}

private void OnProcessCompleted()

{

ProcessCompleted?.Invoke();

}

}

class Program

{

static void Main(string[] args)

{

ProcessBusinessLogic businessLogic = new ProcessBusinessLogic();

businessLogic.ProcessCompleted += BusinessLogic\_ProcessCompleted;

businessLogic.StartProcess();

}

private static void BusinessLogic\_ProcessCompleted()

{

Console.WriteLine("Process Completed");

Process.Start("calc.exe");

}

}

}