Universidad Modelo

Logotipo

Descripción generada automáticamente

Escuela de Ingeniería.

Carrera: Ingeniería en Desarrollo de Tecnología y Software

Asignatura: Ingeniería en software

Nombre del Profesor: Iván Flores Jimenez

Nombre de la Actividad: Tarea 1

Fecha de Entrega: 30/08/2022

Nombre del Alumno:

Roberto Gamboa

Rubén Gamboa

Sebastián Ramos

Índice

Análisis: Pagina 1

Diseño: Pagina 2

Programación/Código: Pagina 3

Pruebas: Pagina 4

Análisis

El objetivo de este trabajo es encontrar el lapso de descanso más largo que una persona tiene entre reuniones.

El formato que se da el horario de cada día consiste en “Ddd hh:mm-hh:mm” y los días de la semana se darán de manera aleatoria por lo que se necesitará una de también ordenarlas. El programa debe devolver el lapso de descanso más largo, cada reunión puede durar un mínimo de un minuto.

Diseño

1.- El diseño del programa deberá empezar por separar la cadena por cada día, hora de inicio, minutos de inicio y horas de fin, al igual que minutos y meterla dentro de una lista, la cual permitirá que las fechas se manipulen más fácilmente.

2.- Las variables separadas se guardan en listas de objetos de una clase que se creará para guardar los datos.

3.- El programa deberá realizar operaciones con las fechas para encontrar los lapsos de descanso que se encuentran entre ellos.

4.- El programa compara todos los tiempos de descanso para encontrar el mayor o mayores.

5.- El programa devuelve el tiempo de descanso o descansos más largo/s.

Código

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Jamesapp

{

class Program

{

static void Main(string[] args)

{

string hellSchedule = "Sun 10:00-20:00\nFri 05:00-10:00\nFri 16:30-23:50\nSat 10:00-24:00\nSun 01:00-04:00\nSat 02:00-06:00\nTue 03:30-18:15\nTue 19:00-20:00\nWed 04:25-15:14\nWed 15:14-22:40\nThu 00:00-23:59\nMon 05:00-13:00\nMon 15:00-21:00";

var listHell = hellSchedule.Split('\n');

var sortedHellList = listHell.ToList();

List<Tiempo> lista = new List<Tiempo>();

List<Tiempo> Nlista = new List<Tiempo>();

List<Tiempo> descansos = new List<Tiempo>();

foreach (var date in sortedHellList)

{

Tiempo temp = new Tiempo();

var list = date.Split(' ');

temp.Day = list[0];

var tiempos = list[1].Split('-');

var tiempos2 = tiempos[0].Split(':');

var tiempos3 = tiempos[1].Split(':');

temp.Day = list[0];

temp.InicioH = tiempos2[0];

temp.InicioM = tiempos2[1];

temp.FinH = tiempos3[0];

temp.FinM = tiempos3[1];

lista.Add(temp);

}

foreach (var record in lista)

{

if (record.Day == "Sun")

{

Nlista.Add(record);

}

}

foreach (var record in lista)

{

if (record.Day == "Mon")

{

Nlista.Add(record);

}

}

foreach (var record in lista)

{

if (record.Day == "Tue")

{

Nlista.Add(record);

}

}

foreach (var record in lista)

{

if (record.Day == "Wed")

{

Nlista.Add(record);

}

}

foreach (var record in lista)

{

if (record.Day == "Thu")

{

Nlista.Add(record);

}

}

foreach (var record in lista)

{

if (record.Day == "Fri")

{

Nlista.Add(record);

}

}

foreach (var record in lista)

{

if (record.Day == "Sat")

{

Nlista.Add(record);

}

}

foreach (var t in Nlista)

{

if (t.Day == "Sun")

{

foreach (var x in Nlista)

{

if (x.Day == "Sun")

{

if (Convert.ToInt32(t.InicioH) > Convert.ToInt32(x.InicioH))

{

Tiempo rest = new Tiempo();

var timeH = Convert.ToInt32(t.FinH) - Convert.ToInt32(x.InicioH);

var timeM = Convert.ToInt32(t.FinM) - Convert.ToInt32(x.InicioM);

rest.Day = t.Day;

rest.InicioH = Convert.ToString(timeH);

rest.InicioM = Convert.ToString(timeM);

descansos.Add(rest);

}

else

{

Tiempo rest = new Tiempo();

var timeH = Convert.ToInt32(x.FinH) - Convert.ToInt32(t.InicioH);

var timeM = Convert.ToInt32(x.FinM) - Convert.ToInt32(t.InicioM);

rest.Day = x.Day;

rest.InicioH = Convert.ToString(timeH);

rest.InicioM = Convert.ToString(timeM);

descansos.Add(rest);

}

}

if (x.Day == "Mon")

{

if (Convert.ToInt32(t.InicioH) > Convert.ToInt32(x.InicioH))

{

Tiempo rest = new Tiempo();

var timeH = Convert.ToInt32(t.FinH) - Convert.ToInt32(x.InicioH);

var timeM = Convert.ToInt32(t.FinM) - Convert.ToInt32(x.InicioM);

rest.Day = t.Day;

rest.InicioH = Convert.ToString(timeH);

rest.InicioM = Convert.ToString(timeM);

descansos.Add(rest);

}

else

{

Tiempo rest = new Tiempo();

var timeH = Convert.ToInt32(x.FinH) - Convert.ToInt32(t.InicioH);

var timeM = Convert.ToInt32(x.FinM) - Convert.ToInt32(t.InicioM);

rest.Day = x.Day;

rest.InicioH = Convert.ToString(timeH);

rest.InicioM = Convert.ToString(timeM);

descansos.Add(rest);

}

}

if (x.Day == "Tue")

{

if (Convert.ToInt32(t.InicioH) > Convert.ToInt32(x.InicioH))

{

Tiempo rest = new Tiempo();

var timeH = Convert.ToInt32(t.FinH) - Convert.ToInt32(x.InicioH);

var timeM = Convert.ToInt32(t.FinM) - Convert.ToInt32(x.InicioM);

rest.Day = t.Day;

rest.InicioH = Convert.ToString(timeH);

rest.InicioM = Convert.ToString(timeM);

descansos.Add(rest);

}

else

{

Tiempo rest = new Tiempo();

var timeH = Convert.ToInt32(x.FinH) - Convert.ToInt32(t.InicioH);

var timeM = Convert.ToInt32(x.FinM) - Convert.ToInt32(t.InicioM);

rest.Day = x.Day;

rest.InicioH = Convert.ToString(timeH);

rest.InicioM = Convert.ToString(timeM);

descansos.Add(rest);

}

}

if (x.Day == "Wed")

{

if (Convert.ToInt32(t.InicioH) > Convert.ToInt32(x.InicioH))

{

Tiempo rest = new Tiempo();

var timeH = Convert.ToInt32(t.FinH) - Convert.ToInt32(x.InicioH);

var timeM = Convert.ToInt32(t.FinM) - Convert.ToInt32(x.InicioM);

rest.Day = t.Day;

rest.InicioH = Convert.ToString(timeH);

rest.InicioM = Convert.ToString(timeM);

descansos.Add(rest);

}

else

{

Tiempo rest = new Tiempo();

var timeH = Convert.ToInt32(x.FinH) - Convert.ToInt32(t.InicioH);

var timeM = Convert.ToInt32(x.FinM) - Convert.ToInt32(t.InicioM);

rest.Day = x.Day;

rest.InicioH = Convert.ToString(timeH);

rest.InicioM = Convert.ToString(timeM);

descansos.Add(rest);

}

}

if (x.Day == "Thu")

{

if (Convert.ToInt32(t.InicioH) > Convert.ToInt32(x.InicioH))

{

Tiempo rest = new Tiempo();

var timeH = Convert.ToInt32(t.FinH) - Convert.ToInt32(x.InicioH);

var timeM = Convert.ToInt32(t.FinM) - Convert.ToInt32(x.InicioM);

rest.Day = t.Day;

rest.InicioH = Convert.ToString(timeH);

rest.InicioM = Convert.ToString(timeM);

descansos.Add(rest);

}

else

{

Tiempo rest = new Tiempo();

var timeH = Convert.ToInt32(x.FinH) - Convert.ToInt32(t.InicioH);

var timeM = Convert.ToInt32(x.FinM) - Convert.ToInt32(t.InicioM);

rest.Day = x.Day;

rest.InicioH = Convert.ToString(timeH);

rest.InicioM = Convert.ToString(timeM);

descansos.Add(rest);

}

}

if (x.Day == "Fri")

{

if (Convert.ToInt32(t.InicioH) > Convert.ToInt32(x.InicioH))

{

Tiempo rest = new Tiempo();

var timeH = Convert.ToInt32(t.FinH) - Convert.ToInt32(x.InicioH);

var timeM = Convert.ToInt32(t.FinM) - Convert.ToInt32(x.InicioM);

rest.Day = t.Day;

rest.InicioH = Convert.ToString(timeH);

rest.InicioM = Convert.ToString(timeM);

descansos.Add(rest);

}

else

{

Tiempo rest = new Tiempo();

var timeH = Convert.ToInt32(x.FinH) - Convert.ToInt32(t.InicioH);

var timeM = Convert.ToInt32(x.FinM) - Convert.ToInt32(t.InicioM);

rest.Day = x.Day;

rest.InicioH = Convert.ToString(timeH);

rest.InicioM = Convert.ToString(timeM);

descansos.Add(rest);

}

}

if (x.Day == "Sat")

{

if (Convert.ToInt32(t.InicioH) > Convert.ToInt32(x.InicioH))

{

Tiempo rest = new Tiempo();

var timeH = Convert.ToInt32(t.FinH) - Convert.ToInt32(x.InicioH);

var timeM = Convert.ToInt32(t.FinM) - Convert.ToInt32(x.InicioM);

rest.Day = t.Day;

rest.InicioH = Convert.ToString(timeH);

rest.InicioM = Convert.ToString(timeM);

descansos.Add(rest);

}

else

{

Tiempo rest = new Tiempo();

var timeH = Convert.ToInt32(x.FinH) - Convert.ToInt32(t.InicioH);

var timeM = Convert.ToInt32(x.FinM) - Convert.ToInt32(t.InicioM);

rest.Day = x.Day;

rest.InicioH = Convert.ToString(timeH);

rest.InicioM = Convert.ToString(timeM);

descansos.Add(rest);

}

}

Console.WriteLine(descansos);

Console.ReadLine();

}

}

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Jamesapp

{

class Tiempo

{

private string day;

private string inicioH;

private string inicioM;

private string finH;

private string finM;

public string Day { get => day; set => day = value; }

public string InicioH { get => inicioH; set => inicioH = value; }

public string InicioM { get => inicioM; set => inicioM = value; }

public string FinH { get => finH; set => finH = value; }

public string FinM { get => finM; set => finM = value; }

}

}

Pruebas

string hellSchedule = "Sun 10:00-20:00\nFri 05:00-10:00\nFri 16:30-23:50\nSat 10:00-24:00\nSun 01:00-04:00\nSat 02:00-06:00\nTue 03:30-18:15\nTue 19:00-20:00\nWed 04:25-15:14\nWed 15:14-22:40\nThu 00:00-23:59\nMon 05:00-13:00\nMon 15:00-21:00";