1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp9

{

class Mat

{

public double a;

public double b;

public double c;

public double x;

public double d;

public double x1;

public double x2;

public Mat(double a, double b, double c, double x, double d, double x1, double x2)

{

this.a = a;

this.b = b;

this.c = c;

this.x = x;

}

public double IniD()

{

double x1;

d = Math.Pow(b, 2) - 4 \* a \* c;

if (d > 0)

{

IniX1();

IniX2();

}

return (0);

}

private double IniX1()

{

return x1 = (-b + d) / (2 \* a);

}

private double IniX2()

{

return x1 = (-b - d) / (2 \* a);

}

public void Vyv()

{

Console.WriteLine("Дискриминант: " + d);

if (d > 0)

{

Console.WriteLine("Корень x1: " + x1);

Console.WriteLine("Корень x2: " + x2);

}

else

Console.WriteLine("Действительных корней нет");

Console.WriteLine();

}

}

//Заданы три квадратных уравнения АХ2 + BX + С = 0; DX2 + FX + E = 0 и ZX2 + YX + S = 0.

//Написать программу нахождения максимального значения корня среди действительных корней этих уравнений.

class Program

{

static void Main(string[] args)

{

double X = 3;

double A = 1, D = 2, Z = -2;

double B = 5, F = 4, Y = 3;

double C = 2, E = 3, S = 4;

Mat m1 = new Mat(A, B, C, X, 0, 0, 0);

Mat m2 = new Mat(D, F, E, X, 0, 0, 0);

Mat m3 = new Mat(Z, Y, S, X, 0, 0, 0);

m1.IniD();

m1.Vyv();

m2.IniD();

m2.Vyv();

m3.IniD();

m3.Vyv();

if (m3.d > 0)

{

if (m3.x1 > m3.x2 && m3.x1 > m2.x1 && m3.x1 > m2.x2 && m3.x1 > m1.x2 && m3.x1 > m1.x1)

Console.WriteLine("Наибольший корень: {0}", m3.x1);

if (m3.x2 > m3.x1 && m3.x1 > m2.x1 && m3.x1 > m2.x2 && m3.x1 > m1.x2 && m3.x1 > m1.x1)

Console.WriteLine("Наибольший корень: {0}", m3.x1);

}

if (m2.d > 0)

{

if (m2.x1 > m3.x2 && m2.x1 > m3.x1 && m2.x1 > m2.x2 && m2.x1 > m1.x2 && m2.x1 > m1.x1)

Console.WriteLine("Наибольший корень: {0}", m2.x1);

if (m2.x2 > m3.x2 && m2.x2 > m3.x1 && m2.x2 > m2.x1 && m2.x2 > m1.x2 && m2.x2 > m1.x1)

Console.WriteLine("Наибольший корень: {0}", m2.x1);

}

if (m1.d > 0)

{

if (m1.x1 > m3.x2 && m1.x1 > m3.x1 && m1.x1 > m2.x2 && m1.x1 > m2.x1 && m1.x1 > m1.x2)

Console.WriteLine("Наибольший корень: {0}", m1.x1);

if (m1.x2 > m3.x2 && m1.x2 > m3.x1 && m1.x2 > m2.x1 && m1.x2 > m2.x2 && m1.x2 > m1.x1)

Console.WriteLine("Наибольший корень: {0}", m1.x1);

}

}

}

}

2

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp9\_

{

class Time

{

public int h;

public int m;

public int s;

public string hms;

public DateTime st;

public TimeSpan sec;

public Time(int h, int m, int s, string hms, DateTime st, TimeSpan sec)

{

this.h = h;

this.m = m;

this.s = s;

this.hms = hms;

this.st = st;

this.sec = sec;

}

public void Info()

{

Console.WriteLine("Время числами - {0}:{1}:{2}", h, m, s);

Console.WriteLine("Время строкой - " + hms);

Console.WriteLine("Время датой - {0}", st.ToLongTimeString());

Console.WriteLine("Время секундами - {0}", sec);

}

protected int SecInt()

{

return h \* 3600 + m \* 60 + s;

}

protected int SecStr()

{

return int.Parse(hms.Substring(0, 2)) \* 3600 + int.Parse(hms.Substring(3, 2)) \* 60 + int.Parse(hms.Substring(6, 2));

}

protected double SecDT()

{

DateTime dt1970 = new DateTime(2017, 12, 23, 0, 0, 0);

TimeSpan tsInterval = st.Subtract(dt1970);

return tsInterval.TotalSeconds;

}

public int SecR()

{

return SecInt() - SecStr();

}

public void SecS()

{

st.Add(sec);

Console.WriteLine(st.ToLongTimeString());

}

public TimeSpan TimeR(int seconds)

{

TimeSpan interval = TimeSpan.FromSeconds(seconds);

return sec - interval;

}

public void TimeSravn()

{

if (SecInt() > SecStr())

Console.WriteLine("Время из величин целого типа больше времени строкового представления.");

else if (SecInt() < SecStr())

Console.WriteLine("Время из величин целого типа меньше времени строкового представления.");

else

Console.WriteLine("Время из величин целого типа равно времени строкового представления.");

}

public void SecTranslate()

{

Console.WriteLine("Перевод времени в секунды: ");

Console.WriteLine("Целый тип: " + SecInt());

Console.WriteLine("Строковый тип: " + SecStr());

Console.WriteLine("Тип даты: " + SecDT());

}

protected int MinInt()

{

if (s >= 30)

return h \* 60 + m + 1;

else

return h \* 60 + m;

}

protected int MinStr()

{

if (int.Parse(hms.Substring(6, 2)) >= 30)

return int.Parse(hms.Substring(0, 2)) \* 60 + int.Parse(hms.Substring(3, 2)) + 1;

else

return int.Parse(hms.Substring(0, 2)) \* 60 + int.Parse(hms.Substring(3, 2));

}

protected double MinDT()

{

DateTime dt1970 = new DateTime(2017, 12, 23, 0, 0, 0);

TimeSpan tsInterval = st.Subtract(dt1970);

return Math.Round(tsInterval.TotalMinutes);

}

public void MinTranslate()

{

Console.WriteLine("Перевод времени в минуты: ");

Console.WriteLine("Целый тип: " + MinInt());

Console.WriteLine("Строковый тип: " + MinStr());

Console.WriteLine("Тип даты: " + MinDT());

}

}

/\* Создать класс Time для работы со временем в формате «час:минута:секунда».

\* Класс должен включать в себя не менее четырех функций инициализации: числами, строкой (например, «23:59:59»), секундами и временем.

\* Обязательными являются операции:

• вычисления разницы между двумя моментами времени в секундах;

• сложение времени и заданного количества секунд;

• вычитание из времени заданного количества секунд;

• сравнение моментов времени;

• перевод в секунды;

• перевод в минуты (с округлением до целой минуты).

\*/

class Program

{

static void Main(string[] args)

{

DateTime dt = new DateTime(2017, 12, 23, 23, 30, 25);

Time t = new Time(12, 12, 12, "04:50:49", dt, TimeSpan.FromSeconds(69));

t.Info();

Console.WriteLine("Вычитание из времени целого типа время строкового типа: " + t.SecR());

Console.WriteLine("Сложение времени из соответсвующего типа с типом временного интервала секунд: ");

t.SecS();

Console.Write("Задайте количество секунд для вычета: ");

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

Console.WriteLine(t.TimeR(sec));

t.TimeSravn();

t.SecTranslate();

t.MinTranslate();

}

}

}