using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class ArrayOperation

{

public int[] getChangedArray(int[] inArray)

{

int[] temp = new int[inArray.Length];

Array.Copy(inArray, temp, temp.Length);

int index\_1 = Array.FindIndex(temp, i => (i < 0));

int index\_2 = Array.FindLastIndex(temp, i => (i < 0));

if ((index\_1 == temp.Length - 1) | (index\_1 == index\_2) | (index\_2 == 0))

{

throw new InvalidOperationException("Операция невозможна для данного аргумента!");

}

else

{

int n = temp[0];

temp[0] = temp[Array.FindIndex(temp, index\_1 + 1, l => (l < 0))];

temp[Array.FindIndex(temp, index\_1 + 1, f => (f < 0))] = n;

}

return temp;

}

}

class Program

{

static void Main(string[] args)

{

int[] arr\_C = { 2, 5, 7, -3, 5, -1 };

int[] arr\_D = { 3, 65, 78, 23, -54, 23, 13, 41, -9, 32 };

ArrayOperation operation = new ArrayOperation();

try

{

int[] changedArray\_C = operation.getChangedArray(arr\_C);

int[] changedArray\_D = operation.getChangedArray(arr\_D);

Console.WriteLine("First Array:");

foreach (int i in arr\_C)

{

Console.Write("{0} ", i);

}

Console.WriteLine("\nChanged First Array:");

foreach (int i in changedArray\_C)

{

Console.Write("{0} ", i);

}

Console.WriteLine();

Console.WriteLine("Second Array");

foreach (int i in arr\_D)

{

Console.Write("{0} ", i);

}

Console.WriteLine("\nChanged Second Array:");

foreach (int i in changedArray\_D)

{

Console.Write("{0} ", i);

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Console.ReadKey();

}

}

}

\*\*\*\*\*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp9\_

{

struct elemDate

{

public int days;

public int months;

public int years;

public elemDate(int days, int months, int years)

{

this.days = days;

this.months = months;

this.years = years;

}

}

class Date

{

public int day;

public int month;

public int year;

public string dmy;

public DateTime dmyt;

public elemDate[] date = new elemDate[3];

public Date(int day, int month, int year, string dmy, DateTime dmyt)

{

this.day = day;

this.month = month;

this.year = year;

this.dmy = dmy;

this.dmyt = dmyt;

}

public int Preobr1()

{

string[] words = dmy.Split('.');

day = int.Parse(words[0]);

month = int.Parse(words[1]);

year = int.Parse(words[2]);

return (0);

}

public int Preobr2()

{

string g = dmyt.ToString();

dmy = g.Substring(0, 10);

Preobr1();

return (0);

}

public int Ini()

{

date[0] = new elemDate(day, month, year);

Preobr1();

date[1] = new elemDate(day, month, year);

Preobr2();

date[2] = new elemDate(day, month, year);

/\*

ConvertDateTimeToInt32(dmyt);

int error = ConvertDateTimeToInt32(DateTime.MaxValue);

Console.WriteLine(error);

\*/

return (0);

}

public void Vyvod()

{

Console.WriteLine("Дата:");

foreach (elemDate a in date)

{

Console.WriteLine("День: {0}", a.days);

Console.WriteLine("Месяц: {0}", a.months);

Console.WriteLine("Год: {0}", a.years);

Console.WriteLine();

}

}

/\* Создать класс Date для работы с датами в формате «день.месяц.год».

\* Дата представляется структурой с тремя полями типа unsigned int:

\* для года, месяца, дня. Класс должен включать не менее трех функций инициализации:

\* числами, строкой вида «день.месяц.год» и датой. Обязательными операциями являются:

\*

· вычисление даты через заданное количество дней;

· вычитание заданного количества дней из даты;

· определение високосности года;

· присвоение и получение отдельных частей (день, месяц, год);

· сравнение дат (равно, до, после);

· вычисление количества дней между датами.

\*/

public int plus()

{

Console.Write("Введите количество дней: ");

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

date[0].days += dayp;

if (date[0].days > 31)

{

int cp = 1 + dayp / 30;

date[0].months += cp;

date[0].days -= cp \* 30;

if (date[0].months > 12)

{

int cpc = date[0].months / 12;

date[0].years += cpc;

date[0].months -= cpc \* 12;

}

}

return (0);

}

public int minus()

{

Console.Write("Введите количество дней: ");

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

date[1].days -= dayp;

if (date[1].days < 1)

{

int cp = 1 + dayp / 30;

date[1].months -= cp;

date[1].days += cp \* 30;

if (date[1].months < 1)

{

int cpc = 1 + date[1].months / 12;

date[1].years -= cpc;

date[1].months += cpc \* 12;

}

}

return (0);

}

public void visokos()

{

if (date[2].years % 4 == 0 && (date[2].years % 100 != 0 || date[2].years % 400 == 0))

Console.WriteLine("Високосный год");

else

Console.WriteLine("Не високосный год");

}

public void ravn()

{

if (date[0].days == date[1].days && date[0].months == date[1].months && date[0].years == date[1].years)

Console.WriteLine("Даты равны");

else if (date[0].days < date[1].days && date[0].months == date[1].months && date[0].years == date[1].years)

Console.WriteLine("Первая дата меньше");

else if (date[0].months < date[1].months && date[0].years == date[1].years)

Console.WriteLine("Первая дата меньше");

else if (date[0].years < date[1].years)

Console.WriteLine("Первая дата меньше");

else if (date[0].days > date[1].days && date[0].months == date[1].months && date[0].years == date[1].years)

Console.WriteLine("Вторая дата меньше");

else if (date[0].months > date[1].months && date[0].years == date[1].years)

Console.WriteLine("Вторая дата меньше");

else if (date[0].years > date[1].years)

Console.WriteLine("Вторая дата меньше");

}

public void sravn()

{

Console.WriteLine("Между датами {0} дней", Math.Abs((date[0].years \* 365 + date[0].months \* 30 + date[0].days) - (date[1].years \* 365 + date[1].months \* 30 + date[1].days)));

}

}

class Program

{

static void Main(string[] args)

{

DateTime dt = new DateTime(2012, 12, 12);

Date d = new Date(17, 12, 2017, "01.01.2000", dt);

d.Ini();

d.Vyvod();

d.plus();

d.Vyvod();

d.minus();

d.Vyvod();

d.visokos();

d.ravn();

d.sravn();

}

}

}