using System;

using System.CodeDom;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Security.Cryptography;

using System.Text;

using System.Threading.Tasks;

using System.Xml.Linq;

using System.Xml.Schema;

//1

/\*class Topic

{

private string subject;

private int number;

private string name;

private double difficulty;

private int points;

public string Subject

{

get { return subject; }

set { subject = value; }

}

public int Number

{

get { return number; }

set { number = value; }

}

public string Name

{

get { return name; }

set { name = value; }

}

public double Difficulty

{

get { return difficulty; }

set

{

if (value >= 1)

{

Console.WriteLine("коефициент складности бильше чи равно 1 тому буде виставлено найблильше число вказаного диапозона");

difficulty = 0.99;

}

else

difficulty = value;

}

}

public int Points

{

get

{

return points;

}

set

{

if (value < 1 || value > 100)

{

if (value > 100)

{

Console.WriteLine("коефициент складности бильше 100 тому буде виставлено найблильше число вказаного диапозона");

points = 100;

}

else

{

Console.WriteLine("коефициент складности меньше 1 тому буде виставлено найменьше число вказаного диапозона");

points = 1;

}

}

else

points = value;

}

}

public int Rating

{

get

{

return (int)Math.Round((double)points\*difficulty);

}

}

public void Print()

{

Console.WriteLine("{0}, {1}, {2}, {3}, {4}", Subject, Number, Name, Difficulty, Points);

}

public Topic(string subject, int number, string name, double difficulty, int points)

{

this.subject = subject;

this.number = number;

this.name = name;

if(difficulty >= 1)

{

Console.WriteLine("коефициент складности бильше чи равно 1 тому буде виставлено найблильше число вказаного диапозона");

this.difficulty = 0.99;

}

else

this.difficulty = difficulty;

if (points<1||points>100)

{

if (points > 100)

{

Console.WriteLine("коефициент складности бильше 100 тому буде виставлено найблильше число вказаного диапозона");

this.points = 100;

}

else

{

Console.WriteLine("коефициент складности меньше 1 тому буде виставлено найменьше число вказаного диапозона");

this.points = 1;

}

}

else

this.points = points;

}

}\*/

//2

/\*enum Countries

{

Usa =1,

Ukraine = 380,

Australian = 61,

Mexico = 52

}

struct Subscriber

{

string name;

int phonenumber;

string address;

Countries country;

public Subscriber(string name, int phonenumber, string address, int country)

{

this.name = name;

this.phonenumber = phonenumber;

this.address = address;

this.country = (Countries)country;

}

public void Input()

{

string tempcountries;

string[] cget=System.Enum.GetNames(typeof(Countries));

Console.WriteLine("Name:");

name = Console.ReadLine();

Console.WriteLine("phoneNum:");

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

Console.WriteLine("Adress:");

address = Console.ReadLine();

Console.WriteLine("Countries(string):");

tempcountries = Console.ReadLine();

int index = 0 ;

bool flag=false;

foreach(string c in cget)

{

if (tempcountries == c)

{

flag = true;

break;

}

index++;

}

if (flag)

{

switch (index)

{

case 0:

this.country = Countries.Usa;

break;

case 1:

this.country = Countries.Mexico;

break;

case 2:

this.country = Countries.Australian;

break;

case 3:

this.country = Countries.Ukraine;

break;

default:

this.country = Countries.Usa;

break;

}

}

else

this.country = Countries.Usa;

}

public void Print()

{

Console.WriteLine("name:{0},phonenumber:{1},adress{2},countryCode:{3}",name,phonenumber,address,(int)country);

}

public string Nmae

{

set { this.name = value; }

get { return name; }

}

public int Phonenumber

{

set { this.phonenumber = value; }

get { return phonenumber; }

}

public string Address

{

set { this.address = value; }

get { return address; }

}

public int Country

{

set

{

int[] val = (int[])System.Enum.GetValues(typeof(Countries));

bool flag = true;

foreach (int val2 in val)

{

if (value == val2)

{

flag = false;

country = (Countries)value;

break;

}

}

if (flag)

country = Countries.Usa;

}

get { return (int)country; }

}

}\*/

//3

/\*class Point

{

double x, y;

public Point(double x, double y)

{

this.x = x;

this.y = y;

}

public Point()

{

x = 0;

y = 0;

}

public double X

{

get { return x; }

private set { x = value; }

}

public double Y

{

get { return y; }

private set { y = value; }

}

public static Point operator +(Point obj, Point obj2)

{

Point obj3 = new Point(obj.x+obj2.x,obj.y+obj2.y);

return obj3;

}

public static Point operator -(Point obj, Point obj2)

{

Point obj3 = new Point(obj.x - obj2.x, obj.y - obj2.y);

return obj3;

}

public static Point operator - (Point obj)

{

return new Point(obj.X\*-1.0,obj.Y\*-1.0);

}

public static bool operator true(Point obj)

{

if (obj.X == 0 && obj.Y == 0)

return true;

else

return false;

}

public static bool operator false(Point obj)

{

if (obj.X == 0 && obj.Y == 0)

return false;

else

return true;

}

public static bool operator ==(Point obj,Point obj2)

{

if (obj.X == obj2.X && obj.Y == obj2.Y)

return true;

else

return false;

}

public static bool operator !=(Point obj, Point obj2)

{

if (obj.X == obj2.X && obj.Y == obj2.Y)

return false;

else

return true;

}

public void Print()

{

Console.WriteLine("x:{0},y:{1}", x, y);

}

}\*/

//4 28.09.23

class Human

{

protected string surname;

protected string name;

protected DateTime birthDate;

public void Info()

{

Console.WriteLine("surname:{0},name:{1},day:{2},month:{3},year:{4}",surname,name,birthDate.Day,birthDate.Month,birthDate.Year);

}

public string Surname

{

set { surname = value; }

get { return surname; }

}

public string Name

{

set { name = value; }

get { return name; }

}

public DateTime BirthDate

{

get { return birthDate; }

set { birthDate = value; }

}

public int Day

{

get { return birthDate.Day;}

set { birthDate = new DateTime(birthDate.Year,birthDate.Month,value); }

}

public int Month

{

get { return birthDate.Month;}

set { birthDate = new DateTime(birthDate.Year, value, birthDate.Day); }

}

public int Year

{

get { return birthDate.Year; }

set { birthDate = new DateTime(value, birthDate.Month, birthDate.Day); }

}

public Human(string surname,string name,int day,int month,int year)

{

this.surname = surname;

this.name = name;

birthDate = new DateTime(year,month,day);

}

}

class Citizen : Human

{

protected string passportNumber;

protected string address;

public string PassportNumber

{

set { passportNumber = value; }

get { return passportNumber; }

}

public string Address

{

set { address = value; }

get { return address; }

}

public Citizen(string surname, string name, int day, int month, int year,string passportNumber,string adress):base(surname,name,day,month,year)

{

this.passportNumber = passportNumber;

this.address = adress;

}

public new void Info()

{

base.Info();

Console.WriteLine("passportNumber:"+passportNumber+" Adress:"+address);

}

}

class ForeignCitizen : Citizen

{

string foreignPassportNumber;

public new void Info()

{

base.Info();

Console.WriteLine("foreignPassportNumber:{0}",foreignPassportNumber);

}

public string ForeignPassportNumber

{

set { foreignPassportNumber = value; }

get { return foreignPassportNumber; }

}

public ForeignCitizen(string surname, string name, int day, int month, int year, string passportNumber, string adress, string foreignPassportNumber) : base(surname, name, day, month, year, passportNumber, adress)

{

this.foreignPassportNumber = foreignPassportNumber;

}

}

namespace Homework19.\_09.\_23

{

internal class Program

{

static void Main(string[] args)

{

//1

/\*Topic topic = new Topic("Math", 1, "Algebra", 2, 0);

topic.Print();

topic.Points = 120;

topic.Difficulty = 1.5;

Console.WriteLine();

topic.Print();\*/

//2

/\*Subscriber subscriber = new Subscriber("Doter", 0986665666,"HELL",380);

subscriber.Print();

subscriber.Input();

subscriber.Print();

subscriber.Country = 380;

subscriber.Print();\*/

//3

/\*Point obj = new Point(10, 2);

Point obj2 = new Point(10, 1);

Point obj3 = obj - obj2;

obj3.Print();

obj3 = -obj3;

obj3.Print();

obj3 = -obj3;

obj3.Print();

if (obj3)

{

Console.WriteLine("да кординаты 0 0");

}

else

{

Console.WriteLine("нет кординаты не 0.0");

}

if (obj3 != obj2)

{

Console.WriteLine("объекты не равны");

}\*/

//4 28.09.23

}

}

}