МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ БІЗНЕС-КОЛЕДЖ

***Циклова комісія програмування***

**Робочий план**

з курсу «Об’єктно-орієнтоване програмування»

Солом’яного Ярослава Сергійовича

*ПІБ студента*

студента групи **2П-18**

|  |  |  |
| --- | --- | --- |
| **Вид роботи** | **Дата** | **Оцінка** |
| *Пром. контроль* |  |  |
| *Залік* |  |  |

Викладач Марченко С. В.

Черкаси-2020

Завдання №1. (Обмін валют) Напишіть програму, яка буде приймати на вхід суму грошей у гривнях та обчислюватиме відповідну суму купівлі та продажу в інших валютах на момент запуску програми. Курси валют візьміть за посиланням.

Лістинг 1.Обмін валют

using System;

using System.Text;

using System.Globalization;

namespace Новая\_папка

{

class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.UTF8;

Console.Write("Веддіть сумму(грн):");

int summ = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Ви ввели сумму(грн): " + summ);

Console.WriteLine("Курс купівлі $:27.9500, продажі:28.3500");

Console.WriteLine("Курс купівлі €:32.5000, продажі:33.1200");

Console.WriteLine("Курс купівлі ₽:0.3550, продажі:0.3850");

Console.WriteLine("Курс купівлі zł:6.9500, продажі:7.5000 ");

Console.WriteLine("Сумма купівлі та продажу:");

double dollarbuy = summ / 27.9500;

double dollarsell = summ / 28.3500;

double eurobuy = summ / 32.5000;

double eurosell = summ / 33.1200;

double rubbuy = summ / 0.3550;

double rubsell = summ / 0.3850;

double zlotiybuy = summ / 6.9500;

double zlotiysell = summ / 7.5000;

Console.WriteLine($"Сумма купівлі $: {dollarbuy}, Сумма продажі $: {dollarsell}");

Console.WriteLine($"Сумма купівлі €: {eurobuy}, Сумма продажі €: {eurosell}");

Console.WriteLine($"Сумма купівлі ₽: {rubbuy}, Сумма продажі ₽: {rubbuy}");

Console.WriteLine($"Сумма купівлі zł: {zlotiybuy}, Сумма продажі zł: {zlotiysell}");

PrintTimeStamp("Ярослав Солом'яний, студент групи 2П18");

}

public static void PrintTimeStamp(string name)

{

Console.OutputEncoding = Encoding.UTF8;

DateTime localDate = DateTime.Now;

var culture = new CultureInfo("ru-RU");

Console.WriteLine("Дата та час компіляції: {0}", localDate.ToString(culture));

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

}

}

}

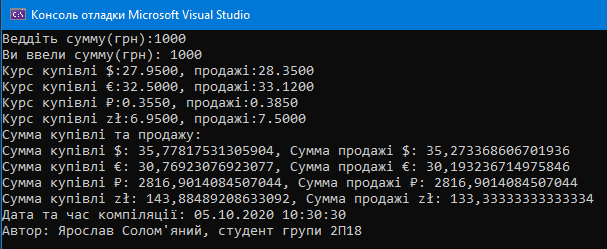


Рис 1.1 Результат виконання завдання №1

Завдання №2. Закон ідеального газу – математичне наближення поведінки газів при зміні тиску, об’єму та температури. Зазвичай він описується рівнянням 𝑃𝑉 = 𝑛𝑅𝑇, де 𝑃 – тиск у паскалях (Па), 𝑉 – об’єм, 𝑛 – кількість молів речовини, 𝑅 = 8.314 Дж моль∙𝐾 – універсальна газова стала, 𝑇 – температура в градусах Кельвіна. Напишіть програму, яка обчислює молярну масу газу, коли користувач задає тиск, об’єм і температуру. Протестуйте свою програму на визначення кількості молей газу в балоні акваланга. Типовий балон містить 12 літрів газу під тиском 20,000,000 Па. Беремо кімнатну температуру (20ºС).

Лістинг 2.Визначення молярної маси.

using System;

using System.Text;

using System.Globalization;

namespace Новая\_папка

{

class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.UTF8;

Console.WriteLine("Введіть Тиск(Па): ");

double p = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Введіть Об'єм(л): ");

double v = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Введіть температуру(℃ ): ");

double t = Convert.ToDouble(Console.ReadLine());

double tempery = t + 273.15;

double R = 8.314;

double Ob = v / 1000;

double n = (p \* Ob) / (R \* tempery);

Console.WriteLine("Молярна маса становить: " + n);

PrintTimeStamp("Ярослав Солом'яний, студент групи 2П18");

}

public static void PrintTimeStamp(string name)

{

Console.OutputEncoding = Encoding.UTF8;

DateTime localDate = DateTime.Now;

var culture = new CultureInfo("ru-RU");

Console.WriteLine("Дата та час компіляції: {0}", localDate.ToString(culture));

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

}

}

}

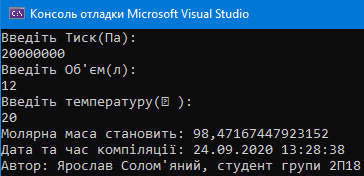


Рис 1.2 Виконання програми №2

3. Створіть консольний проект .NET Framework та напишіть клас, який знаходить площу чотирикутника, утвореного чотирма містами України: Києвом, Львовом, Одесою, Харковом.

Програма має отримувати GPS координати (довготу та широту) міст, знаходити відстані між ними за формулою

𝑑 = 𝑅 ∙ arccos(sin 𝑥1 sin 𝑥2 + cos 𝑥1 cos 𝑥2 cos(𝑦1 − 𝑦2 )),

де 𝑅 – радіус Землі (в середньому 6371,01км), а потім обчислювати площу чотирикутника як суму площ двох трикутників (кожна за формулою Герона), отриманих з’єднанням двох несусідніх міст.

Рекомендується окремо виділити метод для знаходження відстані між двома населеними пунктами, а також виокремити метод для обчислення площі прямокутника.

Портуйте даний проект на .NET Core, як описано, наприклад, за посиланнями https://docs.microsoft.com/en-us/dotnet/core/porting/, https://www.cafeencounter.net/p2380/migrating-net-framework-to-netcore та порівняйте структуру проектів, розміри виконуваних файлів тощо.

Лістинг 3.Знаходження площі прямокутника між містами

using System;

using System.Text;

using System.Globalization;

using System.Security.Cryptography;

namespace Новая\_папка

{

class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.UTF8;

Console.Write("Введіть довготу м.Київ ==>> ");

double longitude\_kiev = Convert.ToDouble(Console.ReadLine());

Console.Write("Введіть широту м.Київ ==>> ");

double latitude\_kiev = Convert.ToDouble(Console.ReadLine());

Console.Write("Введіть довготу м.Львів ==>> ");

double longitude\_lviv = Convert.ToDouble(Console.ReadLine());

Console.Write("Введіть широту м.Львів ==>> ");

double latitude\_lviv = Convert.ToDouble(Console.ReadLine());

Console.Write("Введіть довготу м.Харків ==>> ");

double longitude\_xarkiv = Convert.ToDouble(Console.ReadLine());

Console.Write("Введіть широту м.Харків ==>> ");

double latitude\_xarkiv = Convert.ToDouble(Console.ReadLine());

Console.Write("Введіть довготу м.Одеса ==>> ");

double longitude\_odessa = Convert.ToDouble(Console.ReadLine());

Console.Write("Введіть широту м.Одеса ==>> ");

double latitude\_odessa = Convert.ToDouble(Console.ReadLine());

Console.WriteLine($"Координати для м.Київ ==>> Довгота = {longitude\_kiev} // Широта = {latitude\_kiev}");

Console.WriteLine($"Координати для м.Львів ==>> Довгота = {longitude\_lviv} // Широта = {latitude\_lviv}");

Console.WriteLine($"Координати для м.Харків ==>> Довгота = {longitude\_xarkiv} // Широта = {latitude\_xarkiv}");

Console.WriteLine($"Координати для м.Одеса ==>> Довгота = {longitude\_odessa} // Широта = {latitude\_odessa}");

double radian\_longitude\_kiev, radian\_latitude\_kiev;

double radian\_longitude\_lviv, radian\_latitude\_lviv;

double radian\_longitude\_xarkiv, radian\_latitude\_xarkiv;

double radian\_longitude\_odessa, radian\_latitude\_odessa;

radian\_longitude\_kiev = longitude\_kiev \* Math.PI / 180;

radian\_latitude\_kiev = latitude\_kiev \* Math.PI/180;

radian\_longitude\_lviv = longitude\_lviv \* Math.PI / 180;

radian\_latitude\_lviv = latitude\_lviv \* Math.PI / 180;

radian\_longitude\_xarkiv = longitude\_xarkiv \* Math.PI / 180;

radian\_latitude\_xarkiv = latitude\_xarkiv \* Math.PI / 180;

radian\_longitude\_odessa = longitude\_odessa \* Math.PI / 180;

radian\_latitude\_odessa = latitude\_odessa \* Math.PI / 180;

double r = 6371.01;

double d\_kiev\_xarkiv,d\_kiev\_lviv,d\_xarkiv\_lviv;

d\_kiev\_xarkiv = r \* Math.Acos(Math.Sin(radian\_longitude\_kiev) \* Math.Sin(radian\_longitude\_xarkiv) + Math.Cos(radian\_longitude\_kiev) \* Math.Cos(radian\_longitude\_xarkiv) \* Math.Cos(radian\_latitude\_kiev - radian\_latitude\_xarkiv));

d\_kiev\_lviv = r \* Math.Acos(Math.Sin(radian\_longitude\_kiev) \* Math.Sin(radian\_longitude\_lviv) + Math.Cos(radian\_longitude\_kiev) \* Math.Cos(radian\_longitude\_lviv) \* Math.Cos(radian\_latitude\_kiev - radian\_latitude\_lviv));

d\_xarkiv\_lviv = r \* Math.Acos(Math.Sin(radian\_longitude\_xarkiv) \* Math.Sin(radian\_longitude\_lviv) + Math.Cos(radian\_longitude\_xarkiv) \* Math.Cos(radian\_longitude\_lviv) \* Math.Cos(radian\_latitude\_xarkiv - radian\_latitude\_lviv));

double d\_odessa\_xarkiv, d\_odessa\_lviv;

d\_odessa\_xarkiv = r \* Math.Acos(Math.Sin(radian\_longitude\_odessa) \* Math.Sin(radian\_longitude\_xarkiv) + Math.Cos(radian\_longitude\_odessa) \* Math.Cos(radian\_longitude\_xarkiv) \* Math.Cos(radian\_latitude\_odessa - radian\_latitude\_xarkiv));

d\_odessa\_lviv = r \* Math.Acos(Math.Sin(radian\_longitude\_odessa) \* Math.Sin(radian\_longitude\_lviv) + Math.Cos(radian\_longitude\_odessa) \* Math.Cos(radian\_longitude\_lviv) \* Math.Cos(radian\_latitude\_odessa - radian\_latitude\_lviv));

Console.WriteLine($"Відстань між Києвом та Харьковом ==>>> {d\_kiev\_xarkiv} Км.");

Console.WriteLine($"Відстань між Києвом та Львовом ==>>> {d\_kiev\_lviv} Км.");

Console.WriteLine($"Відстань між Харьковом та Львовом ==>>> {d\_xarkiv\_lviv} Км.");

Console.WriteLine($"Відстань між Одесою та Харьковом ==>>> {d\_odessa\_xarkiv} Км.");

Console.WriteLine($"Відстань між Одесою та Львовом ==>>> {d\_odessa\_lviv} Км.");

double geron1, geron2, p1, p2;

p1 = (d\_kiev\_xarkiv + d\_kiev\_lviv + d\_xarkiv\_lviv) / 2;

geron1 = Math.Sqrt(p1 \* (p1 - d\_kiev\_xarkiv) \* (p1 - d\_kiev\_lviv) \* (p1 - d\_xarkiv\_lviv));

p2 = (d\_odessa\_xarkiv + d\_odessa\_lviv + d\_xarkiv\_lviv) / 2;

geron2 = Math.Sqrt(p2 \* (p2 - d\_odessa\_xarkiv) \* (p2 - d\_odessa\_lviv) \* (p2 - d\_xarkiv\_lviv));

double summ\_geron;

summ\_geron = geron1 + geron2;

Console.WriteLine($"Площа трикутника утвореного з міст (Київ, Харків, Львів) ====>>>>> {geron1}");

Console.WriteLine($"Площа трикутника утвореного з міст (Одеса, Харків, Львів) ====>>>>> {geron2}");

Console.WriteLine($"Площа прямокутника утвореного з усіх міст ====>>>>> {summ\_geron}");

PrintTimeStamp("Ярослав Солом'яний, студент групи 2П-18");

}

public static void PrintTimeStamp(string name)

{

Console.OutputEncoding = Encoding.UTF8;

DateTime localDate = DateTime.Now;

var culture = new CultureInfo("ru-RU");

Console.WriteLine("Дата та час компіляції: {0}", localDate.ToString(culture));

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

}

}

}

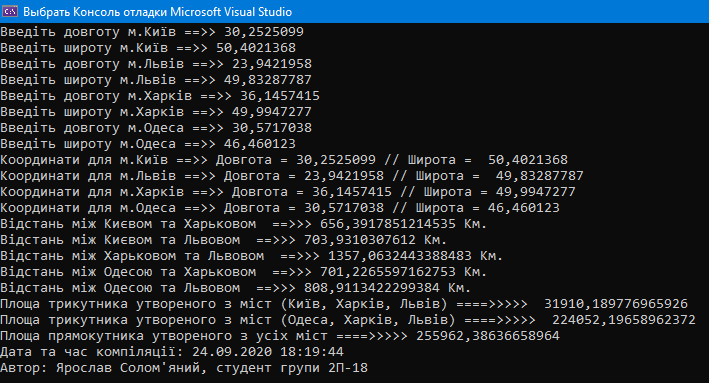


Рис 3.Результат завдання №3.

4. (Варіант 1)Уявімо, що Ви тільки-но відкрили новий накопичувальний рахунок зі ставкою 14% річних. Отриманий прибуток виплачується в кінці року та додається до балансу рахунку. Напишіть програму, що починає роботу зі зчитування кількості покладених на рахунок грошей. Далі вона має обчислити та вивести на екран суму на рахунку через 1, 2 і 3 роки. Показ сум обмежується 2 знаками після крапки.

Лістинг 4.Визначення суми через 1,2,3

роки на накопичувальному рахунку.

using System;

using System.Text;

using System.Globalization;

namespace Завдання\_3

{

class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.UTF8;

Console.Write("Введіть суму, яку хочете покласти під % ==>> ");

double Summ;

Summ = Convert.ToDouble(Console.ReadLine());

int t1, t2, t3;

double w;

w = 0.14;

t1 = 1; t2 = 2; t3 = 3;

double s;

s = Math.Round(Summ\*(Math.Pow(1 + w, t1)),2);

Console.WriteLine($"Кінцева сумма через рік ==>> {s} грн.") ;

s = Math.Round(Summ \* (Math.Pow(1 + w, t2)), 4);

Console.WriteLine($"Кінцева сумма через два роки ==>> {s} грн.");

s = Math.Round(Summ \* (Math.Pow(1 + w, t3)), 2);

Console.WriteLine($"Кінцева сумма через три роки ==>> {s} грн.");

PrintTimeStamp("Ярослав Солом'яний, студент групи 2П-18");

}

public static void PrintTimeStamp(string name)

{

Console.OutputEncoding = Encoding.UTF8;

DateTime localDate = DateTime.Now;

var culture = new CultureInfo("ru-RU");

Console.WriteLine("Дата та час компіляції: {0}", localDate.ToString(culture));

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

}

}

}

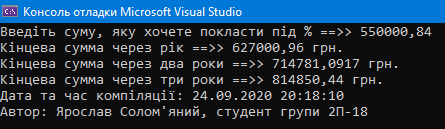


Рис 4.Результат роботи програми №4.

5. Напишіть програму, яка буде виводити, скільки днів пройшло від Вашого дня народження до поточного часу.

Лістинг 5.Визначення часу з дня народження

using System;

using System.Text;

using System.Globalization;

namespace Завдання\_5

{

class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.UTF8;

DateTime date1 = new DateTime(2003, 5, 1);

Console.Write("Днів з мого дня народження: ");

Console.WriteLine(DateTime.Now.Subtract(date1));

PrintTimeStamp("Ярослав Солом'яний, студент групи 2П-18");

}

public static void PrintTimeStamp(string name)

{

Console.OutputEncoding = Encoding.UTF8;

DateTime localDate = DateTime.Now;

var culture = new CultureInfo("ru-RU");

Console.WriteLine("Дата та час компіляції: {0}", localDate.ToString(culture));

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

}

}

}

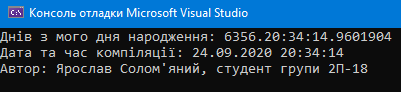


Рис 5.Результат роботи програми №5.

Завдання 6. Для заданого RGB-кольору, представленого у вигляді шістнадцяткового числа, яке вводить користувач, визначте окремі значення для червоного, зеленого та синього каналів. Для цього слід використати побітові оператори.

Лістинг 6.Знаходження RGB для кольора за HEX значенням

using System;

using System.Text;

using System.Globalization;

namespace ConsoleApp2

{

class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.UTF8;

Console.WriteLine("Введіть число у вигляді шістнадцяткового:");

uint color = Convert.ToUInt32(Console.ReadLine(), 16);//Зчитуємо шістнадцяткове число

var red = (color & 0xFF0000) >> 16;//виділяє з цього числа три компонента кольору для пікселів з глибиною кольору 16 біт.

var green = (color & 0x00FF00) >> 8;

var blue = color & 0x0000FF;

Console.WriteLine($"RGB [{red},{green},{blue}]");

PrintTimeStamp("Ярослав Солом'яний, студент групи 2П-18");

}

public static void PrintTimeStamp(string name)

{

Console.OutputEncoding = Encoding.UTF8;

DateTime localDate = DateTime.Now;

var culture = new CultureInfo("ru-RU");

Console.WriteLine("Дата та час компіляції: {0}", localDate.ToString(culture));

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

}

}

}

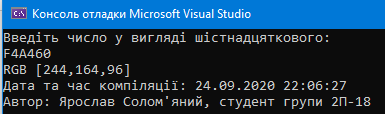


Рис 6. Результат роботи програми №6

7. Напишіть програму, яка перевірятиме, чи дві клітинки шахової дошки однакового кольору, не застосовуючи оператори галуження чи циклу. Відповідь повинна бути булевою змінною, значення якої true означає однаковість кольору, а false – відмінність. Клітинка представляється координатою з двох цілочисельних значень від 1 до 8.

Лістинг 7.Перевірка клітинок шахової дошки

using System;

using System.Text;

using System.Globalization;

namespace ConsoleApp3

{

class Program

{

enum Letters { a, b, c, d, e, f, g, h }

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.UTF8;

Console.WriteLine( ((int)Letters.c + 3) % 2 == ((int)Letters.g + 6) % 2);

PrintTimeStamp("Ярослав Солом'яний, студент групи 2П-18");

}

public static void PrintTimeStamp(string name)

{

Console.OutputEncoding = Encoding.UTF8;

DateTime localDate = DateTime.Now;

var culture = new CultureInfo("ru-RU");

Console.WriteLine("Дата та час компіляції: {0}", localDate.ToString(culture));

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

}

}

}

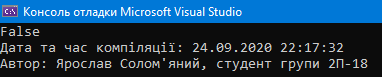


Рис 7. Розв’язок задачі №7

Завдання №8. (Системні типи) Виведіть таблицю з інформацією щодо значимих системних типів мови програмування C#, використовуючи програмні можливості мови (sizeof(), typeof() та ін.). Приклад подібного виводу:

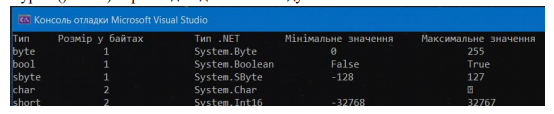


Рис 8. Приклад виводу розв’язку

Лістинг 8.Вивід результатів про різні типи данних

using System;

using System.Text;

using System.Globalization;

using System.ComponentModel.DataAnnotations;

namespace ConsoleApp4

{

class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.UTF8;

Console.WriteLine($"Тип \t Розмір у байтах \t Тип .NET \t\t Мінімальне значення \t\t\t Максимальне значення ") ;

Console.WriteLine($"Byte \t\t {sizeof(byte)} \t\t{typeof(byte)}\t\t\t {byte.MinValue} \t\t\t\t\t {byte.MaxValue}");

Console.WriteLine($"Bool \t\t {sizeof(bool)} \t\t{typeof(bool)}\t\t\t False \t\t\t\t\t True");

Console.WriteLine($"Sbyte \t\t {sizeof(sbyte)} \t\t{typeof(sbyte)}\t\t\t {sbyte.MinValue} \t\t\t\t\t {sbyte.MaxValue}");

Console.WriteLine($"Char \t\t {sizeof(Char)} \t\t{typeof(Char)}\t\t\t {Char.MinValue} \t\t\t\t\t {Char.MaxValue}");

Console.WriteLine($"Short \t\t {sizeof(short)} \t\t{typeof(short)}\t\t\t {short.MinValue} \t\t\t\t {short.MaxValue}");

Console.WriteLine($"Int \t\t {sizeof(int)} \t\t{typeof(int)}\t\t\t {int.MinValue} \t\t\t\t {int.MaxValue}");

Console.WriteLine($"Float \t\t {sizeof(float)} \t\t{typeof(float)}\t\t\t {float.MinValue} \t\t\t {float.MaxValue}");

Console.WriteLine($"Ulong \t\t {sizeof(ulong)} \t\t{typeof(ulong)}\t\t\t\t {ulong.MinValue} \t\t\t\t {ulong.MaxValue}");

Console.WriteLine($"Decimal \t{sizeof(decimal)} \t\t{typeof(decimal)}\t\t {decimal.MinValue} \t {decimal.MaxValue}\n");

PrintTimeStamp("Ярослав Солом'яний, студент групи 2П18");

}

public static void PrintTimeStamp(string name)

{

Console.OutputEncoding = Encoding.UTF8;

DateTime localDate = DateTime.Now;

var culture = new CultureInfo("ru-RU");

Console.WriteLine("Дата та час компіляції: {0}", localDate.ToString(culture));

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

}

}

}

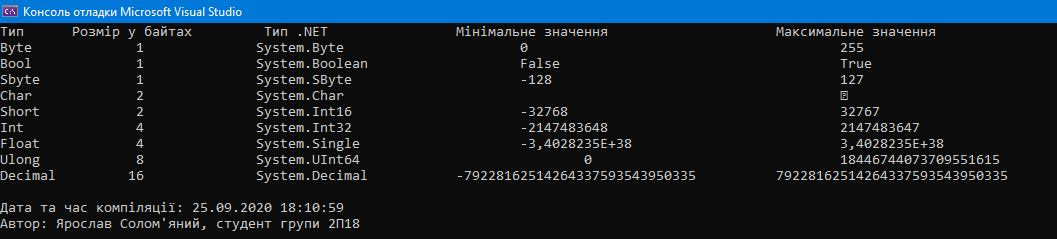


Рис 9.Вивід програми про різні типи данних