**ЗВІТ**

****Про виконання лабораторної роботи №4

Назва роботи: Створення класів в C#. Поліморфізм.

***Виконав*** : Беліч Олександр

***Група,факультет*** : К-21, ІТЕ

***Лектор*** : Артеменко О.І.

**Мета:** Вивчити особливості роботи з власними класами у C#.

**GIT:** *<https://github.com/Alxich/Labs_Csh_lang/tree/main/Lab4>*

Варіант 4

**(2 бал)**

***1.Розробити консольний застосунок для роботи з класом згідно свого варіанту. Створити не менше 10 записів для виконання операцій, вказаних у завданні. Для ідентифікації спроби введення з клавіатури некоректних даних описати виключення.***

**Завдання:** Описати клас для бази зданих з інформацією про конфігурацію комп’ютера з полями: тип процесора, тактова частота, обсяг оперативної пам’яті, обсяг дискової пам’яті, характеристика монітора та ін.Відсортувати записи по типу процесора та вивести на екран у формі таблиці. Визначити комп’ютери з найбільшим обсягом оперативної і дискової пам’яті.

Використаний наступний код:

using System;

using System.Text;

namespace computerStock

{

public class Computer

{

// Комп'ютер

private string buildIndex;

private string buildName;

private string cpu;

private string cpuType;

private string matherboard;

private string ram;

private string gpu;

private string diskMem;

private string pcCase;

// Переферія

private string monitor;

private string keyboard;

private string mouse;

public Computer(

// Комп'ютер

string buildIndex,

string buildName,

string cpu,

string cpuType,

string gpu,

string matherboard,

string ram,

string diskMem,

string pcCase,

// Переферія

string monitor,

string mouse,

string keyboard

)

{

this.buildIndex = buildIndex;

this.buildName = buildName;

this.cpu = cpu;

this.cpuType = cpuType;

this.gpu = gpu;

this.matherboard = matherboard;

this.ram = ram;

this.diskMem = diskMem;

this.pcCase = pcCase;

this.monitor = monitor;

this.mouse = mouse;

this.keyboard = keyboard;

}

/\*\*

\* Функція показу списку пк

\*/

internal string returnMeName()

{

return buildName;

}

internal void showMeNames(int index, bool showIndex = true)

{

if(!showIndex == false)

{

index++;

Console.Write("\n - №" + index + "\t Комп'ютер: " + buildName + "; \t");

}

else

{

Console.Write(buildName);

}

}

/\*\*

\* Функція показу таблиці пк

\*/

internal void showMe(int index)

{

index =+ 1;

Console.Write("\n |~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|");

Console.Write("\n | |");

Console.Write("\n | № " + index + " Комп'ютер: " + buildName + ";");

Console.Write("\n | |");

Console.Write("\n |~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|");

Console.Write("\n | |");

Console.Write("\n |-----------------------------------------------------------------------------------|");

Console.Write("\n | Процессор: " + cpu + ";");

Console.Write("\n |-----------------------------------------------------------------------------------|");

Console.Write("\n | Тип процессора: " + cpuType + ";");

Console.Write("\n |-----------------------------------------------------------------------------------|");

Console.Write("\n | Відеокарта: " + gpu + ";");

Console.Write("\n |-----------------------------------------------------------------------------------|");

Console.Write("\n | Материнская плата: " + matherboard + ";");

Console.Write("\n |-----------------------------------------------------------------------------------|");

Console.Write("\n | Оперативна пам'ять: " + ram + ";");

Console.Write("\n |-----------------------------------------------------------------------------------|");

Console.Write("\n | Накопичувач: " + diskMem + ";");

Console.Write("\n |-----------------------------------------------------------------------------------|");

Console.Write("\n | Корпус: " + pcCase + ";");

Console.Write("\n |-----------------------------------------------------------------------------------|");

Console.Write("\n | Монітор: " + monitor + ";");

Console.Write("\n |-----------------------------------------------------------------------------------|");

Console.Write("\n | Мишка: " + mouse + ";");

Console.Write("\n |-----------------------------------------------------------------------------------|");

Console.Write("\n | Клавіатура: " + keyboard + ";");

Console.Write("\n |-----------------------------------------------------------------------------------|");

Console.Write("\n | |");

Console.Write("\n |~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|\n");

}

// Отримаємо тип процессору у числах

public string showType()

{

int valueType;

int.TryParse(string.Join("", cpuType.Where(c => char.IsDigit(c))), out valueType);

return valueType.ToString();

}

// Отримаємо кількість пам'яті

public int showMemory()

{

int valueRam;

int.TryParse(string.Join("", ram.Where(c => char.IsDigit(c))), out valueRam);

int valueDisk;

int.TryParse(string.Join("", diskMem.Where(c => char.IsDigit(c))), out valueDisk);

int value = valueRam + valueDisk;

return value;

}

internal static string WriteContent()

{

string content = Console.ReadLine();

bool indexTrue;

do

{

if (content != null)

{

return content;

indexTrue = true;

}

else

{

indexTrue = false;

return "Nothing";

}

} while (indexTrue != false);

}

internal bool EditKoll(int kollIndex)

{

do

{

switch (kollIndex)

{

case 0:

Console.Write("\n Процессор: ");

cpu = WriteContent();

return true;

case 1:

Console.Write("\n Тип процессора: ");

cpuType = WriteContent();

return true;

case 2:

Console.Write("\n Відеокарта: ");

gpu = WriteContent();

return true;

case 3:

Console.Write("\n Материнская плата: ");

matherboard = WriteContent();

return true;

case 4:

Console.Write("\n Оперативна пам'ять: ");

ram = WriteContent();

return true;

case 5:

Console.Write("\n Накопичувач: ");

diskMem = WriteContent();

return true;

case 6:

Console.Write("\n Корпус: ");

pcCase = WriteContent();

return true;

case 7:

Console.Write("\n Монітор: ");

monitor = WriteContent();

return true;

case 8:

Console.Write("\n Мишка: ");

mouse = WriteContent();

return true;

case 9:

Console.Write("\n Клавіатура: ");

keyboard = WriteContent();

return true;

}

}

while (true);

}

internal string returnFullString()

{

string stringFull =

buildIndex + "," +

buildName + "," +

cpu + "," +

cpuType + "," +

gpu + "," +

matherboard + "," +

ram + "," +

diskMem + "," +

pcCase + "," +

monitor + "," +

mouse + "," +

keyboard;

return stringFull;

}

}

public class Program

{

/\*\*

\* Параметри меню

\*/

static public void MenuChoose()

{

Console.WriteLine(" 1. Розпочати програму");

Console.WriteLine(" 2. Очистити консоль");

Console.WriteLine(" 3. Вихід");

}

/\*\*

\* Параметри під-меню

\*/

static public void SubMenuChoose()

{

Console.WriteLine("\n 1. Отримати інформацію о комп'ютерах");

Console.WriteLine(" 2. Редагувати інформацію о комп'ютерах");

Console.WriteLine(" 3. Повернутися");

}

/\*\*

\* Стартове меню

\*/

private static bool Menu()

{

Console.WriteLine("\n\n ### MENU ### ");

//Виводимо меню з пунктами

MenuChoose();

do

{

Console.Write("\n Ведіть одне з пунктів меню: ");

switch (Console.ReadLine())

{

case "1":

MainProgram();

return true;

case "2":

Console.Clear();

Menu();

return false;

case "3":

return false;

default:

Console.Clear();

Console.WriteLine("\n Немає такої команди в меню ведіть знову ! \n");

MenuChoose();

break;

}

} while (true);

}

/\*\*

\* Стартове підменю

\*/

private static bool SubMenu()

{

SubMenuChoose();

do

{

Console.Write("\n Ведіть одне з пунктів меню: ");

switch (Console.ReadLine())

{

case "1":

GeTInfoAbout();

return true;

case "2":

SetInfoAbout();

return true;

case "3":

Menu();

return false;

default:

Console.WriteLine("\n Немає такої команди в меню ведіть знову ! \n");

SubMenuChoose();

break;

}

} while (true);

}

/\*\*

\* Перевірка на відповідність вводу

\*/

static public double CheckIfNum(int amount)

{

double num = 0;

while (true)

{

if (double.TryParse(Console.ReadLine(), out num) && num > 0 && num <= amount + 1)

{

// Якщо було веденно вірне число буде закінчення циклу

return num - 1;

}

else

{

// Якщо було веденно невірне число буде повернення циклу

Console.WriteLine(" Ведено невірне значення. Спробуйте знову !");

Console.Write(" Ваша значення: ");

}

}

}

static public List<Computer> Indexing()

{

//Отримуємо данні з текстовой бази даних

List<string> lists = new List<string>();

foreach (string tmp in File.ReadAllLines(@"C:\Users\Alxich\Desktop\Labs\_Csh\_lang\Lab4\Task 1\Task\datebase.txt"))

{

lists.Add(tmp);

}

// Створюємо масив з новими екземплярами

List<Computer> koll = new List<Computer>();

foreach (string s in lists)

{

//Для кожного нового свойства (Типу комплектуючого й інше) відокремлюємо комою

string[] pcConfigFile = s.Split(',');

// Заповнення значень

string BuildIndex = pcConfigFile[0];

string BuildName = pcConfigFile[1];

string Cpu = pcConfigFile[2];

string CpuType = pcConfigFile[3];

string Gpu = pcConfigFile[4];

string Matherboard = pcConfigFile[5];

string Ram = pcConfigFile[6];

string DiskMem = pcConfigFile[7];

string PcCase = pcConfigFile[8];

string Monitor = pcConfigFile[9];

string Mouse = pcConfigFile[10];

string Keyboard = pcConfigFile[11];

// Вносимо в масив новий екземпляр

koll.Add(new Computer(BuildIndex, BuildName, Cpu, CpuType, Gpu, Matherboard, Ram, DiskMem, PcCase, Monitor, Mouse, Keyboard));

}

return koll;

}

static public void WritingMeth(List<Computer> koll)

{

using (TextWriter tw = new StreamWriter(@"C:\Users\Alxich\Desktop\Labs\_Csh\_lang\Lab4\Task 1\Task\datebase.txt"))

{

foreach (var PC in koll)

{

tw.WriteLine(string.Format(PC.returnFullString()));

}

}

}

/\*\*

\* Отримати інформацію о комп'ютерах

\*/

static public void GeTInfoAbout()

{

List<Computer> koll = Indexing();

Console.Clear();

// Показуємо топ пк за кількістю пам'яті

string showTop(int length)

{

List<Computer> koll2 = new List<Computer>(koll);

// Сортируємо за кількістю пам'яті

koll2.Sort(delegate (Computer pc1, Computer pc2)

{

int pc1Amount = pc1.showMemory();

int pc2Amount = pc2.showMemory();

return pc2Amount.CompareTo(pc1Amount);

});

// Показуємо найбільший за пам'ятю пк

Console.WriteLine();

Console.Write("\n Назва комп'ютера\n");

int indexCount = 1;

for (int i = 0; i <= 4; i++)

{

Console.Write(" \n -" + koll2[i].returnMeName() + " ");

indexCount++;

}

Console.WriteLine();

return koll2[0].returnMeName();

}

// Сортируємо за типом процесора

koll.Sort(delegate (Computer pc1, Computer pc2)

{

return pc1.showType().CompareTo(pc2.showType());

});

// Виводимо список з усіма доступними пк

Console.WriteLine("\n Комп'ютери котрі є у базі.");

Console.Write("\n Індекс\t Назва комп'ютера");

int kollLength = -1;

int i = 0;

foreach (var PC in koll)

{

PC.showMeNames(i);

i++;

kollLength++;

}

// Вибираємо один пк з списку (Зроблено щоб не перевантажувати интерфейс користувачу)

Console.WriteLine();

Console.Write("\n Про який комп'ютер хочете дізнатись більше? \n");

Console.Write(" Перші 5 комп'ютера з найбільшим обсягом оперативної і дискової пам’яті: ");

showTop(kollLength);

Console.Write("\n Я хочу дізнатись про комп'ютер з індексом: ");

double theIndexIWant = 0;

theIndexIWant = CheckIfNum(kollLength);

// Вибираємо пк за індексом і виконуємо функцію показу таблиці

koll[(int)theIndexIWant].showMe((int)theIndexIWant);

SubMenu();

}

/\*\*

\* Редагувати один з комп'ютеров

\*/

static public void SetInfoAbout()

{

List<Computer> koll = Indexing();

Console.WriteLine("\n Комп'ютери котрі є у базі.");

Console.Write("\n Індекс\t Назва комп'ютера");

int kollLength = -1;

int i = 0;

foreach (var PC in koll)

{

PC.showMeNames(i);

i++;

kollLength++;

}

Console.WriteLine("\n Котрий з цих комп'ютерів ви хочете редагувати ?");

Console.Write(" Індекс комп'ютера: ");

int theIndexIWant = (int)CheckIfNum(kollLength);

koll[theIndexIWant].showMe(theIndexIWant);

Console.WriteLine("\n Виберіть колонку котру хочете редагувати:");

Console.Write("\n 1. Процессор");

Console.Write("\n 2. Тип процессора");

Console.Write("\n 3. Відеокарта");

Console.Write("\n 4. Материнская плата");

Console.Write("\n 5. Оперативна пам'ять");

Console.Write("\n 6. Накопичувач");

Console.Write("\n 7. Корпус");

Console.Write("\n 8. Монітор");

Console.Write("\n 9. Мишка");

Console.Write("\n 10. Клавіатура");

Console.Write("\n\n Виберіть одне з списку: ");

int editIndex = (int)CheckIfNum(9);

Console.WriteLine();

koll[theIndexIWant].EditKoll(editIndex);

Console.Clear();

Console.WriteLine();

Console.Write("\n\n Результат вашого редагування: ");

koll[theIndexIWant].showMe(theIndexIWant);

WritingMeth(koll);

SubMenu();

}

/\*\*

\* Головна програма

\*/

static public void MainProgram()

{

SubMenu();

}

/\*\*

\* Виведення результату програми

\*/

static void Main()

{

Console.OutputEncoding = System.Text.Encoding.Default;

Menu();

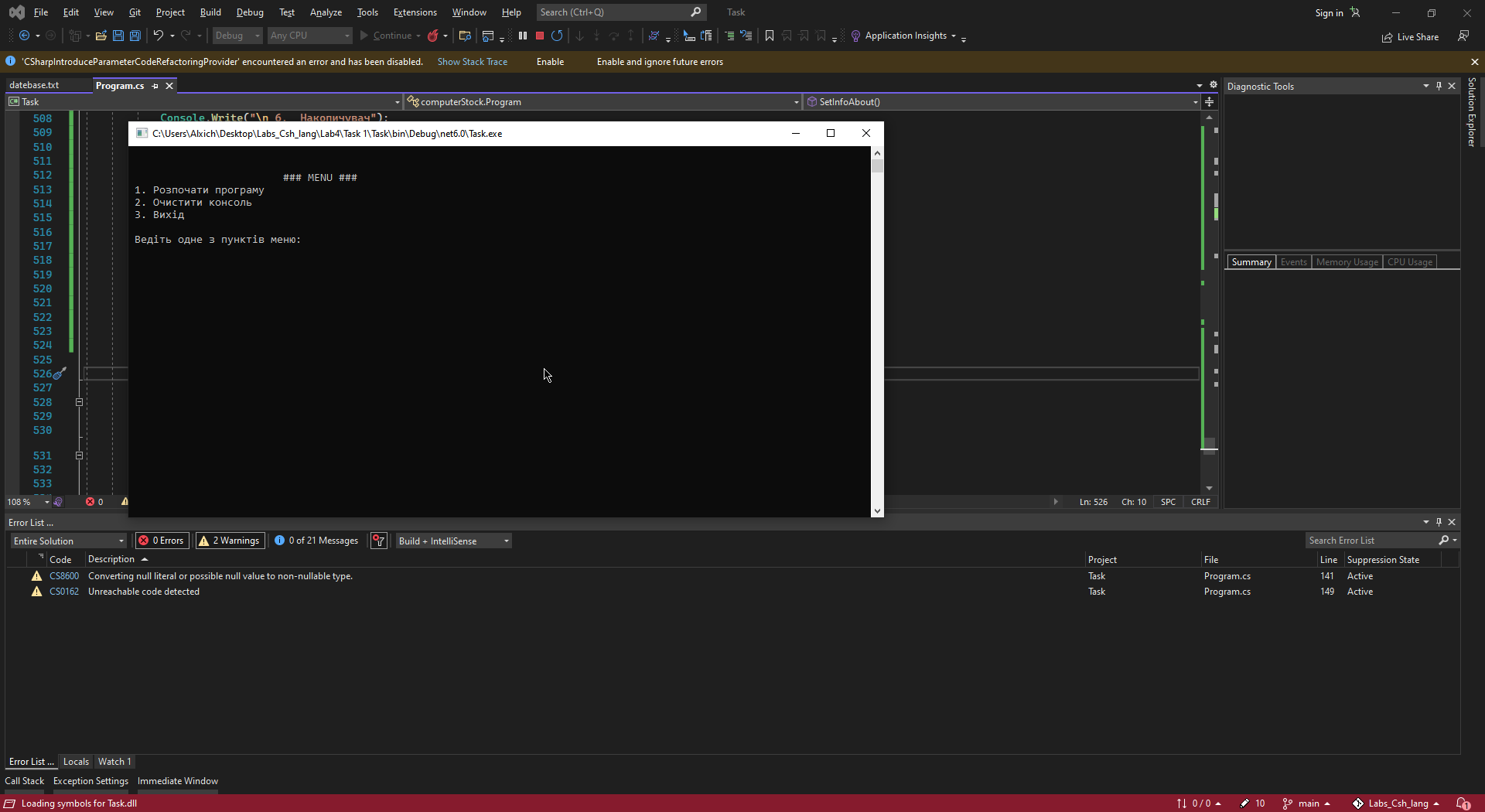
}

}

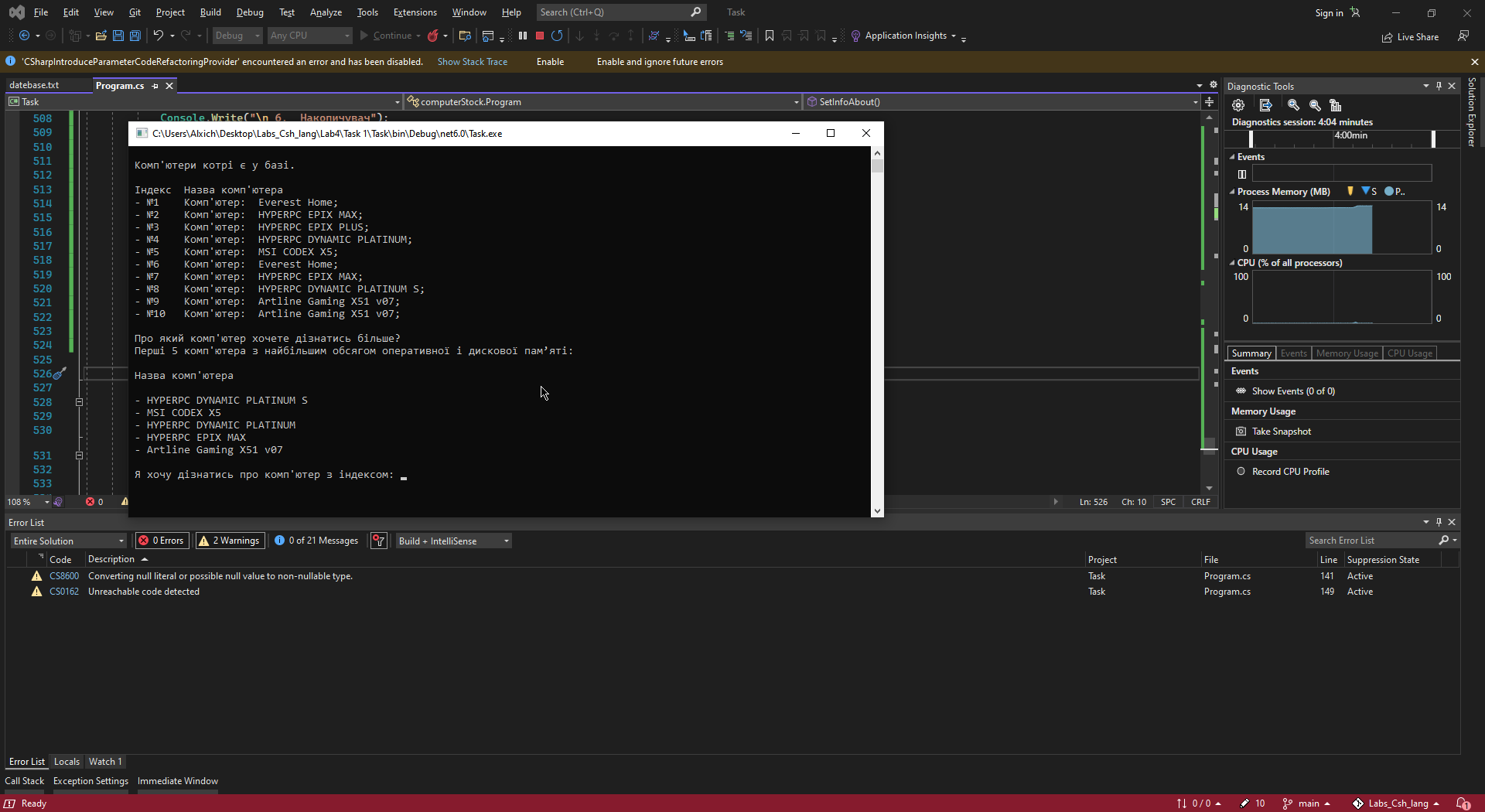
}

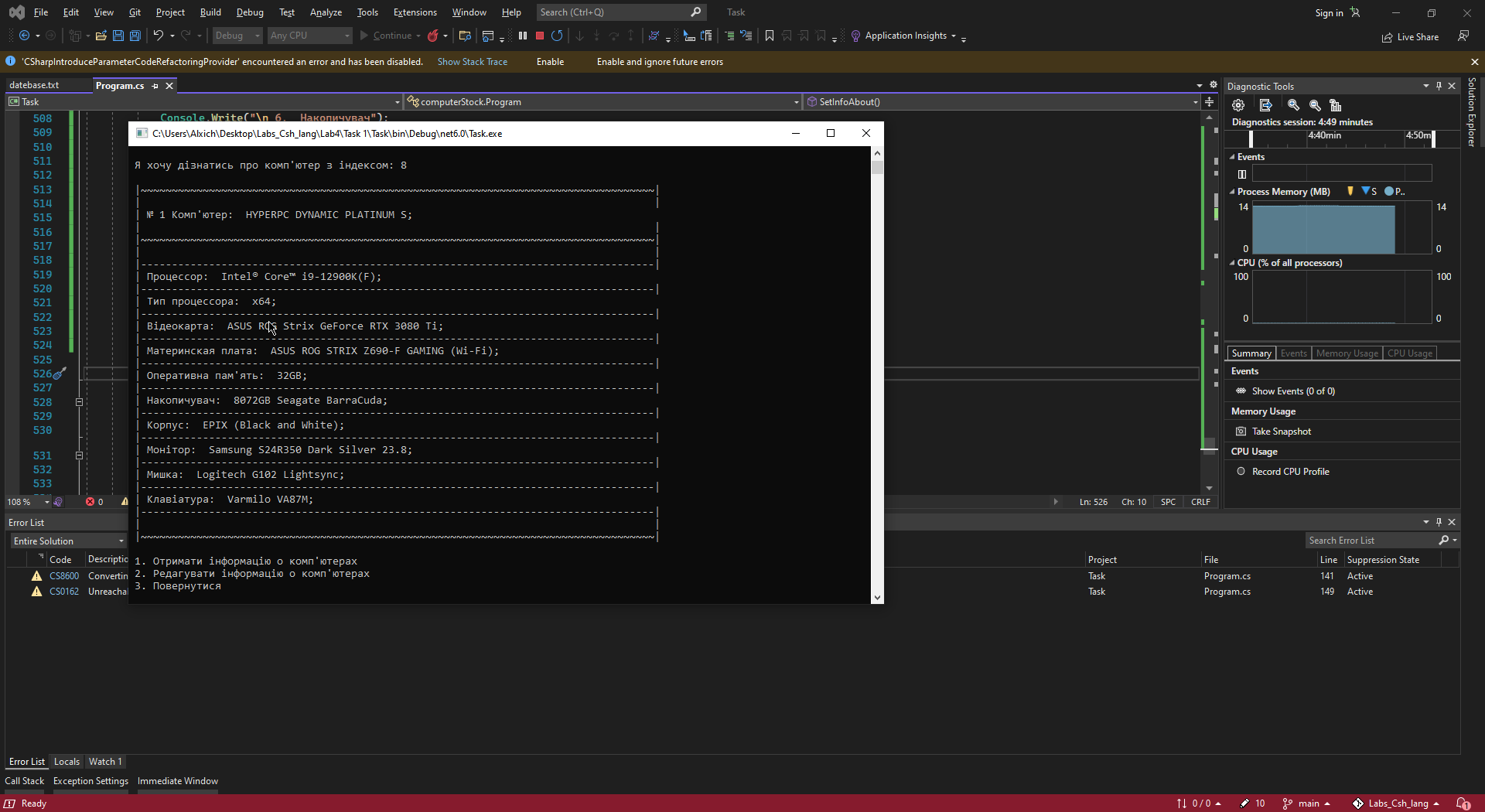
**Результат:**

Початок програми

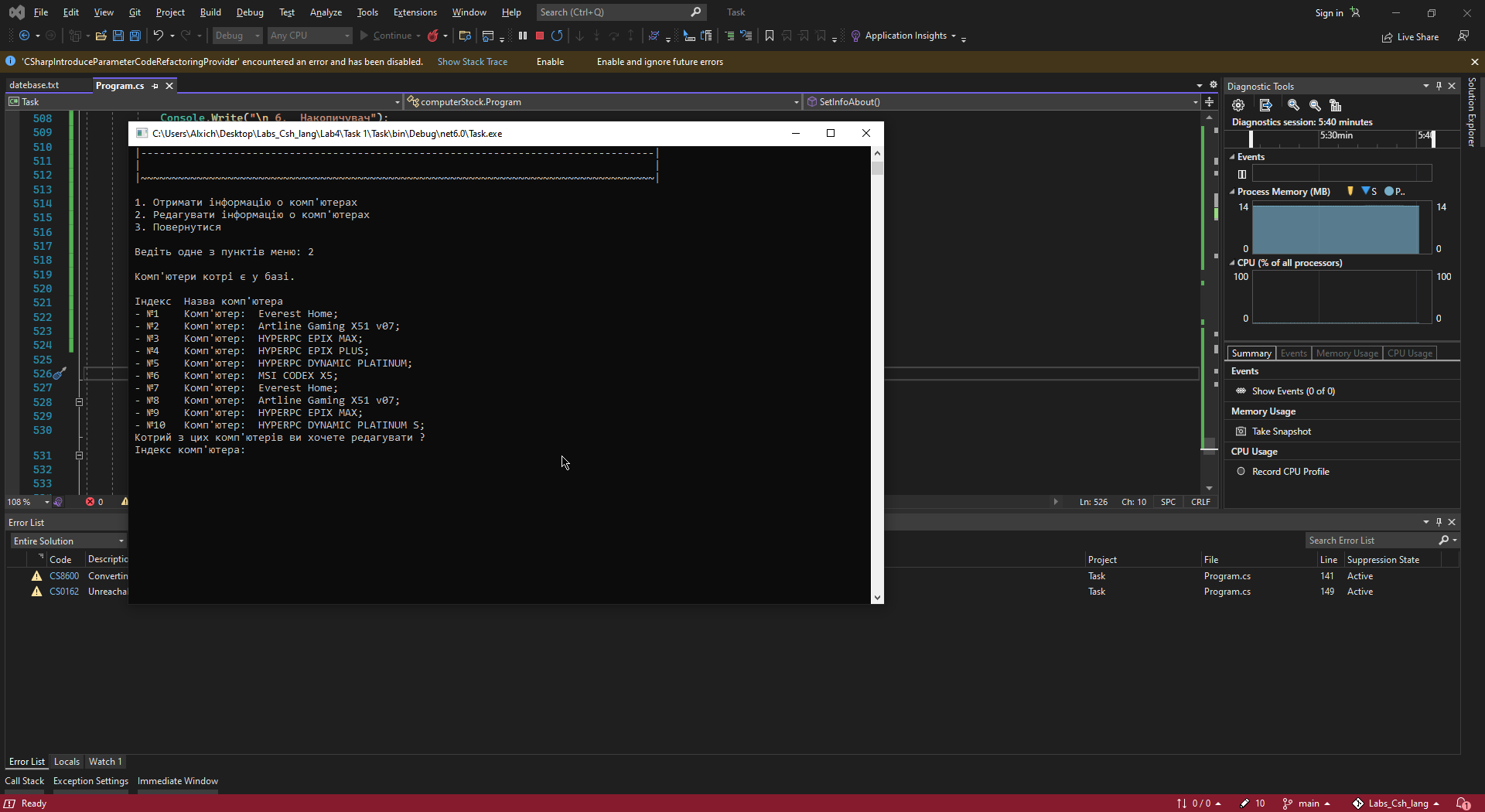


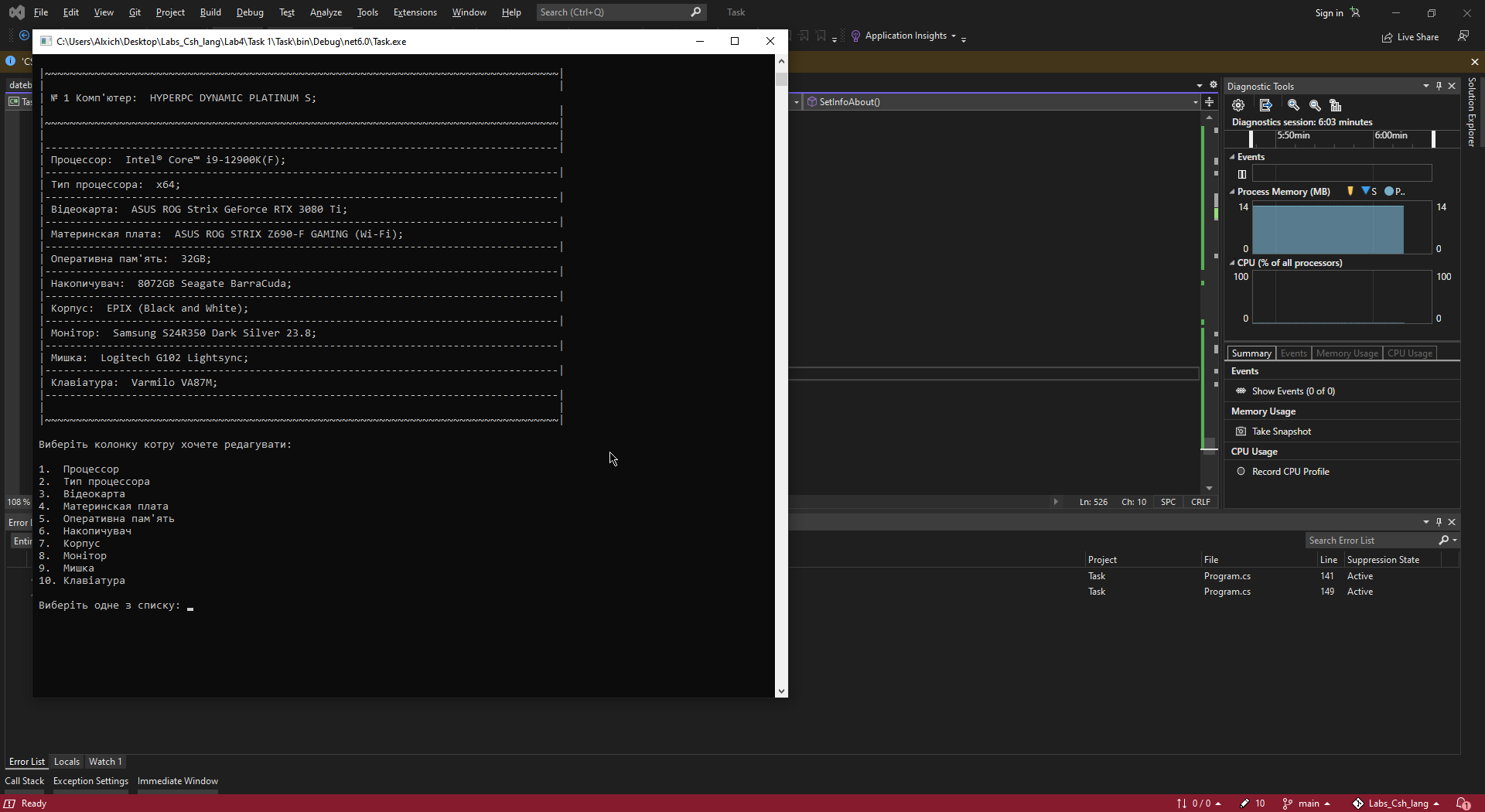
Початок виводу таблиці та результат

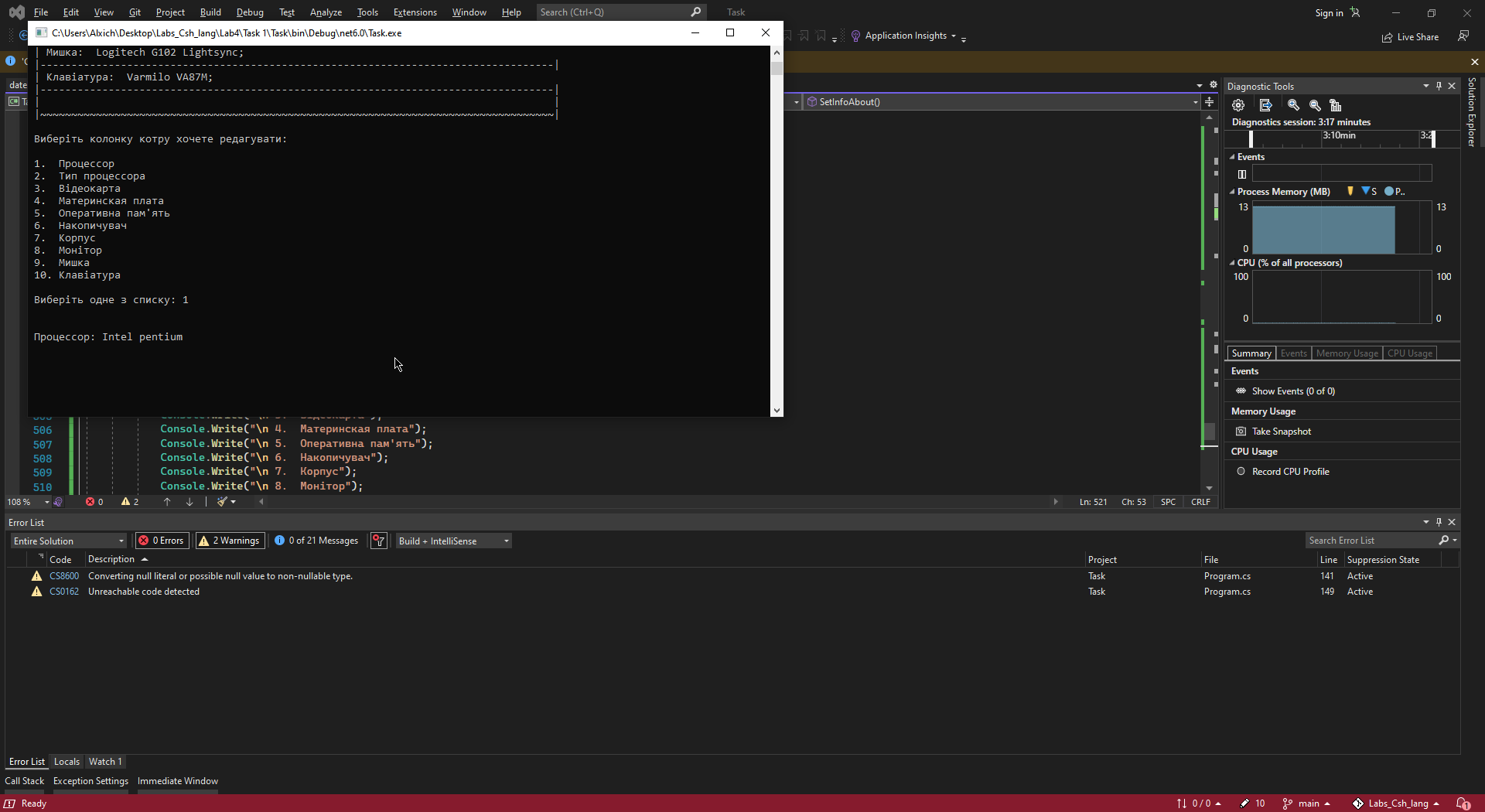


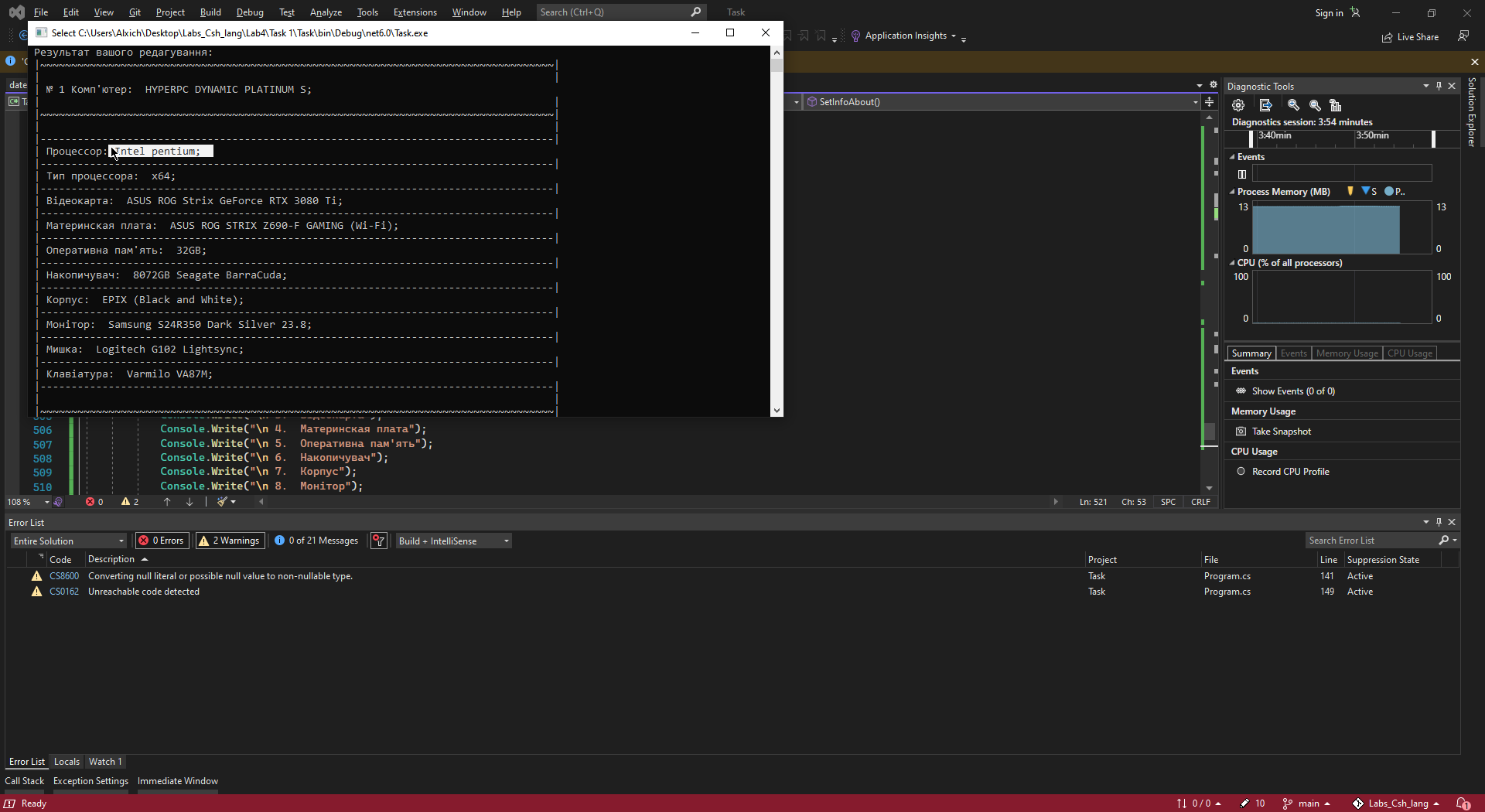


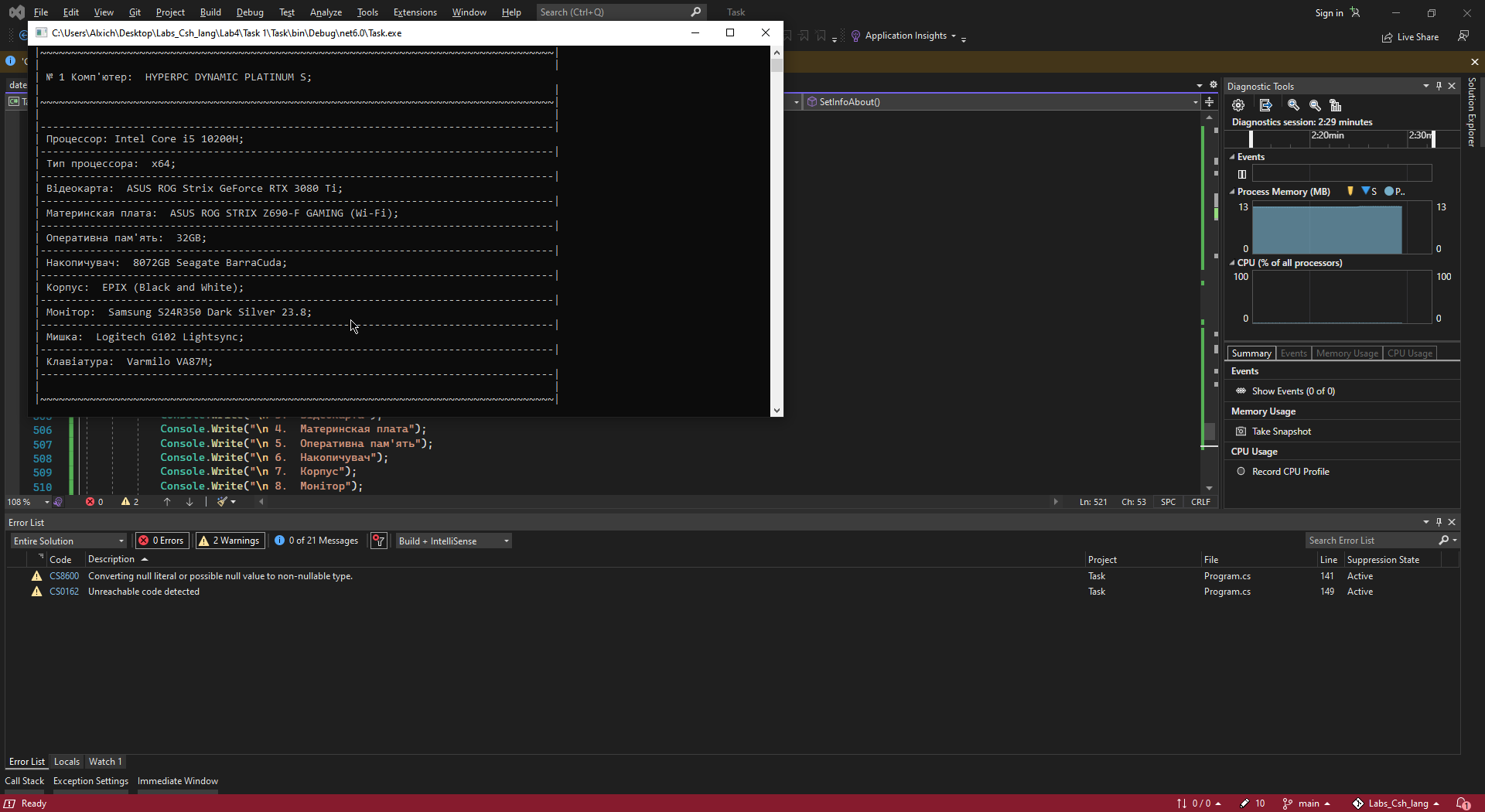
Редагування значення одного з комп’ютера



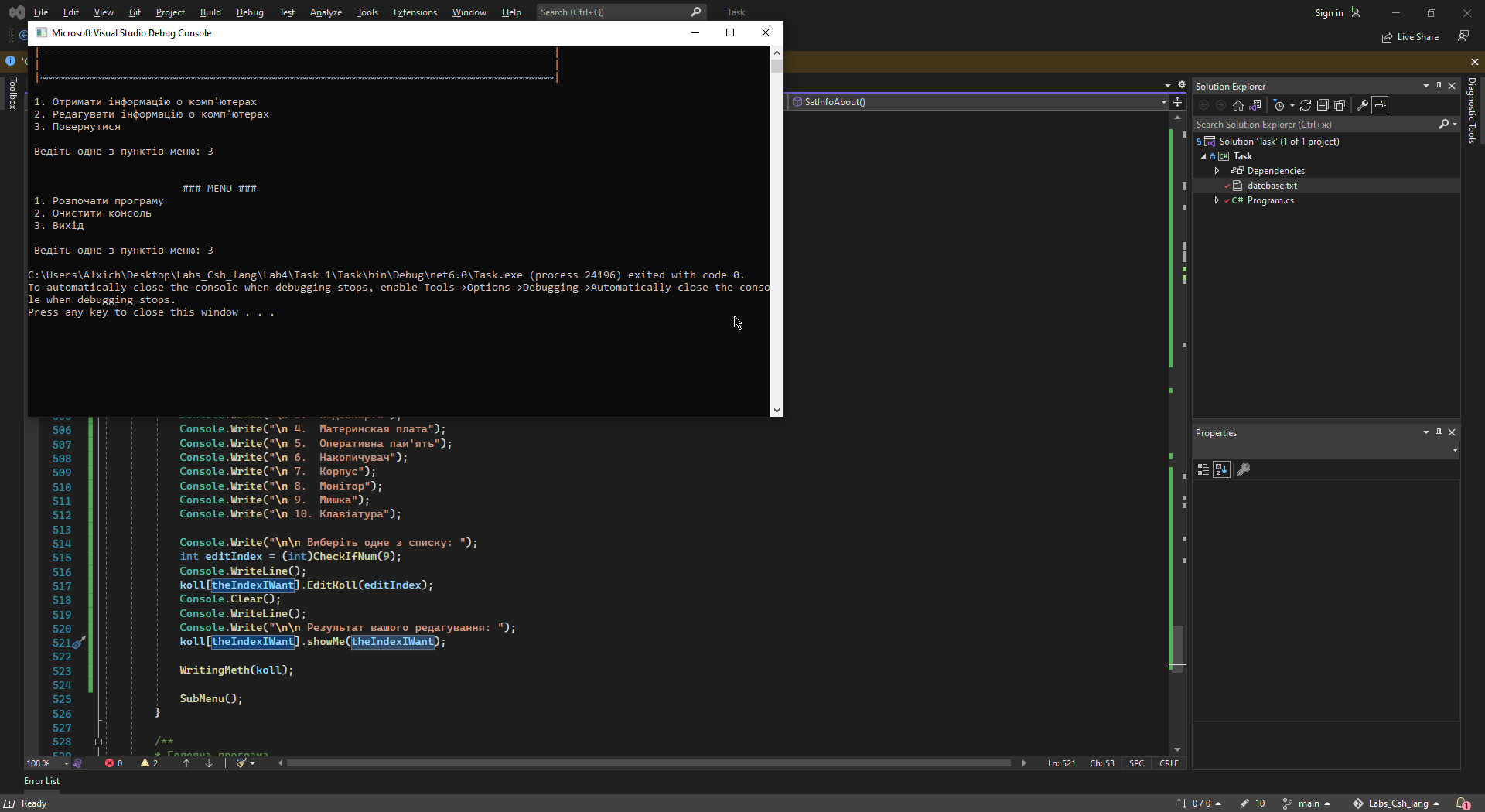








Вихід з програми



**(3 бал)**

***2.Розробити консольний застосунок для роботи з базою даних, що зберігається у текстовому файлі (початковий масив не менше 10 записів). Структура бази даних описується класом згідно вашого варіанта. Передбачити роботу з довільною кількістю***

***записів. Для ідентифікації спроби введення з клавіатури некоректних даних описати виключення. Всі поля класу зробити закритими, а доступ до них реалізувати через get i set.***

***Реалізувати конструктори з параметрами та без параметрів, а ініціалізацію полів виконати через властивості. Реалізувати методи для:***

***– додавання записів;***

***– редагування записів;***

***– знищення записів;***

***– виведення інформації з файла на екран;***

***– пошук потрібної інформації за конкретною ознакою (поле Параметр пошуку);***

***– сортування за різними полями (поле Параметр сортування).***

***Меню програми реалізувати по натисненню на певні клавіші: наприклад, Enter – вихід, п - пошук, р – редагування тощо.***

**Завдання:**



Не виконано