## Звіт з лабораторної роботи №3

### Масиви об’єктів

### Мета роботи: Ознайомитися з особливостями ООП, навчитися створювати масиви об’єктів

### 

***Варіант* №22**

Створити масив об’єктів класу „матеріальна точка”. Кожна точка характеризується координатами початкового пололження та вектором рівномірної швидкості . Визначити номера точок, які за введений користувачем час *t* попадають в перший октант.

Текст програми:

namespace Lab02

{

internal class MaterialPoint

{

private double x; // Поле класу

private double y;

private double z;

private double vx;

private double vy;

private double vz;

// Конструктор з параметрами

public MaterialPoint(double x = 0, double y = 0, double z = 0, double vx = 0, double vy = 0, double vz = 0)

{

this.x = x; //SetX (х) //

this.y = y;

this.z = z;

this.vx = vx;

this.vy = vy;

this.vz = vz;

}

public void SetX(double x)

{

this.x = x;

}

public void SetY(double y)

{

this.y = y;

}

public void SetZ(double z)

{

this.z = z;

}

public void SetVX(double vx)

{

this.vx = vx;

}

public void SetVY(double vy)

{

this.vy = vy;

}

public void SetVZ(double vz)

{

this.vz = vz;

}

public double GetX()

{

return this.x;

}

public double GetY()

{

return this.y;

}

public double GetZ()

{

return this.z;

}

public double GetVX()

{

return this.vx;

}

public double GetVY()

{

return this.vy;

}

public double GetVZ()

{

return this.vz;

}

public bool IsInFirstOctant(double t)

{

// x1, y1, z1 це координати точки після переміщення за час t

double x1 = this.x + vx \* t;

double y1 = this.y + vy \* t;

double z1 = this.z + vz \* t;

if (x1 > 0 && y1 > 0 && z1 > 0)

{

return true;

}

return false;

}

public void Input()

{

Console.Write("x = ");

x = double.Parse(Console.ReadLine());

Console.Write("y = ");

y = double.Parse(Console.ReadLine());

Console.Write("z = ");

z = double.Parse(Console.ReadLine());

Console.Write("vx = ");

vx = double.Parse(Console.ReadLine());

Console.Write("vy = ");

vy = double.Parse(Console.ReadLine());

Console.Write("vz = ");

vz = double.Parse(Console.ReadLine());

}

public void Print()

{

Console.WriteLine($"x = {x}, y = {y}, z = {z}");

Console.WriteLine($"vx = {vx}, vy = {vy}, vz = {vz}");

}

}

//Program.cs

Console.WriteLine("Лабораторна 3. Масиви обєктів");

MaterialPoint[] array = new MaterialPoint[3];

for (int i = 0; i < array.Length; i++)

{

array[i] = new MaterialPoint();

Console.WriteLine($"Введіть координати і швидкість для точки {i + 1}:");

array[i].Input();

}

Console.Write($"Введіть час = ");

double time = double.Parse(Console.ReadLine());

for (int i = 0; i < array.Length; i++)

{

if (array[i].IsInFirstOctant(time))

{

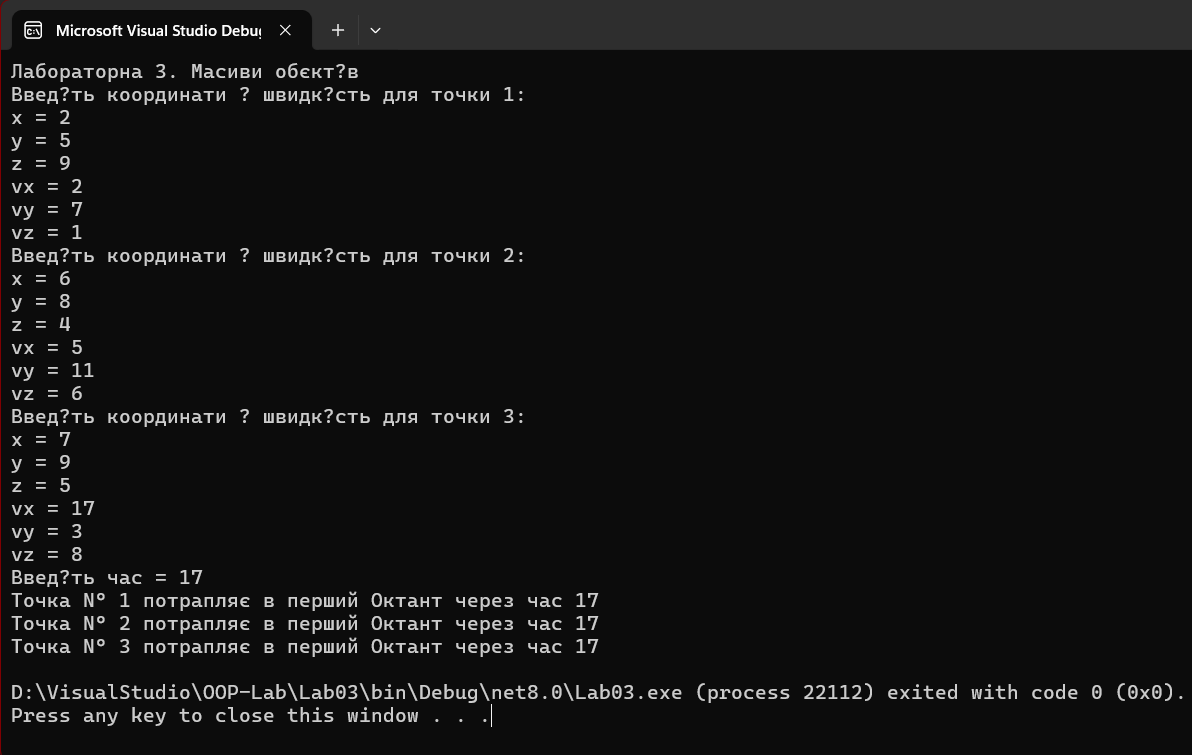
Console.WriteLine($"Точка N° {i + 1} потрапляє в перший Октант через час {time}");

}

}

}

Скрін виконання програми:



//Програма демонстрації роботи збірника сміття:

class LargeObject

{

private int[] data;

private int id;

private static int lastId = 0;

public LargeObject(int size)

{

// Ініціалізуємо великий масив (розміру size)

data = new int[size];

id = ++lastId;//ID об'єкта (порядковий номер)

Console.ForegroundColor = ConsoleColor.Blue;

Console.WriteLine($"Object created: {id}");

}

// Деструктор (finalizer) - буде спрацьовувати при вилученні об'єкта збіриником сміття

~LargeObject()

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine($"Destructor called for object:# {id}");

}

}

class Program

{

static void Main(string[] args)

{

// Створили об'єкт для вимірювання часу

Stopwatch stopwatch = new Stopwatch();

// Змінні для відстеження використання пам'яті

long memoryBefore, memoryAfter;

// Початкові заміри пам'яті

memoryBefore = GC.GetTotalMemory(false);

Console.WriteLine($"Memory before creation: {memoryBefore / 1024} KB");

// Кількість створюваних об'єктів

const int numObjects = 100000; // 1000

stopwatch.Start(); // починає вимірювати час

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

{

var obj = new LargeObject(100000);//Створення великого об'єкта

// Зануляємо посилання, щоб об'єкт став доступним для збору сміття

if (i % 100 == 0) obj = null;

//Примусовий виклик збирача сміття кожні 100 ітерацій

if (i % 100 == 0)

{

GC.Collect();

GC.WaitForPendingFinalizers();

}

}

stopwatch.Stop();//Зупинили вимірювання часу

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine($"Object creation completed in: {stopwatch.ElapsedMilliseconds} ms");

// Кількість зборів для кожного покоління

Console.WriteLine($"Gen 0 collections: {GC.CollectionCount(0)}");

Console.WriteLine($"Gen 1 collections: {GC.CollectionCount(1)}");

Console.WriteLine($"Gen 2 collections: {GC.CollectionCount(2)}");

Console.ResetColor();

Console.ReadKey();

// Замір пам'яті після створення об'єктів

memoryAfter = GC.GetTotalMemory(false);

Console.WriteLine($"Memory after creation: {memoryAfter / 1024} KB");

// Додатковий виклик збирача сміття

Console.WriteLine("Forcing garbage collection...");

GC.Collect();

GC.WaitForPendingFinalizers();

Console.ResetColor();

// Замір пам'яті після збору сміття

long memoryAfterGC = GC.GetTotalMemory(false);

Console.WriteLine($"Memory after GC: {memoryAfterGC / 1024} KB");

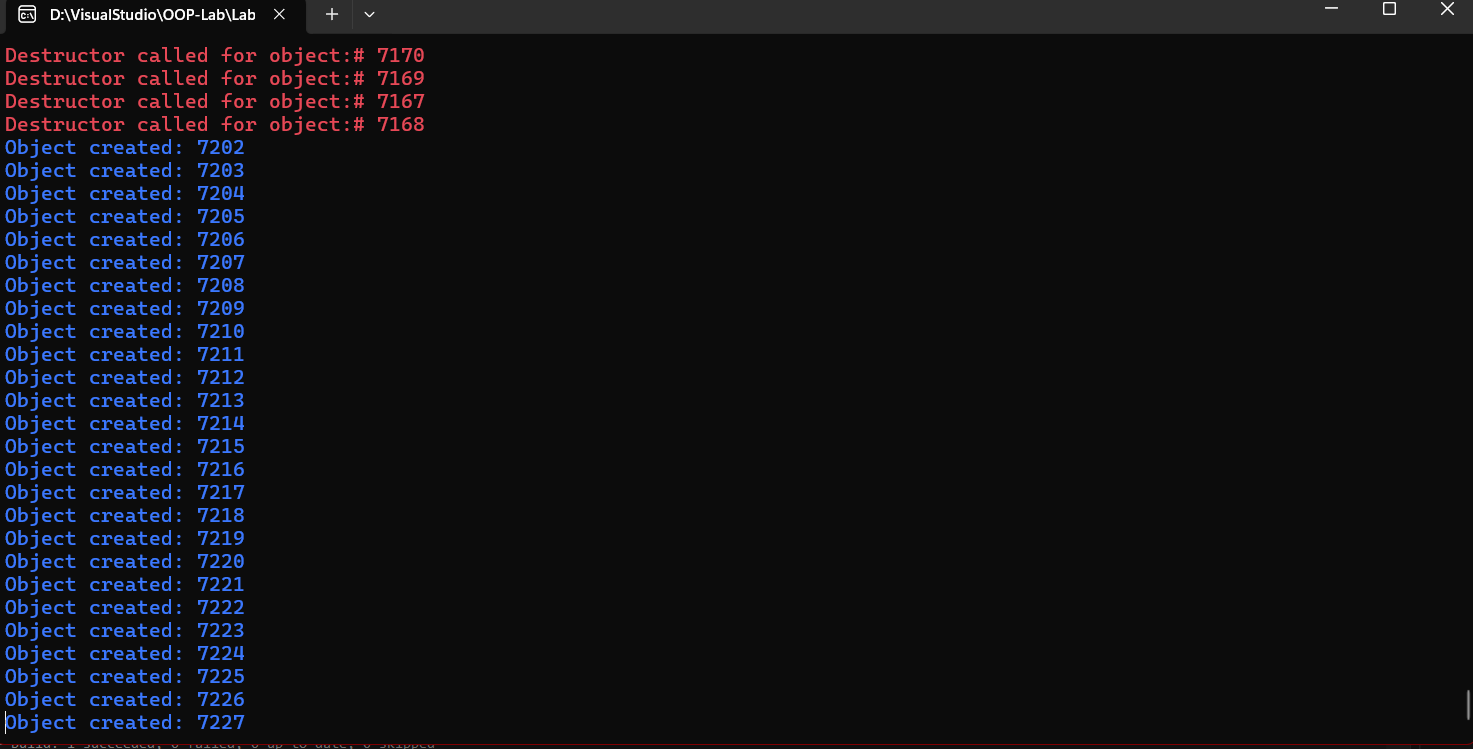
Console.WriteLine("Program finished. Press any key to exit.");

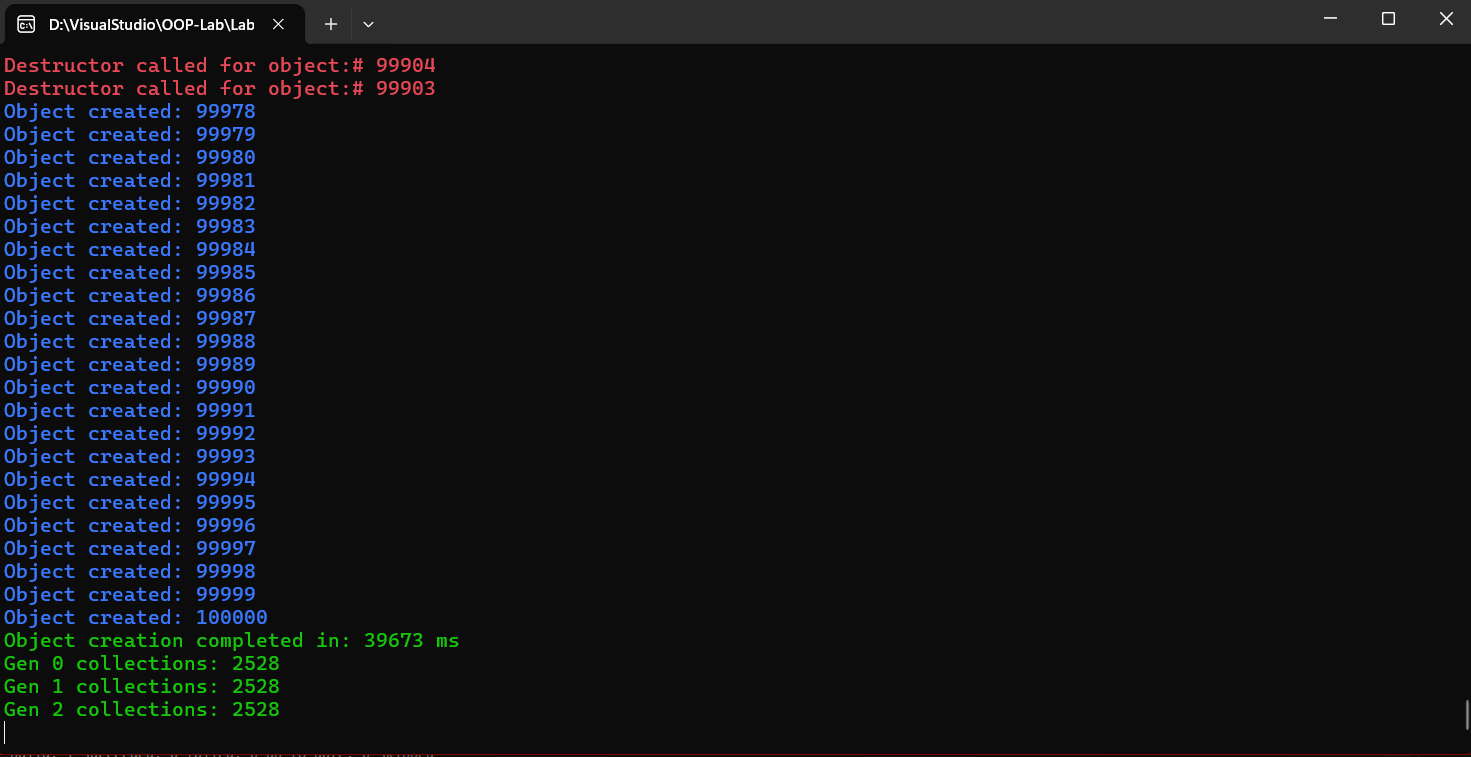
Console.ReadKey();

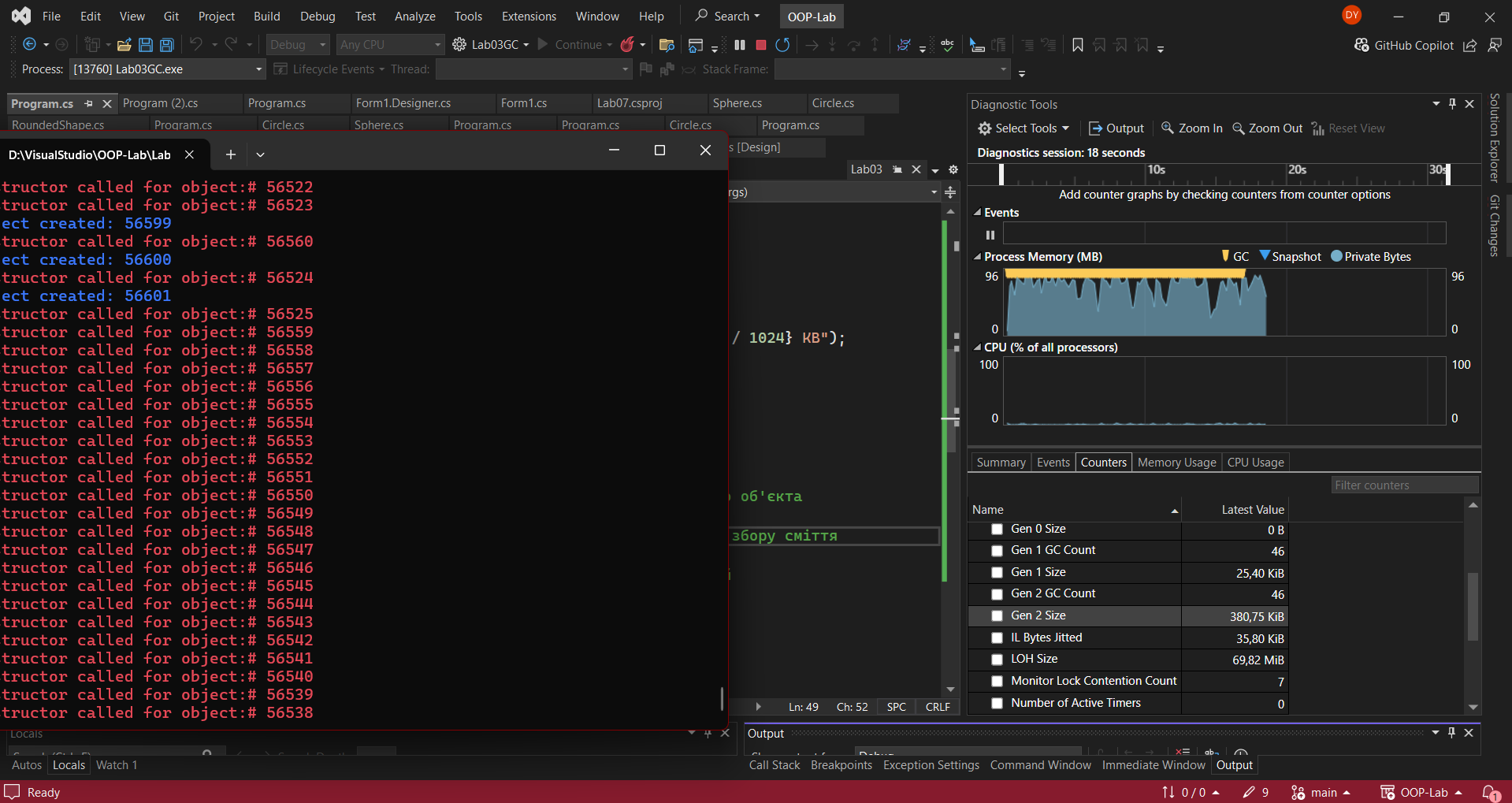
}

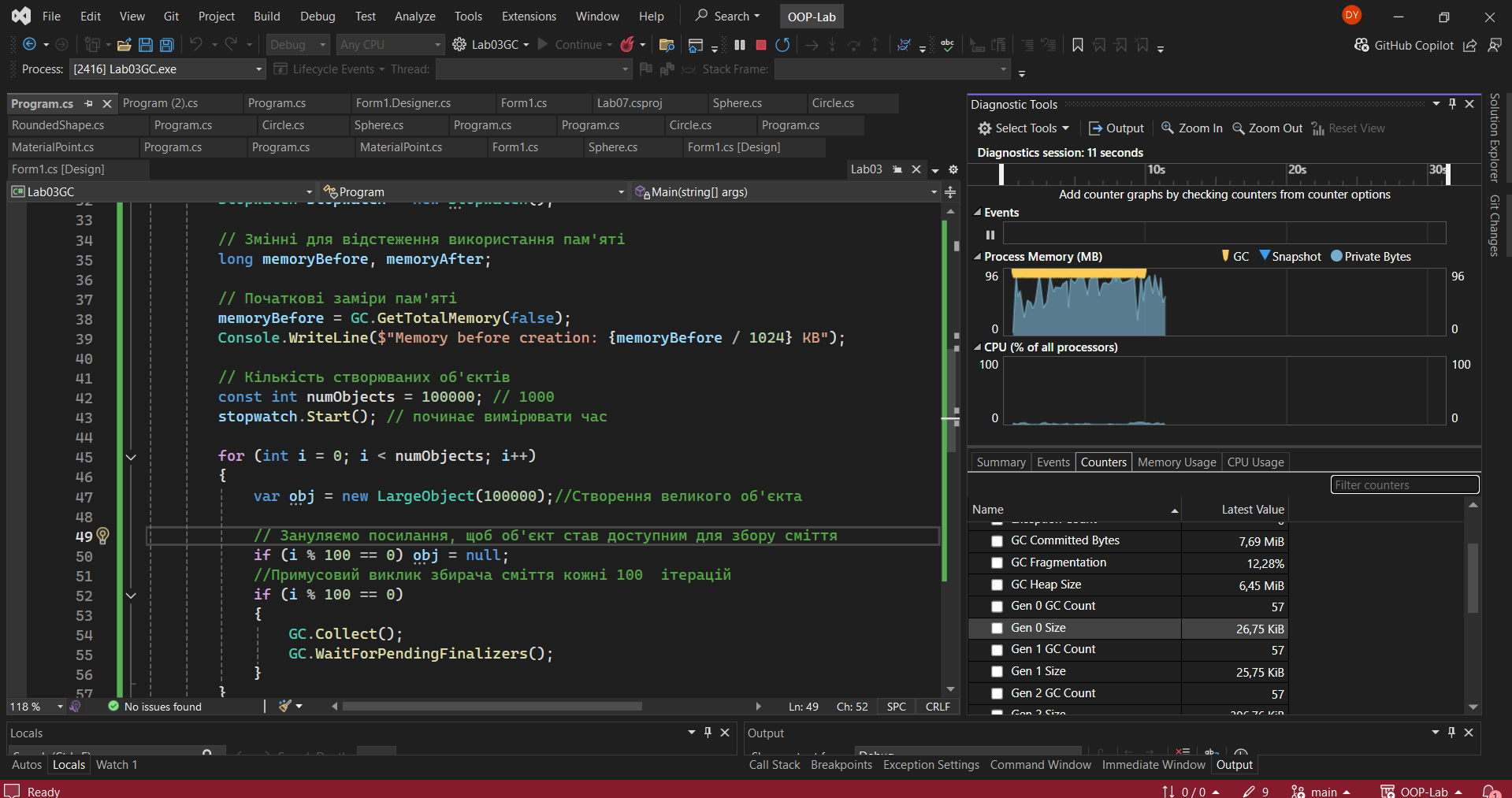
}

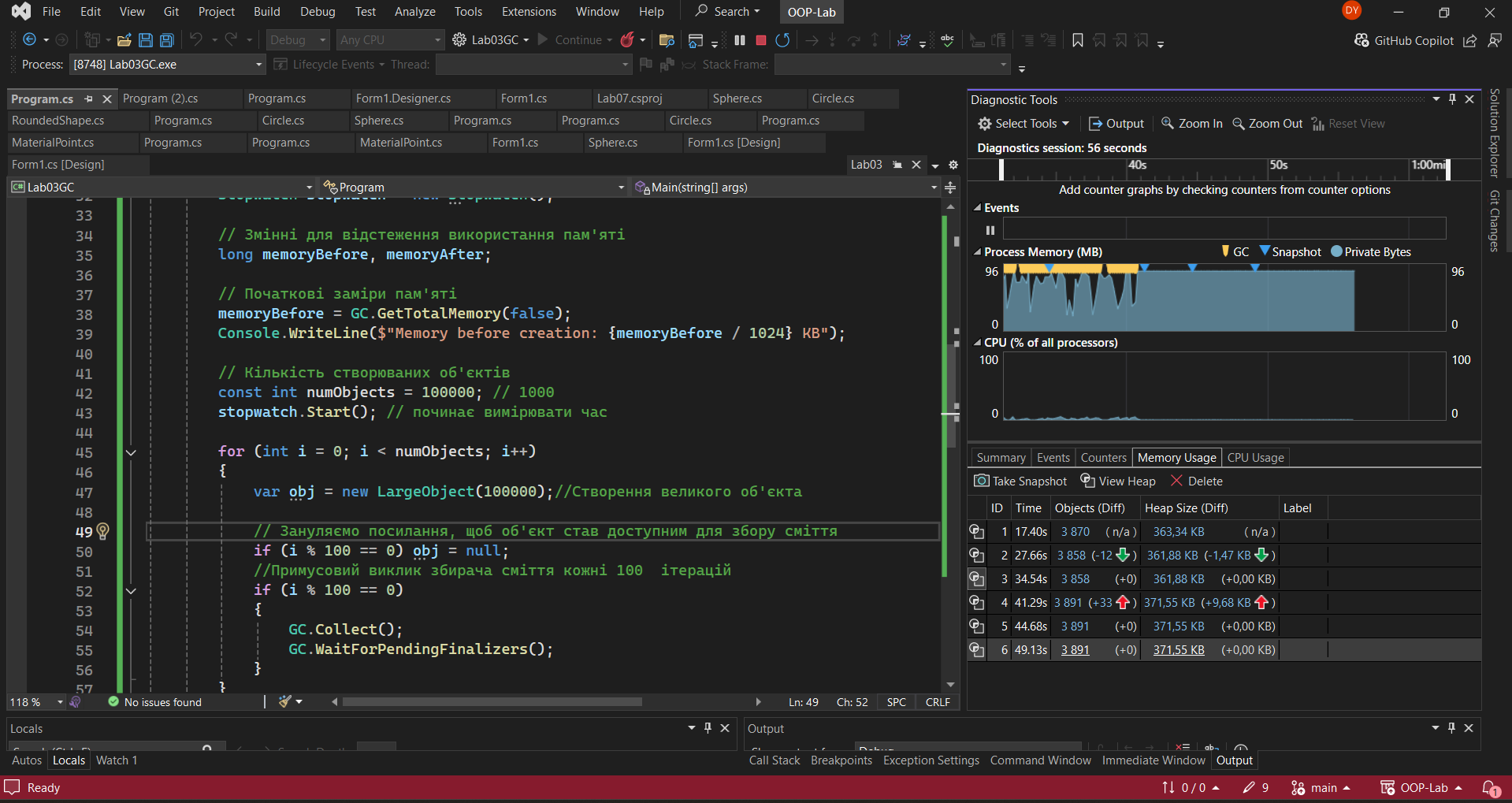
Скріни роботи програми:











Висновки

В ході виконання роботи було створено масив об’єктів класу Матеріальна точка, та визначено номера точок, які за введений користувачем час потрапляють в перший октант. При виконані роботи було використано клас матеріальної точки, розроблений в лабораторній роботі №2 (згідно варіанту). Клас матеріальна точка доповнений деструктором. Окрім того в ході виконання лабораторної було досліджено збірник сміття. Було створено клас LargeObject з конструктором та деструктором. Конструктор цього класу виводить повідомлення про створення об’єкта, деструктор цього класу сповіщає про вилучення об’єкта. У функції Main() створено велике число об’єктів типу LargeObject деяким посиланням присвоєно null з метою їх вилучення збірником сміття. Кожні сто ітерацій циклу примусово викликався збірник сміття (GC.Collect()). Під час робити програми спостерігали роботу збірника сміття (відповідні повідомлення деструктора). Також було виконано заміри часу, пам’яті, та числа збірок сміття у різних поколіннях (GC.CollectionCount()).