Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 2 з дисципліни

«Основи програмування-2. Методології програмування»

«Класи та об'єкти»

Варіант 06

Виконав студент ІП-11 Головатюк Владислав Ігорович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 2022

**Лабораторна робота №2**

**Класи та об'єкти**

***Мета роботи:*** вивчити механізми створення і використання класів та об'єктів.

***Завдання:*** Розробити клас «площина *Ax + By + Cz + D = 0*», яка задана своїми коєфіцієнтами. Створити масив об'єктів даного класу. Визначити площини, яким належить введена користувачем точка.

***Постановка задачі:*** Створюємо класс площини. В собі він містить 4 коефіцієнта, конструктор для присвоювання всіх значень коефіцієнців, метод для перевірки, чи належить точка цій площині. Також створюємо класс точки з 3 координатами та конструктором для їх присвоювання

1. **С#**
   1. **Код:**
      1. **Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab3

{

internal class Program

{

private static Random random = new Random();

static void Main(string[] args)

{

List<Plane> planes = new List<Plane>();

planes = new List<Plane>() {

new Plane(random.Next(-10,10),random.Next(-10,10),random.Next(-10,10), random.Next(-10,10)),

new Plane(random.Next(-10,10),random.Next(-10,10),random.Next(-10,10), random.Next(-10,10)),

new Plane(random.Next(-10,10),random.Next(-10,10),random.Next(-10,10), random.Next(-10,10)),

new Plane(random.Next(-10,10),random.Next(-10,10),random.Next(-10,10), random.Next(-10,10)),

new Plane(random.Next(-10,10),random.Next(-10,10),random.Next(-10,10), random.Next(-10,10)),

new Plane(random.Next(-10,10),random.Next(-10,10),random.Next(-10,10), random.Next(-10,10))

};

//do

//{

// planes.Add(ConsoleWorker.GetPlaneFromUser());

// Console.WriteLine("================");

//} while (ConsoleWorker.GetAnswer("Continue? [y/n]: "));

ConsoleWorker.PrintPlanes("All planes: ", planes.ToArray());

Point = ConsoleWorker.GetPointFromUser();

ConsoleWorker.PrintPlanes("Planes, what owns this point: ",

planes.Where(p => p.IsOwnsPoint(point)).ToArray());

Console.ReadLine();

}

}

}

* + 1. **Plane.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab3

{

internal class Plane

{

public double A { get; private set; }

public double B { get; private set; }

public double C { get; private set; }

public double D { get; private set; }

public Plane(double a, double b, double c, double d)

{

A = a;

B = b;

C = c;

D = d;

}

public bool IsOwnsPoint(Point point)

{

return A \* point.X + B \* point.Y + C \* point.Z + D \* 1 == 0;

}

public override string ToString()

{

string result = "";

double[] coeffs = new double[] { A, B, C, D };

string[] varChars = new string[] { "x", "y", "z", "" };

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

{

if (coeffs[i] == 0) continue;

result += coeffs[i].ToString("+ 0.0;- 0.0") + varChars[i] + " ";

}

if (result.StartsWith("+"))

result = result.Substring(2);

return result + "= 0";

}

}

}

* + 1. **Point.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab3

{

internal class Point

{

public double X { get; private set; }

public double Y { get; private set; }

public double Z { get; private set; }

public Point(double x, double y, double z)

{

X = x;

Y = y;

Z = z;

}

public override string ToString()

{

return $"({X}; {Y}; {Z})";

}

}

}

* + 1. **ConsoleWorker.cs**

using System;

namespace Lab3

{

internal static class ConsoleWorker

{

public static Point GetPointFromUser()

{

return new Point(

Convert.ToDouble(Input("Enter x like 0,00: ", str => double.TryParse(str, out \_))),

Convert.ToDouble(Input("Enter y like 0,00: ", str => double.TryParse(str, out \_))),

Convert.ToDouble(Input("Enter z like 0,00: ", str => double.TryParse(str, out \_)))

);

}

public static Plane GetPlaneFromUser()

{

return new Plane(

Convert.ToDouble(Input("Enter A like 0,00: ", str => double.TryParse(str, out \_))),

Convert.ToDouble(Input("Enter B like 0,00: ", str => double.TryParse(str, out \_))),

Convert.ToDouble(Input("Enter C like 0,00: ", str => double.TryParse(str, out \_))),

Convert.ToDouble(Input("Enter D like 0,00: ", str => double.TryParse(str, out \_)))

);

}

public static bool GetAnswer(string header)

{

return Input(header, str => str == "y" || str == "n") == "y";

}

public static void PrintPlanes(string header, Plane[] elems)

{

Console.WriteLine(header + "\n===============");

foreach (Plane item in elems)

{

Console.WriteLine(item.ToString() + "\n===============");

}

}

public static string Input(string header, Func<string, bool> func)

{

Console.Write(header);

string input = Console.ReadLine();

while (func(input) == false)

{

Console.Write("Incorect input. Try again: ");

input = Console.ReadLine();

}

return input;

}

public static string Input(string header)

{

Console.Write(header);

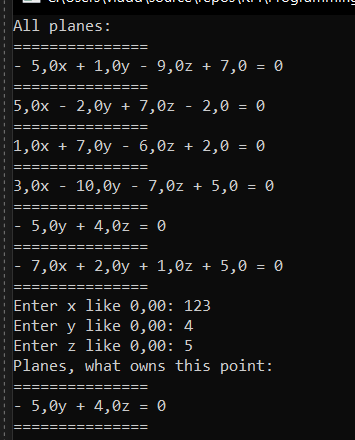
return Console.ReadLine();

}

}

}

* 1. **Запуск програми**

****

***Висновок:*** В процесі виконання лабораторної роботи були освоїні та опрацьовані навички у роботі з класами та їх об'ектами. Визначені типи зв'язків між класами та способи їх взаємодії між собою.