**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ**

**Національний технічний університет України**

**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Кафедра інформаційних систем та технологій**

**Звіт**

**з лабораторної  роботи № 1**

**«Класи та об’єкти. Конструктори та деструктори. Модифікатори.»**

**з дисципліни**

**«Програмування – 2»**

**Варіант №15**

|  |  |  |
| --- | --- | --- |
| **Перевірив:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |  | **Виконала:** |
|  |  | **Мірзоєва Гунел гр. ІС-13 , ФІОТ** |
| **Посада, ПІБ викладача** |  | **1 курс,**  **залікова книжка № IC-1235** |

**Київ 2021**

**Завдання.**

Створити клас с атрибутами та конструктором. У методі main() ініціалізувати створення екземплярів класу та продемонструвати роботу його методів згідно умов завдання.

15. Описати клас, що представляє трикутник. Передбачити методи для створення об'єктів, обчислення площі, периметра і точки перетину медіан. Описати властивості для отримання стану об'єкта.

**Виконання**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Text.Json;

using System.Threading.Tasks;

namespace lab1

{

[Serializable]

class Triangle

{

public Triangle()

{

}

public Triangle(double firstSide, double secondSide, double thirdSide)

{

if (firstSide <= 0 || secondSide <= 0 || thirdSide <= 0)

{

throw new IsNotTriangleException("attempt of creating triangle with side <= 0");

}

if (firstSide + secondSide <= thirdSide

|| firstSide + thirdSide <= secondSide

|| secondSide + thirdSide <= firstSide)

{

throw new IsNotTriangleException("One of the side more than sum of two others");

}

this.\_FirstSide = firstSide;

this.\_SecondSide = secondSide;

this.\_ThirdSide = thirdSide;

}

public double \_FirstSide { get; set; }

public double \_SecondSide { get; set; }

public double \_ThirdSide { get; set; }

public double calcPerimeter()

{

return \_FirstSide + \_SecondSide + \_ThirdSide;

}

public double calcArea()

{

double result = 0.25 \* Math.Sqrt((\_FirstSide + \_SecondSide + \_ThirdSide)

\* (\_FirstSide + \_SecondSide - \_ThirdSide)

\* (\_FirstSide + \_ThirdSide - \_SecondSide)

\* (\_SecondSide + \_ThirdSide - \_FirstSide));

return result;

}

public void calcMediansIntersectionPoint(out double x, out double y)

{

x = 0;

y = 0;

}

override public String ToString()

{

String triangleInfo = "Треугольник со сторонами " + \_FirstSide

+ ", " + \_SecondSide + " и " + \_ThirdSide;

return triangleInfo;

}

}

class JSON

{

public static void serealize(Triangle triangle)

{

string fileName = "triangle.json";

string jsonString = JsonSerializer.Serialize(triangle);

File.WriteAllText(fileName, jsonString);

Console.WriteLine("Інформація записана у файл");

Console.WriteLine(File.ReadAllText(fileName));

}

public static void deserealize()

{

string fileName = "triangle.json";

string jsonString = File.ReadAllText(fileName);

Triangle triangle = JsonSerializer.Deserialize<Triangle>(jsonString)!;

Console.WriteLine("Об'єкт з JSON файлу створений");

}

}

class IsNotTriangleException : Exception

{

public IsNotTriangleException(String message)

{

\_Message = message;

}

private String \_Message;

public override string ToString()

{

return \_Message;

}

}

internal class Program

{

static void Main(string[] args)

{

Random random = new Random();

for (int counter = 0; counter < 10; counter++)

{

Triangle triangle;

try

{

triangle = new Triangle(random.Next(3, 10), random.Next(-1, 15), random.Next(3, 10));

Console.WriteLine(triangle);

Console.WriteLine("Периметр треугольника = " + triangle.calcPerimeter().ToString("F"));

Console.WriteLine("Площадь треугольника = " + triangle.calcArea().ToString("F"));

Console.WriteLine();

}

catch (IsNotTriangleException exception)

{

Console.WriteLine(exception);

Console.WriteLine();

}

catch (Exception exception)

{

Console.WriteLine("Something is really wrong " + exception);

Console.WriteLine();

}

}

//JSON

Triangle triangle1 = new Triangle(random.Next(3, 10), random.Next(-1, 15), random.Next(3, 10));

JSON.serealize(triangle1);

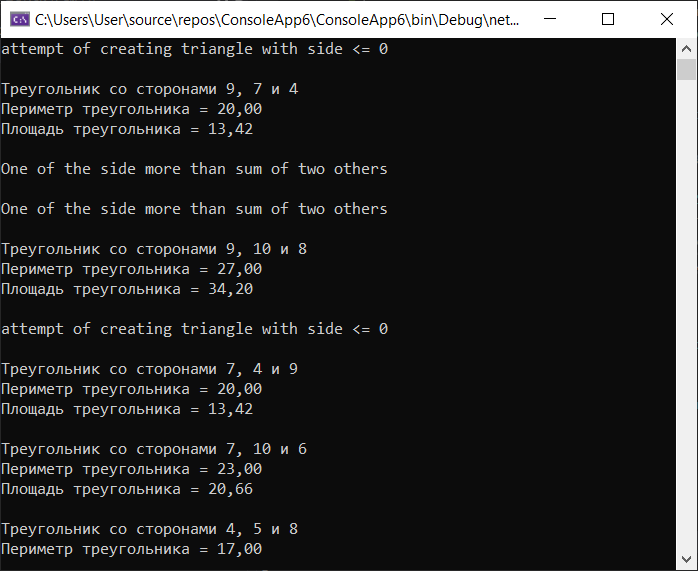
JSON.deserealize();

Console.ReadLine();

}

}

}



**Завдання 2**

Створити у попередньому завданні два методи з використанням серіалізації та десеріалізації JSON.

* **Метод 1**. Зберігає створений об’єкт класу з Завдання 1 у JSON файл
* **Метод 2.** Відкриває JSON файл з даними та створює об’єкт класу з цими даними для виконання Завдання 1.

class JSON

{

public static void serealize(Triangle triangle)

{

string fileName = "triangle.json";

string jsonString = JsonSerializer.Serialize(triangle);

File.WriteAllText(fileName, jsonString);

Console.WriteLine("Інформація записана у файл");

Console.WriteLine(File.ReadAllText(fileName));

}

public static void deserealize()

{

string fileName = "triangle.json";

string jsonString = File.ReadAllText(fileName);

Triangle triangle = JsonSerializer.Deserialize<Triangle>(jsonString)!;

Console.WriteLine("Об'єкт з JSON файлу створений");

}

