МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

Національний технічний університет України

Факультет інформатики та обчислювальної техніки

«Київський політехнічний інститут імені Ігоря Сікорського»

Кафедра інформаційних систем та технологій

Звіт

з лабораторної роботи № 1

з дисципліни

«Програмування – 2»

Варіант № 30

|  |  |  |
| --- | --- | --- |
| Перевірив: |  | Виконала: Шелест Поліна |
|  |  | Студентка гр. ІС-11 , ФІОТ |
|  |  | 1 курс |

Київ 2021

**Лабораторна робота № 1**

**Тема:** Класи та об’єкти. Конструктори та деструктори. Модифікатори

**Завдання 1:** Створити клас с атрибутами та конструктором. У методі main() ініціалізувати створення екземплярів класу та продемонструвати роботу його методів згідно умов завдання.

1. Скласти опис класу для вектору. Зберігає координати кінця вектора в n-вимірному просторі, який виходить з початку координат. Методи: збільшення кількості координат, зменшення кількості координат, сума двох векторів, добуток двох векторів, різниця двох векторів, множення вектора на скаляр, довжина вектора.

**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.

Код (класи):

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.Json;

using System.Text.Json.Serialization;

using System.IO;

namespace Lab1

{

public class Vector

{

public int Dimension { get; set; }

public decimal[] Cords { get; set; }

[JsonConstructor]

public Vector(int Dimension, decimal[] Cords)

{

this.Dimension= Dimension ;

this.Cords = Cords;

}

public Vector MinusVector(Vector Vecto) //віднімання векторів

{

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

Cords[i] = Vecto.Cords[i] - Cords[i];

return this;

}

public Vector multConst(decimal Nconst) //множення вектора на скаляр

{

decimal[] Cordinates = new decimal[Dimension];

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

Cordinates[i] = Cords[i] \* Nconst;

return new Vector(Dimension, Cordinates);

}

public Vector incCords() //збільшення кіькості координат

{

Dimension++;

decimal[] Cordinates = new decimal[Dimension];

for (int i = 0; i < Dimension - 1; i++)

Cordinates[i] = Cords[i];

Cordinates[Dimension - 1] = 0;

return new Vector(Dimension, Cordinates);

}

public Vector decCords() //зменшення кількості координат

{

Dimension--;

decimal[] Cordinates = new decimal[Dimension];

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

Cordinates[i] = Cords[i];

return new Vector(Dimension, Cordinates);

}

public decimal MultiVector(Vector Vector1) //множення векторів

{

decimal Multirez = 0;

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

{

Multirez += Vector1.Cords[i] \* Cords[i];

}

return Multirez;

}

public decimal lengthVector()//довжина вектора

{

decimal rez = 0;

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

{

rez += Cords[i] \* Cords[i];

}

return (decimal)Math.Sqrt((double)rez);

}

public void SaveVector()//зберегти джсон

{

string fileName = "Vector.json";

string jsonString = JsonSerializer.Serialize(this);

File.WriteAllText(fileName, jsonString);

}

public Vector LoadVector()//завантажити джсон

{

string fileName = "Vector.json";

string jsonString = File.ReadAllText(fileName);

Vector Vector = JsonSerializer.Deserialize<Vector>(jsonString);

return Vector;

}

}

}