

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 5

Виконав студент ІП-13 Вальчишен Ярослав Олександрович
(шифр, прізвище, ім'я, по батькові)

Перевірив Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота

Класи та об'єкти

Варіант 5

5. Розробити клас "кубічний многочлен" (многочлен виду $a_3x^3 + a_2x^2 + a_1x + a_0$), який заданий своїми коефіцієнтами. Створити масив об'єктів даного класу. Ви-

8

значити многочлен, який приймає найменше мінімальне значення на введеному користувачем відрізку $[a, b]$ із заданою точністю ϵ (використати метод простого перебору).

Код програми

C#

```
namespace ProgLab3
{
    class Program
    {
        static void Main(string[] args)
        {
            Input input = new Input();

            CubicPolynomial[] cubicPolynomials = input.GetPolynomials(input.GetPolynomialsAmount());
            float[] segment = input.GetSegment();

            PolynomialManager polynomialManager = new PolynomialManager(segment[0], segment[1], input.GetEpsilon());

            Console.WriteLine("Result: " + polynomialManager.GetMin(cubicPolynomials));
        }
    }
}
```

```

3 namespace ProgLab3
4 {
5     public class CubicPolynomial
6     {
7         private float _a0, _a1, _a2, _a3;
8
9         public CubicPolynomial(float a0, float a1, float a2, float a3)
10        {
11            _a0 = a0;
12            _a1 = a1;
13            _a2 = a2;
14            _a3 = a3;
15        }
16
17        public float GetValue(float x)
18        {
19            return (float)(_a3 * Math.Pow(x, 3) + _a2 * Math.Pow(x, 2) + _a1 * x + _a0);
20        }
21
22        public string GetPolynomial()
23        {
24            return $"{_a3}x^3 + {_a2}x^2 + {_a1}x + {_a0}";
25        }
26    }
27
28    public class PolynomialManager
29    {
30        private float _a;
31        private float _b;
32        private float _epsilon;
33
34        public PolynomialManager(float a, float b, float epsilon)
35        {
36            _a = a;
37            _b = b;
38            _epsilon = epsilon;
39        }
40
41        public float GetMin(CubicPolynomial[] polynomials)
42        {
43            float minValue = GetMinPolynomialValue(polynomials[0]);
44
45            for (int i = 0; i < polynomials.Length; i++)
46            {
47                float minPolynomialValue = GetMinPolynomialValue(polynomials[i]);
48                Console.WriteLine($"Min value in {polynomials[i].GetPolynomial()} is {minPolynomialValue}\n");
49
50                if (minValue >= minPolynomialValue)
51                    minValue = minPolynomialValue;
52            }
53
54            return minValue;
55        }
56    }

```

```

private float GetMinPolynomialValue(CubicPolynomial polynomial)
{
    float minValue = polynomial.GetValue(_a);

    for (float i = _a; i <= _b; i += _epsilon)
    {
        float value = polynomial.GetValue(i);
        Console.WriteLine($"{{polynomial.GetPolynomial()}} with {{i}} value equals {{value}}");

        if (minValue >= value)
            minValue = value;
    }

    return minValue;
}

```

```

3 namespace ProgLab3
4 {
5     public class Input
6     {
7         public int GetPolynomialsAmount()
8         {
9             int amount = 0;
10
11             Console.WriteLine("Enter amount of polynomials: ");
12             string input = Console.ReadLine();
13
14             if (int.TryParse(input, out int result))
15             {
16                 amount = result;
17             }
18
19             return amount;
20         }
21     }
22
23     public CubicPolynomial[] GetPolynomials(int amount)
24     {
25         CubicPolynomial[] cubicPolynomials = new CubicPolynomial[amount];
26
27         Console.WriteLine("Enter polynomials coefficients (format: a3 a2 a1 a0): ");
28
29         for (int i = 0; i < cubicPolynomials.Length; i++)
30         {
31             string input = Console.ReadLine();
32
33             CubicPolynomial cubicPolynomial = CreatePolynomial(input);
34             if (cubicPolynomial != null)
35             {
36                 cubicPolynomials[i] = cubicPolynomial;
37                 Console.WriteLine(cubicPolynomials[i].GetPolynomial());
38             }
39             else
40             {
41                 Console.WriteLine("Error! Wrong format. Try again!");
42                 i--;
43             }
44         }
45
46         return cubicPolynomials;
47     }
48
49     public float[] GetSegment()
50     {
51         float[] segment = new float[2];
52         float result = 0;
53
54         Console.WriteLine("Enter a: ");
55         while (!float.TryParse(Console.ReadLine(), out result))
56         {
57             Console.WriteLine("Error! Wrong format. Try again!");
58         }
59
60         segment[0] = result;
61
62         Console.WriteLine("Enter b: ");
63         while (!float.TryParse(Console.ReadLine(), out result))
64         {
65             Console.WriteLine("Error! Wrong format. Try again!");
66         }
67
68         segment[1] = result;
69
70         if (segment[0] > segment[1])
71         {
72             Console.WriteLine("a and b will be swapped, since a > b");
73             float temp = segment[0];
74             segment[0] = segment[1];
75             segment[1] = temp;
76         }
77
78         return segment;
79     }
80
81     public float GetEpsilon()
82     {
83         Console.WriteLine("Enter epsilon: ");
84         float result = 0;
85
86         while (!float.TryParse(Console.ReadLine(), out result))
87         {
88             Console.WriteLine("Error! Wrong format. Try again!");
89         }
90
91         return result;
92     }
93
94     private CubicPolynomial CreatePolynomial(string line)
95     {
96         float[] coefficients = new float[4];
97         string[] elements = line.Split(" ");
98
99         if (elements.Length != 4)
100             return null;
101         for (int i = 0; i < elements.Length; i++)
102             if (float.TryParse(elements[i], out float result))
103                 coefficients[i] = result;
104             else
105                 return null;
106
107         return new CubicPolynomial(coefficients[3], coefficients[2], coefficients[1], coefficients[0]);
108     }
109 }
110
111 }
112
113 }
114

```


Тестування:

```
Enter amount of polynomials:
2
Enter polynomials coefficients (format: a3 a2 a1 a0):
1 2 3 -2
Error! Wrong format. Try again
-2 23 4 5
-2x^3 + 23x^2 + 4x + 5
1 -2 23 5
1x^3 + -2x^2 + 23x + 5
Enter a:
1
Enter b:
2
Enter epsilon:
0.1
Error! Wrong format. Try again!
0.1
-2x^3 + 23x^2 + 4x + 5 with 1 value equals 30
-2x^3 + 23x^2 + 4x + 5 with 1.1 value equals 34.568
-2x^3 + 23x^2 + 4x + 5 with 1.2 value equals 39.464
-2x^3 + 23x^2 + 4x + 5 with 1.3000001 value equals 44.676003
-2x^3 + 23x^2 + 4x + 5 with 1.4000001 value equals 50.192005
-2x^3 + 23x^2 + 4x + 5 with 1.5000001 value equals 56.000008
-2x^3 + 23x^2 + 4x + 5 with 1.6000001 value equals 62.08801
-2x^3 + 23x^2 + 4x + 5 with 1.7000002 value equals 68.44401
-2x^3 + 23x^2 + 4x + 5 with 1.8000002 value equals 75.056015
-2x^3 + 23x^2 + 4x + 5 with 1.9000002 value equals 81.91202
-2x^3 + 23x^2 + 4x + 5 with 1 value equals 30
-2x^3 + 23x^2 + 4x + 5 with 1.1 value equals 34.568
-2x^3 + 23x^2 + 4x + 5 with 1.2 value equals 39.464
-2x^3 + 23x^2 + 4x + 5 with 1.3000001 value equals 44.676003
-2x^3 + 23x^2 + 4x + 5 with 1.4000001 value equals 50.192005
-2x^3 + 23x^2 + 4x + 5 with 1.5000001 value equals 56.000008
-2x^3 + 23x^2 + 4x + 5 with 1.6000001 value equals 62.08801
-2x^3 + 23x^2 + 4x + 5 with 1.7000002 value equals 68.44401
-2x^3 + 23x^2 + 4x + 5 with 1.8000002 value equals 75.056015
-2x^3 + 23x^2 + 4x + 5 with 1.9000002 value equals 81.91202
Min value in -2x^3 + 23x^2 + 4x + 5 is 30

1x^3 + -2x^2 + 23x + 5 with 1 value equals 27
1x^3 + -2x^2 + 23x + 5 with 1.1 value equals 29.211
1x^3 + -2x^2 + 23x + 5 with 1.2 value equals 31.448
1x^3 + -2x^2 + 23x + 5 with 1.3000001 value equals 33.717003
1x^3 + -2x^2 + 23x + 5 with 1.4000001 value equals 36.024002
1x^3 + -2x^2 + 23x + 5 with 1.5000001 value equals 38.375004
1x^3 + -2x^2 + 23x + 5 with 1.6000001 value equals 40.776005
1x^3 + -2x^2 + 23x + 5 with 1.7000002 value equals 43.233
1x^3 + -2x^2 + 23x + 5 with 1.8000002 value equals 45.752007
1x^3 + -2x^2 + 23x + 5 with 1.9000002 value equals 48.339005
Min value in 1x^3 + -2x^2 + 23x + 5 is 27

Result: 27
```

Висновки:

Виконуючи лабораторну роботу, я застосував на практиці знання щодо створення та роботи з класами та об'єктами й побачив їх програмну реалізацію у мові програмування C#.