Deakin University

Object Oriented Development

OnTrack Submission

MyPolynomial class

Submitted By:
Romil Bijarnia
s222528574
2024/08/20 02:26

Tutor:
Jotham Barazani

Outcome	**Weight**
Evaluate Code	♦♦♦◊◊
Principles	♦♦♦◊◊
Build Programs	♦♦♦◊◊
Design	♦♦♦◊◊
Justify	♦♦♦◊◊

nice assignment

August 20, 2024

Produced by Doubtfire

```
using System;
using System.Text;

public class MyPolynomial
{
    private double[] _coeffs;

    public MyPolynomial(double[] coeffs)
    {
        _coeffs = coeffs;
    }

    public int GetDegree()
    {
        return _coeffs.Length - 1;
    }

    public override string ToString()
    {
        StringBuilder sb = new StringBuilder();
        for (int i = _coeffs.Length - 1; i >= 0; i--)
        {
            if (_coeffs[i] != 0)
            {
                if (sb.Length > 0 && _coeffs[i] > 0)
                    sb.Append(" + ");

                if (i == 0)
                    sb.Append(_coeffs[i]);
                else if (i == 1)
                    sb.Append($"{_coeffs[i]}x");
                else
                    sb.Append($"{_coeffs[i]}x^{i}");
            }
        }
        return sb.ToString();
    }

    public double Evaluate(double x)
    {
        double result = 0;
        for (int i = 0; i < _coeffs.Length; i++)
        {
            result += _coeffs[i] * Math.Pow(x, i);
        }
        return result;
    }

    public MyPolynomial Add(MyPolynomial another)
    {
        int maxDegree = Math.Max(this.GetDegree(), another.GetDegree());
        double[] resultCoeffs = new double[maxDegree + 1];

        for (int i = 0; i <= maxDegree; i++)
        {
            double coeff1 = (i <= this.GetDegree()) ? this._coeffs[i] : 0;
            double coeff2 = (i <= another.GetDegree()) ? another._coeffs[i] : 0;
            resultCoeffs[i] = coeff1 + coeff2;
        }

        return new MyPolynomial(resultCoeffs);
    }

    public MyPolynomial Multiply(MyPolynomial another)
    {
        int newDegree = this.GetDegree() + another.GetDegree();
        double[] resultCoeffs = new double[newDegree + 1];

        for (int i = 0; i <= this.GetDegree(); i++)
        {
            for (int j = 0; j <= another.GetDegree(); j++)
            {
                resultCoeffs[i + j] += this._coeffs[i] * another._coeffs[j];
            }
        }

        return new MyPolynomial(resultCoeffs);
    }
}
```

```
using System;

public class TestMyPolynomial
{
    public static void Main(string[] args)
    {
        double[] coeffs1 = { 11, -4, 3 }; // Represents 3x^2 - 4x + 11
        MyPolynomial poly1 = new MyPolynomial(coeffs1);

        Console.WriteLine("Polynomial 1: " + poly1.ToString());
        Console.WriteLine("Degree of Polynomial 1: " + poly1.GetDegree());

        double[] coeffs2 = { 1, 0, 2 }; // Represents 2x^2 + 1
        MyPolynomial poly2 = new MyPolynomial(coeffs2);

        Console.WriteLine("Polynomial 2: " + poly2.ToString());
        Console.WriteLine("Sum: " + poly1.Add(poly2).ToString());
        Console.WriteLine("Product: " + poly1.Multiply(poly2).ToString());
        Console.WriteLine("Value of Polynomial 1 at x=2: " + poly1.Evaluate(2));
    }
}
```