

|  |
| --- |
| LAB Assignment 01 |
|  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Presented to: | Muhammad Aksam Iftikhar | | **Course:** | Object Oriented Programming | | **Presented by:** | Asad Ali | | **Class:** | SP21-BCS-A | | **Registration ID:** | SP21-BCS-007 | | COMSATS  University Islamabad Lahore Campus |



**Polynomial Class**

import java.util.ArrayList;

public class Polynomial {

    private int degree;

    private ArrayList<Double> coefficients;

    public Polynomial(int degree) {

        setDegree(degree);

        setCoefficient();

    }

    public Polynomial() {

        setDegree(0);

        setCoefficient();

    }

    public Polynomial(int degree, double... coefficients) {

        setDegree(degree);

        if (coefficients.length == degree) {

            setCoefficient(coefficients);

        } else {

            System.out.println("\nYour given list of coefficients is not matching the degree of polynomial. So all coefficients are set to '0'........! \n");

            setCoefficient();

        }

    }

    private void setCoefficient(double[] coefficients) {//Helper Method used by Constructor

        this.coefficients = new ArrayList<Double>();

        for (int i = 0; i < getDegree(); i++) {

            this.coefficients.add(coefficients[i]);

        }

    }

    private void setCoefficient() {//Helper Method used by Constructor

        this.coefficients = new ArrayList<Double>();

        for (int i = 0; i < getDegree(); i++) {

            this.coefficients.add(0.0);

        }

    }

    public int getDegree() {

        return degree;

    }

    public void setDegree(int degree) {

        this.degree = degree;

    }

    public ArrayList<Double> getCoefficients() {

        return coefficients;

    }

    public void setCoefficients(int i, double coefficients) {

        this.coefficients.add(coefficients);

    }

    public String toString() {

        String string = "P(x) = ";

        for (int i = 0; i < this.coefficients.size(); i++) {

            String string1 = coefficients.get(i) + "x^" + (this.coefficients.size() - (i + 1));

            string = string.concat(string1);

            if (i < this.coefficients.size() - 1) {

                String string2 = " + ";

                string = string.concat(string2);

            }

        }

        return string;

    }

    public double evaluatePolynomial(int x) {

        double result = 0.0;

        for (int i = 0; i < getDegree(); i++) {

            result += this.coefficients.get(i) \* Math.pow(x, i);

        }

        return result;

    }

    public void addPolynomial(Polynomial polynomialToBeAdded) {

        if (getDegree() == polynomialToBeAdded.getDegree()) {

            for (int i = 0; i < this.coefficients.size(); i++) {

                this.coefficients.set(i, this.coefficients.get(i) + polynomialToBeAdded.coefficients.get(i));

            }

        } else if (getDegree() > polynomialToBeAdded.getDegree()) {

            for (int i = 0; i < this.coefficients.size(); i++) {

                if (i < polynomialToBeAdded.getDegree())

                    this.coefficients.set(i, this.coefficients.get(i) + polynomialToBeAdded.coefficients.get(i));

            }

        } else if (getDegree() < polynomialToBeAdded.getDegree()) {

            for (int i = 0; i < polynomialToBeAdded.getDegree(); i++) {

                if (i < getDegree())

                    this.coefficients.set(i, this.coefficients.get(i) + polynomialToBeAdded.coefficients.get(i));

                else {

                    this.coefficients.add(polynomialToBeAdded.coefficients.get(i));

                    this.setDegree(getDegree() + 1);

                }

            }

        }

    }

    public static Polynomial addPolynomial(Polynomial polyn1, Polynomial polyn2) {

        Polynomial resultPolyn = new Polynomial();

        if (polyn1.getDegree() == polyn2.getDegree()) {

            resultPolyn = new Polynomial(polyn1.getDegree());

            for (int i = 0; i < polyn1.coefficients.size(); i++) {

                resultPolyn.coefficients.set(i, polyn1.coefficients.get(i) + polyn2.coefficients.get(i));

            }

        } else if (polyn1.getDegree() > polyn2.getDegree()) {

            resultPolyn = new Polynomial(polyn1.getDegree());

            for (int i = 0; i < polyn1.coefficients.size(); i++) {

                if (i < polyn2.getDegree())

                    resultPolyn.coefficients.set(i, polyn1.coefficients.get(i) + polyn2.coefficients.get(i));

                else

                    resultPolyn.coefficients.set(i, polyn1.coefficients.get(i));

            }

        } else if (polyn1.getDegree() < polyn2.getDegree()) {

            resultPolyn = new Polynomial(polyn2.getDegree());

            for (int i = 0; i < polyn2.coefficients.size(); i++) {

                if (i < polyn1.getDegree())

                    resultPolyn.coefficients.set(i, polyn1.coefficients.get(i) + polyn2.coefficients.get(i));

                else

                    resultPolyn.coefficients.set(i, polyn2.coefficients.get(i));

            }

        }

        return resultPolyn;

    }

}

**PolynomialTest Class**

public class PolynomialTest {

    public static void main(String[] args) {

        Polynomial polynomial1 = new Polynomial(4, 2, 3, 4, 3);

        Polynomial polynomial2 = new Polynomial(3, 6, 1, 5);

        System.out.println();

        System.out.println();

        System.out.println(polynomial1);

        System.out.println(polynomial2);

        System.out.println();

        System.out.println();

        System.out.println("Answer of Polynomial " + polynomial1 + " for x = 2 is: " + polynomial1.evaluatePolynomial(2));

        System.out.println("Answer of Polynomial " + polynomial2 + " for x = 2 is: " + polynomial2.evaluatePolynomial(2));

        System.out.println();

        System.out.println();

        System.out.println("Sum of Polynomial by Method Call on an Object (Different Degree): ");

        System.out.print("Sum of " + polynomial1 + "\n" +

                         "     + " + polynomial2 + "\n");

        polynomial1.addPolynomial(polynomial2);

        System.out.println("  is = " + polynomial1);

        System.out.println();

        System.out.println("Sum of Polynomial by Static Method Call using Class Name (Same Degree): ");

        System.out.print("Sum of " + polynomial1 + "\n" +

                         "     + " + polynomial2 + "\n");

        Polynomial resultPolinomal = Polynomial.addPolynomial(polynomial1, polynomial2);

        System.out.println("  is = " + resultPolinomal);

        System.out.println();

    }

}

**UML Diagram**

|  |
| --- |
| Polynomial |
| * degree: int |
| * coefficients: ArrayList<Double> |
| <<constructor>> Poynimial (degree: int) |
| <<constructor>> Poynimial () |
| <<constructor>> Poynimial (degree: int, coefficients: double…) |
| - setCoefficients (coefficients: double[]): void |
| - setCoefficients (): void |
| + getDegree (): int |
| + getCoefficients (): ArrayList<Double> |
| + setDegree (degree: int): void |
| + setCoefficients (i: int, coefficients: double): void |
| + toString (): String |
| + evaluatePolynomial (x: int): double |
| + addPolynomial (polynomialToBeAdded: Polynomial): void |
| + addPolynomial (polyn1: Polynomial, polyn2: Polynomial): Polynomial |