

**Lösung der Programmieraufgabe  
als Ersatz für den Java-Kompaktkurs  
im April 2000 von Michael Wack  
Matrikelnummer: 2234088  
Email: MichaelWack@gmx.de**

## Programmbeschreibung

Die Klasse **Polygon** arbeitet, wie in der Aufgabenstellung verlangt, und ist in der Lage mit der Methode **plot** einen beliebigen Polygonzug (auch mit negativen Koordinaten) auf die maximale Grösse skaliert zu zeichnen, dazu wird die Größe des Applets in **Dim** übergeben. Dadurch kann es natürlich zu Verzerrungen kommen, aber man erhält zumindest immer ein schönes Bild.

**Funct** implementiert lediglich die Methode **plot** und erzeugt darin eine Instanz von **Polygon** mit **n+1** Punkten, die anschliessend gezeichnet werden kann.

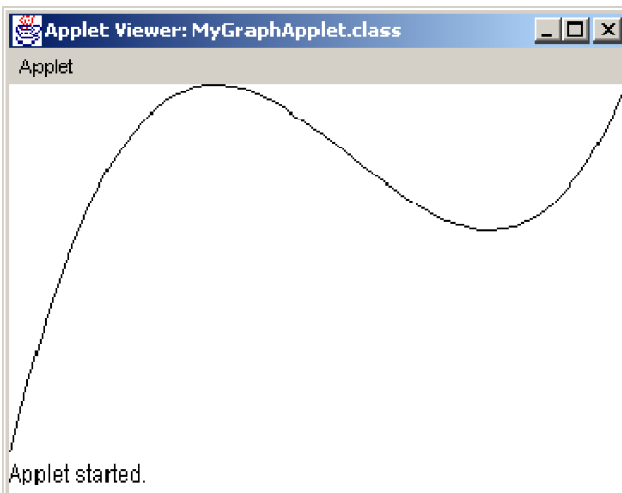
Die Klasse **Polynom** ist von **Funct** abgeleitet und implementiert die Methode **eval** mit der sich die Funktionswerte eines Polynoms mit den Koeffizienten im Array **Koeff** berechnen lassen.

Die Klasse **MyGraphApplet** erzeugt in der Methode **init** ein neues Polynom und daraus ein **Polygon**, das sich schliesslich in **paint** selber zeichnet.

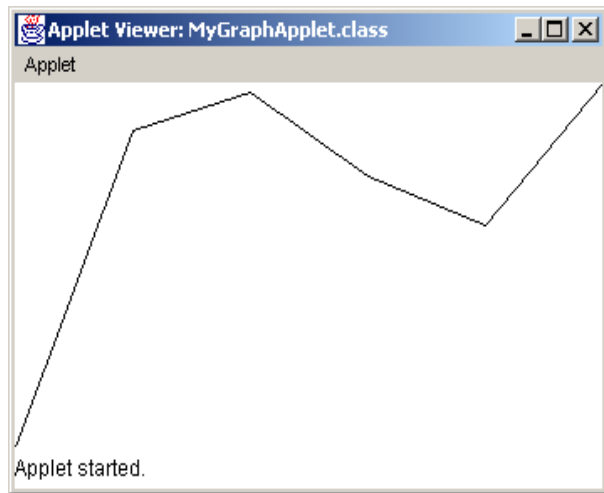
## Testläufe

Hier folgen nun einige Beispiele für den Programmablauf. Jeweils mit Angabe der Funktion und der Anzahl der Teilintervalle.

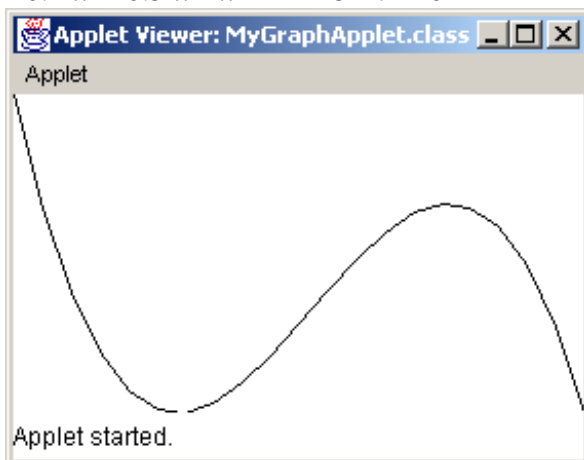
$$2 \cdot x^3 - 4 \cdot x^2 \quad x = -1 \dots 2 \quad n = 50$$



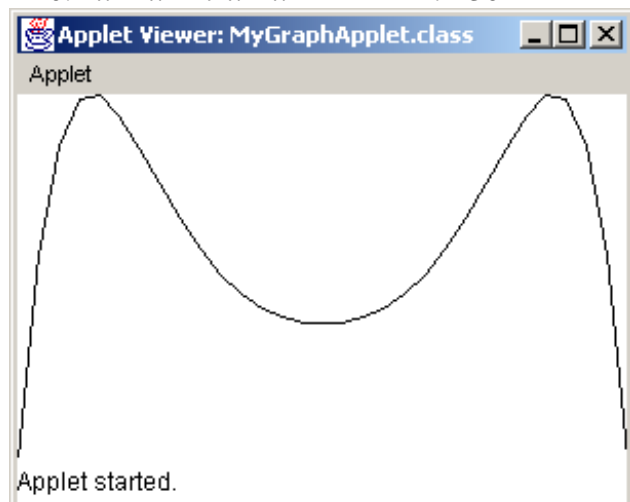
$$2 \cdot x^3 - 4 \cdot x^2 \quad x = -1 \dots 2 \quad n = 5$$



$$-0.1 \cdot x^3 + 0.5 \cdot x^2 \quad x = -2 \dots 5 \quad n = 20$$



$$-0.1 \cdot x^6 + x^4 + 7 \cdot x^2 \quad x = -4 \dots 4 \quad n = 30$$



## Sourcen

```
// Polygon
// (c) M.Wack 2000
// 51 lines

public class Polygon extends Object {

    public double[] m_x; // X - Koordinaten
    public double[] m_y; // Y - Koordinaten

    /* Konstruktor */
    public Polygon(double[] x, double[] y) {
        m_x = x;
        m_y = y;
    }

    public void plot( java.awt.Graphics g, java.awt.Dimension Dim) {
        int[] px = new int[ m_x.length];
        int[] py = new int[ m_x.length];

        for( int c = 0; c < m_x.length; c++) {
            px[ c] = (int)java.lang.Math.round((m_x[ c] - MinX()) * Dim.width / (MaxX() - MinX()));
            py[ c] = (int)java.lang.Math.round((m_y[ c] - MinY()) * Dim.height / (MaxY() - MinY()));
        }

        g.drawPolyline( px, py, px.length); // Das skalierte Polygon zeichnen
    }

    public double MaxX() {
        return m_x[ m_x.length - 1]; // den grössten x-Wert zurückgeben
    }

    public double MinX() {
        return m_x[ 0];
    }

    public double MaxY() {
        double my = m_y[ 0];
        for( int c = 1; c < m_x.length; c++) {
            if( my > m_y[ c]) my = m_y[ c];
        }
        return my; // den grössten y-Wert zurückgeben
    }

    public double MinY() {
        double my = m_y[ 0];
        for( int c = 1; c < m_x.length; c++) {
            if( my < m_y[ c]) my = m_y[ c];
        }
        return my; // den kleinsten y-Wert zurückgeben
    }
}
```

```

// Funct
// (c) M.Wack 2000
// 26 lines

abstract class Funct extends Object {

    public Funct() {
    }

    // Funktionswert berechnen
    abstract double eval( double x);

    // Polygon aus n Funktionswerten erstellen
    public Polygon plot( double x0, double x1, int n) {
        double[] x = new double[n+1];
        double[] y = new double[n+1];

        for( int c = 0; c <= n; c++) {
            x[ c] = x0 + (x1 - x0) * c/n;
            y[ c] = eval( x[c]);
        }

        Polygon APolygon = new Polygon( x, y);
        return APolygon;
    }
}

```

```
// Polynom
// (c) M. Wack 2000
// 22 lines

public class Polynom extends Funct {
    private double[] m_Koeff; // Koeffizienten des Polynoms

    // Konstruktor
    public Polynom( double[] Koeff) {
        m_Koeff = Koeff;
    }

    // Funktionswert berechnen
    public double eval( double x) {
        int k;
        double e=0;
        for( k=0; k < m_Koeff.length; k++) {
            e = e + java.lang.Math.pow( x, k) * m_Koeff[ k];
        }
        return e;
    }
}
```

```

// MyGraphApplet
// (c) M. Wack 2000
// 24 lines

import java.applet.*;
import java.awt.*;

public class MyGraphApplet extends java.applet.Applet {

    Polygon APolygon;
    public void init () {

        // kleinste Potenz von x zuerst
        // {1,2,3} bedeutet 3*x^2 + 2*x^1 + 1*x^0
        double[] Koeff = { 0, 0, 7, 0, 1, 0, -1};
        Polynom APolynom = new Polynom( Koeff);
        APolygon = APolynom.plot( -4, 4, 30);
    }

    public void paint( Graphics g) {
        Dimension Dim = new Dimension();
        APolygon.plot( g, getSize( Dim));
    }
}

```