Inhaltsverzeichnis

1.1 Die Klasse Punkt

a und b sind Objekte vom Typ Punkt. Z.B.:

```
Punkt a = new Punkt(-50,10);
Punkt b = new Punkt(20,40);Kochkurve
```

allgemein

```
double x = a.abstand(b);
```

für Sierpinski Dreieck

```
Punkt m = a.mitte(b);
```

für Kochkurve

```
Punkt p1 = a.erstesDrittel(b);
Punkt p2 = a.spitze(b);
Punkt p3 = a.zweitesDrittel(b);
```

geheInRichtung-Methoden

Gehe von a aus das erste Viertel in Richtung b:

```
Punkt t1 = a.geheInRichtung(b, 0.25);
```

Gehe von a aus in x-Richtung die Hälfte und in y-Richtung die ganze Strecke in Richtung b:

```
Punkt t2 = a.geheInRichtung(b, 0.5, 1);
```

Gehe von a aus im 90° Winkel 50 Einheiten weit:

```
Punkt t3 = a.geheInRichtung(90, 50);
```

1.2 Aufteilung in verschiedene Klassen

Erhöht die Übersichtlichkeit und erleichtert es, mehrere Objekte des gleichen Typs zu erzeugen.

Die Klasse Main

```
private Sierpinski dreieck1, dreieck2;
private Kochkurve koch;

public void start() {
    koch = new Kochkurve(-50, 10, 90, 90, "rot");
    dreieck1 = new Sierpinski(-100, -100, 50, -100, -50,
       ↪ 10, "blau");
    dreieck2 = new Sierpinski(50, -100, 100, -100, 90,
       ↪ 90, "lila");
}
public void zeichnen() {
    koch.zeichnen();
    dreieck1.zeichnen();
    dreieck2.zeichnen();
}
public void taste(int tastencode) {
    if(tastencode == '+') {
        koch.kleiner();
        dreieck1.kleiner();
        dreieck2.kleiner();
        repaint();
    }
}
```

Die Klasse Kochkurve

```
private double minlänge;
private Punkt p1, p2;
private String farbe;

public Kochkurve(double x1, double y1, double x2, double
   ↪  y2, String farbe) {
    p1 = new Punkt (x1,y1);
    p2 = new Punkt (x2,y2);
    this.farbe = farbe;
    minlänge = 1.1*p1.abstand(p2);
}
private void zeichneLinie(Punkt a, Punkt b) {
    if(a.abstand(b) >= minlänge) {
        Punkt p = a.erstesDrittel(b);
        ...
        zeichneLinie(a, p);
        ...
    } else {
        B.linie(farbe, a, b);
    }
}
public void zeichnen() {
    zeichneLinie(p1, p2);
}
public void kleiner() {
    minlänge /= 3;
}
```

Die Klasse Sierpinski

Die Klasse Sierpinski sieht genauso aus. Im Erzeuger werden die Eckpunkte und die Farbe gespeichert und minlänge auf einen geeigneten Startwert gesetzt. die Methode zeichnen() muss das Dreieck zeichnen.

1.3 Teilbilder erzeugen, speichern und einfügen

Beispiel: drehendes Sierpinskidreieck

Die Klasse Main

```
private Sierpinski dreieck;
private Image img;
private double alpha;

public void start() {
    dreieck = new Sierpinski();
    img = B.erzeugeBild(800, 800, dreieck);
    alpha = 0;
}
public void zeichnen() {
    B.bild(alpha, img, 0, 0);
}
public void metronom() {
    alpha += 1;
    repaint();
}
```

Die Klasse Sierpinski

```
private double minlänge = 2;

private void zeichneDreieck(Punkt a, Punkt b, Punkt c) {
    if(a.abstand(b) >= minlänge) {
        Punkt ab = a.mitte(b);
        Punkt bc = b.mitte(c);
        Punkt ac = a.mitte(c);
        zeichneDreieck(a, ab, ac);
        zeichneDreieck(b, ab, bc);
        zeichneDreieck(c, ac, bc);
    } else {
        B.vieleck("blau", a, b, c);
    }
}
public void zeichnen() {
    Punkt o = new Punkt(0, 0);
    Punkt a = o.geheInRichtung(-30, 100);
    Punkt b = o.geheInRichtung(90, 100);
    Punkt c = o.geheInRichtung(210, 100);
    zeichneDreieck(a, b, c);
}
```