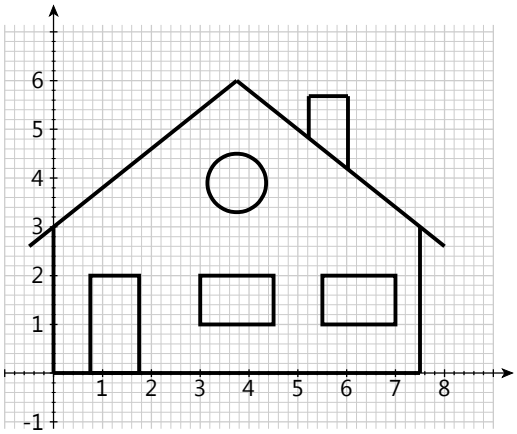
Aufgabenblatt 3

Haus



Vorbereitung

In dieser Aufgabe soll eine Grafik programmiert werden. Erzeugen Sie dazu zunächst die beiden folgenden Klassen *HausTVG* und *Haus*. Starten Sie die Klasse *Haus*. Dadurch wird der Physolator gestartet und das physikalische System *Haus* in den Physolator geladen zusammen mit der Grafikkomponente *HausTVG*. Es erscheint eine leere Grafik mit einem Koordinatensystem.

```
import de.physolator.usr.tvg.TVG;

public class HausTVG extends TVG {

    public HausTVG() {
        geometry.setUserArea(-1, 9, -1, 7);
        geometry.setRim(30, 30, 30, 30);
        scalesStyle.visible = true;
    }

    public void paint() {
        style.useUCS = true;
        style.strokeWidth = 3;
        //  Platz für die Zeichenbefehle
    }
}
```

```
import de.physolator.usr.*;

public class Haus extends PhysicalSystem {

    public void initGraphicsComponents(GraphicsComponents g) {
        g.addTVG(new HausTVG());
    }

    public static void main (String args[]) {
        start();
    }
}
```

Erläuterungen

Die Klasse *HausTVG* ist eine Grafikkomponente. Die Klasse *Haus* ist ein physikalisches System, das lediglich dazu dient, die Klasse *HausTVG* in den Physolator zu laden und im Physolator anzuzeigen.

Im Konstruktor der Klasse *HausTVG* werden grundlegende Einstellungen für die Grafik gemacht. Mit dem Befehl *geometry.setUserArea(-1, 9, -1, 7)* wird festgelegt, dass ein Koordinatensystem verwendet werden soll, das in x-Richtung von -1 bis 9 und in y-Richtun von -1 bis 7 reichen soll. Mit dem Befehl *geometry.setRim(30, 30, 30, 30)* wird festgelegt, dass um die Grafik herum in alle vier Richtungen ein Rand von 30 Pixeln sein soll. Die Zuweisung *scalesStyle.visible=true;* legt fest, dass die Koordinatenachsen angezeigt werden sollen und die nachfolgenden Zuweisungen legen fest, dass auch Gitternetzlinien angezeigt werden sollen.

Die Methode *paint()* der Klasse *HausTVG* sollen die Zeichenbefehle an der markierten Stelle untergebracht werden. Die beiden Zuweisungen zu Beginn der Methode legen fest, wie die nachfolgenden wirken sollen. Mit der Zuweisung *style.useUCS=true;* wird festgelegt, dass die nachfolgenden Zeichenbefehle das im Konstruktor mit *setUserArea* definierte Benutzer-Koordinatensystem verwenden sollen und nicht das Pixelkoordinatensystem. Die Zuweisung *style.strokeWidth=3;* legt fest, dass die nachfolgenden Zeichenbefehle eine Linienbreite von 3 Pixeln verwenden sollen. Der Standardwert für die Linienbreite wäre 1. Linien mit eine Breite von einem Pixel sind vor dem Hintergrund schwerer zu erkennen.

1. Teilaufgabe

Das Haus in der obigen Abbildung soll auf den Bildschirm gezeichnet werden. Fügen Sie dazu *drawLine*- und *drawCircle*-Befehle an der markierten Stelle in die *paint*-Methode ein.

drawLine(x1,y1,x2,y2);	zeichnet eine Linie von (x1,y1) nach (x2,y2)
drawCircle(x,y,r);	zeichnet einen Kreis mit Mittelpunkt (x,y) und Radius r

Hinweis: Wenn Sie Änderungen im Programmcode von HausTVG vorgenommen haben, können Sie die Änderungen einfach dadurch übernehmen, dass Sie die Änderungen zunächst abspeichern und dann den Reload-Button  des Physolators drücken. Die Änderungen werden dadurch übernommen.

2. Teilaufgabe

Jetzt sollen mehrere Häuser an unterschiedlichen Stellen in die Zeichnung eingefügt werden. Fügen Sie dazu in die Klasse *HausTVG* eine Methode *zeichneHaus(dx,dy)* ein. Diese Methode soll ein Haus zeichnen, das im Vergleich zum Originalbild um *(dx,dy)* verschoben ist. Das Haus soll also so positioniert werden, dass die linke untere Ecke des Hauses an der Position *(dx,dy)* liegt.

```
private void zeichneHaus(double dx, double dy) {
    // Platz für Zeichenbefehle
}
```

Damit genügend Zeichenfläche für mehrere Häuser zur Verfügung, ändern Sie im Programmcode den Befehl *geometry.setUserArea(-1, 9, -1, 7);* zu *geometry.setUserArea(-1, 28, -1, 15);* .

3. Teilaufgabe

Im nächsten Schritt soll eine größere Anzahl von Häusern gezeichnet werden, die in einem Raster angeordnet sind: 8 Häuser in horizontaler Richtung und 6 Häuser in vertikaler Richtung. Insgesamt also 48 Häuser. Der Rasterabstand, also der Abstand der linken unteren Ecke eines Hauses zur linken unteren Ecke des Nachbarhauses soll 9 betragen.

Vergrößern Sie zunächst die Zeichenfläche in geeigneter Weise, indem Sie die Parameterwerte von *geometry.setUserArea(...)* anpassen.Tipp: Verwenden Sie zum Zeichnen der Häuser ein oder mehrere for-Schleifen.

4. Teilaufgabe

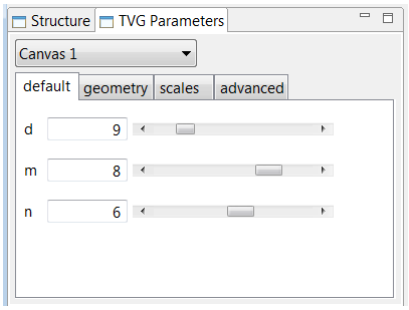
Die Anzahl der Häuser in horizontaler Richtung wird nun mit *m* bezeichnete, die Anzahl der Häuser in vertikaler Richtung mit *n* und der Rasterabstand mit *d*. In der vorangegangenen Teilaufgabe galt *m=8*, *n=6* und *d=9*. Während die Zeichnung dargestellt wird sollen diese Wert jetzt verändert werden können. Fügen Sie dazu oberhalb der *paint*-Methode den folgenden Programmcode mit Variablendeklarationen von *m*, *n* und *d* ein.

```
@Parameter
@Slider(min = 7, max = 20, step = 0.1, width = 200)
public double d = 9;

@Parameter
@Slider(min = 1, max = 10, width = 200)
public int m = 8;

@Parameter
@Slider(min = 1, max = 10, width = 200)
public int n = 6;
```

Ersetzen Sie nun in der *paint*-Methode die Zahlen 8, 6 und 9 durch die Variablen *m*, *n* und *d*. Nachdem die neue Grafik geladen wurde, gehen Sie mit der Maus in die Grafik und drücken Sie dort auf das Symbol  . Es erscheint folgender Dialog, in dem Sie die drei Variablen *m*, *n* und *d*. Sobald die Werte in diesem Dialog verändert werden, soll sich die Grafik entsprechend verändern.



Erläuterungen: Die Annotationen *@Parameter* und *@Slider* stehen vor einer Variablen. Die Annotation *@Parameter* bewirkt, dass die Variable in dem TVG-Parameter-Dialog angezeigt wird. Die Variable erscheint dann dort mit ihrem Namen und ihrem Zahlenwert. Der Zahlenwert steht in einem Textfeld und kann vom Benutzer verändert werden. Möchte man den Wert zusätzlich durch einen Slider (Schieberegler) verändern können, so benötigt man neben der *@Paramter*-Annotation eine *@Slider*-Annotation. Die *@Slider*-Annotation hat vier Parameter *min*, *max*, *step* und *width*. Die Werte *min* und *max* legen die Untergrenze und die Obergrenzen fest, also die Werte, die erreicht werden, wenn man den Slider ganz nach links beziehungsweise ganz nach rechts schiebt. Der Parameter step steht für die

Schrittweite. Er beschreibt, um wie viel sich der Wert ändern soll, wenn man auf die Pfeiltasten am linken und rechten Ende des Sliders drückt. Der Parameter *width* gibt an, wie breit der Slider auf dem Bildschirm sein soll. Diese Angabe erfolgt in Pixeln.