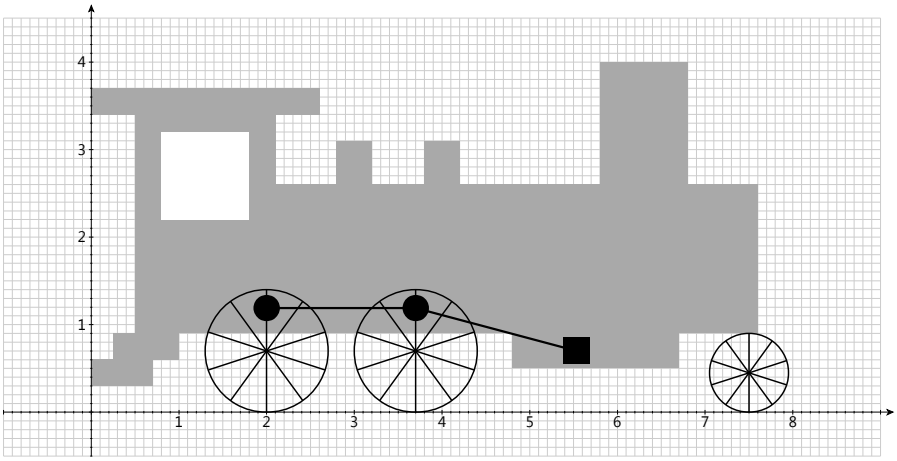
Aufgabenblatt 7

Lokomotive



Vorbereitung

In dieser Aufgabe soll eine Grafik programmiert werden. Erzeugen Sie analog zu Aufgabenblatt 3 zunächst die beiden folgenden Klassen *LokomotiveTVG* und *Lokomotive*. Laden Sie die Klasse *Lokomotive* in den Physolator.

```
import static de.physolator.usr.tvg.Shape.*;
import static java.lang.Math.*;
import de.physolator.usr.tvg.TVG;
import de.physolator.usr.util.Colors;

public class LokomotiveTVG extends TVG {

    private Lokomotive lokomotive;

    public Lokomotive0TVG(Lokomotive lokomotive) {
        this.lokomotive = lokomotive;
        geometry.setRim(20, 20, 20, 20);
        geometry.setUserArea(-2, 10, -1, 5);
        scalesStyle.visible = true;
    }

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

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

public class Lokomotive extends PhysicalSystem {

    public void initGraphicsComponents(GraphicsComponents g) {
        g.addTVG(new LokomotiveTVG(this));
    }
}
```

1. Teilaufgabe

Die Lokomotive gemäß der obigen Abbildung gezeichnet werden. Verwenden Sie dazu die folgenden Befehle.

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*
drawRectangle(x1,y1,x2,y2);	zeichnet ein Rechteck, bei dem die linke untere Ecke die Position *(x1,y1)* und die rechte obere Ecke die Position *(x2,y2)* hat Die Kanten des Rechteckse verlaufen parallel zur x- bzw. y-Achse.
drawCircle(x,y,r,POLYGON);	zeichnet eine Kreisfläche analog zu *drawCircle(x,y,r);*
drawRectangle(x1,y1,x2,y2,POLYGON);	zeichnet eine Rechteckfläche analog zu *drawRectangle(x1,y1,x2);*

Die ersten drei Befehle der Tabelle zeichnen Linien, die letzten beiden Befehle Flächen. Mit den Variablen *style.strokeColor* wird bei den ersten drei Zeichenbefehlen festgelegt, welche Farbe die Linie haben soll und mit *style.strokeWidth* die Linienbreite in Pixeln. Die Variable *style.fillColor* bezieht sich auf die letzten beiden Befehle und legt fest, mit welche Farbe die Fläche gefüllt werden soll. Die Zeichenbefehle verwenden beim Zeichnen immer die aktuellen Werte dieser Variablen. Damit die Zeichenbefehle die gewünschte Farbe und die gewünschte Linienbreite verwenden, muss man diesen Variablen vor der Ausführung der Zeichenbefehle die gewünschten Werte zuweisen. Beispiel: Die folgende Befehle bewirken, dass alle nachfolgenden Zeichenbefehle eine Linienbreite von 3 Pixeln verwenden, dass die Linien mit der Farbe *Colors.black* und die Flächen mit der Farbe *Colors.darkGrey* gezeichnet werden.

```
style.strokeWidth = 3;
style.strokeColor = Colors.black;
style.fillColor = Colors.darkGrey;
```

Verwenden Sie für den Rumpf der Lokomotive die Farbe *Colors.darkGrey*, für das Fenster die Farbe *Colors.white* und für die Räder und den Stangenantrieb die Farbe *Colors.black.* Die Linien der Räder und die Linien der Speichen sollen eine Breite von 2 Pixeln und die Stangen einen Breite von 3 Pixeln haben.

Empfehlungen

Teilen Sie die Gesamtaufgabe in mehrere Methode auf. Schreiben Sie eine Methode, mit der ein Rad an einer vorgegebenen Position und vorgegebenem Radius zeichnet. Verwenden Sie diese Methode zum Zeichnen von allen drei Rädern. Beim Zeichnen der der Speichen der Räder empfiehlt es sich einen for-Schleife zu verwenden. In jedem Schleifendurchlauf wird eine Speiche gezeichnet. Die Lage der Speiche ergibt sich aus der Laufvariable *i.* Verwenden Sie die Sinus- und die Kosinusfunktion zur Bestimmung der Linienposition in Abhängigkeit von *i.*

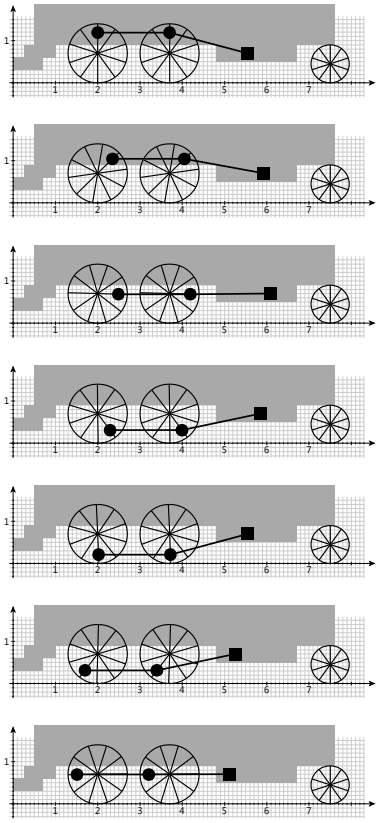
2. Teilaufgabe

Schreiben Sie ein parameterbehaftete Methode *zeichneLokomotive.* mit dem folgenden Aufbau:

```
public void zeichneLokomotive(double x, double y, double alpha, double beta) {
    // Platz für Zeichenbefehle
}
```

Mit den Parametern *x* und *y* wird festgelegt, wohin die Lokomotive gezeichnet werden soll. Die Lokomotive soll im Vergleich zur ursprünglichen Position (obige Abbildung) um *(x,y)* verschoben dargestellt werden.

Die Räder sollen gedreht werden können. Die Parameter *alpha* und *beta* beschreiben, um welchen Winkel die Räder im Vergleich zur ursprünglichen Darstellung gedreht werden sollen. Beide Winkel seien in Bogenmaß angeben. Der Winkel *alpha* bezieht sich auf den Drehwinkel der beiden großen, hinteren Räder. Diese beiden Räder sind über den Stangenantrieb miteinander und mit dem Zylinder (schwarzes Quadrat) verbunden. Beide Räder haben immer den gleichen Drehwinkel. Der Zylinder bewegt sich nur in horizontaler Richtung. Seine Position ist abhängig von *alpha.*



Der Winkel *beta* bezieht sich auf das vordere, kleinere Rad. Das kleine Rad ist nicht mit den großen Rädern verbunden.

Rufen sie Methode *zeichneLokomotive* in der Methode *paint* auf. Führen Sie, analog zu Teilaufgabe 2 aus Aufgabenblatt 4, vier Objektattribute *x, y*, *alpha* und *beta* ein. Rufen Sie die Methode *zeichneLokomotive* mit diesen Objektattributwerten auf. Fügen Sie *@Parameter*- und *@Slider*-Annotations ein, damit die Objektattribute vom Benutzer während der Laufzeit verändert werden können.

Schema:

```
@Parameter
@Slider(min = 0, max = 10, step=0.01, width = 200)
public double x = 0;
...
```

3. Teilaufgabe

Die Lokomotive soll zum Fahren gebracht werden. Die Lokomotive soll sich mit einer Geschwindigkeit von $1\frac{m}{s}$ nach rechts bewegen.

Fügen Sie zunächst folgende Zeile am Anfang in die paint-Methode ein:

```
double t = lokomotive.getSimInfo().getActualFrameData().simulationTime;
```

Sie erhalten damit Zugriff auf die aktuelle Simulationszeit *t*. Bei der vorangegangen konnten die Werte der Parameter *x*, *y*, *alpha* und *beta* vom Benutzer beliebig und unabhängig voneinander festgelegt werden. Jetzt sollen die Werte dieser Parameter in Abhängigkeit von *t* so bestimmt werden, dass der Zug sich mit einer Geschwindigkeit $1\frac{m}{s}$ nach rechts bewegt und sich die Räder mit der richtigen Rotationsgeschwindigkeit drehen.

Starten Sie die Simulation!