INFORMATIK · KLASSE 10

LÖSUNGEN

# Lösungsblatt: Aufgabe 1

Wiederholung - Klassen, Objekte und Vererbung

> Die Formulierungen sind Musterlösungen. Sinngemäß richtige Antworten und technisch gleichwertige Programmvarianten sind ebenfalls korrekt.

## 1 · Musterlösungen zu a), b), c) und f)

### a) Bedeutung der Programmzeilen

`Kiste block = new Kiste();` erzeugt ein neues Objekt der Klasse `Kiste`. Die Objektvariable `block` verweist auf dieses Objekt.

`Ball ball1 = new Ball(100);` erzeugt ein neues Objekt der Klasse `Ball`. Die Objektvariable `ball1` verweist darauf; dem Konstruktor wird der Wert `100` übergeben.

### b) Bedeutung der 100

Die `100` ist das **Argument** (auch: aktueller Parameter) beim Konstruktoraufruf. Es wird dem formalen Parameter `startX` zugewiesen und legt im Konstruktor die x-Position des Kreismittelpunkts fest.

`Ball(float startX)` → `new Circle(startX, 50, 30)`

### c) Klasse und Objekt

Eine **Klasse** ist der Bauplan. Sie beschreibt Attribute und Methoden, zum Beispiel die Klassen `Ball` und `Kiste`.

Ein **Objekt** ist eine konkrete Instanz dieses Bauplans. `ball1` bezeichnet ein Ball-Objekt, `block` ein Kiste-Objekt.

### f) `extends Actor`

`Ball` **erbt** von der Oberklasse `Actor`. Ein Ball ist damit ein spezieller Actor und kann dessen Eigenschaften und Methoden verwenden. Die Methode `act()` wird von der Umgebung wiederholt aufgerufen und bestimmt das Verhalten des Balls.

## 2 · Klassendiagramme zu d)

**Actor** △ - - - **Ball und Kiste**

| Ball |
|---|
| - geschwindigkeit: int<br>- ball: Circle |
| + Ball(startX: float)<br>+ act(): void<br>+ setzeBallfarbe(neueFarbe: String): void<br>+ bewegeBall(): void<br>+ setzteBallnachoben(): void |

| Kiste |
|---|
| - geschwindigkeit: int<br>- farbe: Color<br>- kiste: Rectangle |
| + Kiste()<br>+ act(): void<br>+ bewegeKisterechts(): void<br>+ setzeKistenfarbe(neueFarbe: String): void |

## 3 · Objektdiagramm zu e)

| block : Kiste |
|---|
| geschwindigkeit = 10<br>farbe = Color.green<br>kiste = Rectangle-Objekt |

| ball1 : Ball |
|---|
| geschwindigkeit = 5<br>ball = Circle-Objekt<br>Startposition = (100 \| 50) |

INFORMATIK · KLASSE 10

BONUSLÖSUNGEN

# Programmierlösungen

Eine mögliche Umsetzung der Bonusaufgaben g) bis k)

## 4 · Änderungen in Kiste

**g) Kiste am unteren Rand erzeugen**

```
Kiste()
{
   kiste = new Rectangle(300, 550, 100, 100);
   setzeKistenfarbe(farbe);
   geschwindigkeit = 10;
}
```

**i) Rechts hinaus - links wieder erscheinen**

```
void act()
{
   bewegeKisterechts();

   if(kiste.isOutsideView())
   {
      kiste.moveTo(50, kiste.getCenterY());
   }
}
```

Bei einer Zeichenfläche mit anderer Höhe muss die y-Position in g) angepasst werden. Bei 600 Pixeln Höhe und einer 100 Pixel hohen Kiste liegt ihr Mittelpunkt am unteren Rand bei `y = 550`.

## 5 · Änderungen in Ball

**h) Oben neu erscheinen**

Diese Funktion ist im Ausgangsprogramm bereits enthalten:

```
void act()
{
   bewegeBall();

   if(ball.isOutsideView())
   {
      setzteBallnachoben();
   }
}

void setzteBallnachoben()
{
   ball.moveTo(ball.getCenterX(), 0);
}
```

**j) Bis zur Geschwindigkeit 20 beschleunigen**

```
void setzteBallnachoben()
{
   ball.moveTo(ball.getCenterX(), 0);

   if(geschwindigkeit < 20)
   {
      geschwindigkeit =
         geschwindigkeit + 1;
   }
}
```

**k) Bei maximaler Geschwindigkeit rot färben**

```
void setzteBallnachoben()
{
   ball.moveTo(ball.getCenterX(), 0);

   if(geschwindigkeit < 20)
   {
      geschwindigkeit = geschwindigkeit + 1;
   }

   if(geschwindigkeit >= 20)
   {
      ball.setFillColor(Color.red);
   }
}
```

**Hinweis zur Bewertung:** Entscheidend sind die Begrenzung auf 20, das Zurücksetzen nach oben und der Farbwechsel. Ob diese Schritte in einer Hilfsmethode oder direkt in `act()` stehen, ist für die Funktion unerheblich.

## 6 · Kurze Prüfkriterien

**Begriffe**

Klasse, Objekt, Konstruktor, Argument/Parameter, Attribut, Methode und Vererbung werden korrekt verwendet.

**Diagramme**

Klassendiagramme zeigen Struktur und Typen; Objektdiagramme zeigen konkrete Instanzen und aktuelle Attributwerte.