



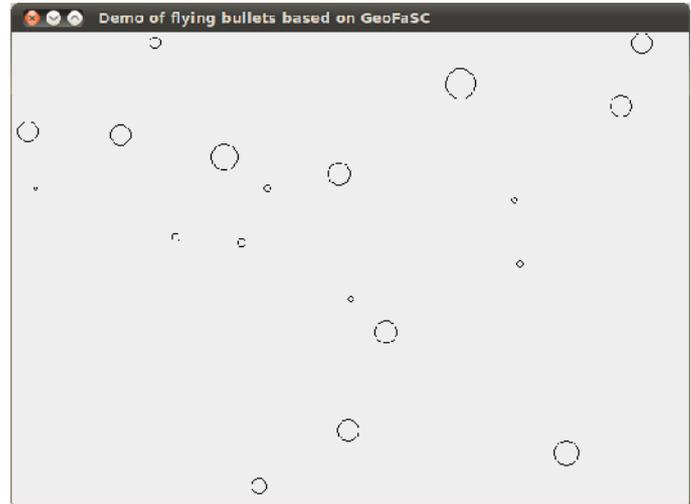
## Aufgabe 1 (Bullets)

### **Bezug:**

Projekt „Billardsimulation“ aus Kapitel 6 „Eigene Klassen I“ [Schriek 2006, S. 57 ff.]

**Lernaspekte:** (siehe detaillierter [Schriek 2006, S. 52])

- Attribute einer Klasse,
- Klassenbeziehungen:
  - spezialisierende Vererbung,
  - hat- und kennt-Beziehung,
  - Beziehung mit sich selbst.
- Listen-Datenstruktur als Black-Box.



### **Materialien (auf der CD):**

- GeoFaSC-Bibliothek,
- Vorgefertigtes BlueJ-Projekt mit Quelltextextrahnen,
- GeoFaSC-Dokumentation (als JavaDoc)/ -Referenz (als pdf-Datei).

### **Aufgabenstellung:**

Eine `Bullet` (dt. Kugel) kann sich mit einer konstanten Geschwindigkeit von 5 Pixeln in eine bestimmbare Richtung bewegen. Dabei soll sie einfach als `Circle` (dt. Kreis) auf einer `Canvas` (dt. Zeichenfläche) visualisiert werden. Aus diesem Grund und zur Nutzung vorhandener Eigenschaften, insbesondere der Bewegungsrichtung der `Bullet`, ist die Klasse `Bullet` aus der Klasse `Circle` abgeleitet. Eine mögliche Modellierung ist mit dem umseitigen Klassendiagramm gegeben.

Die Klasse `BulletsMover` dient lediglich als Hauptprogramm, in dem mehrere `Bullets` zufälliger Größe und Position innerhalb der `Canvas` erzeugt, bewegt und schließlich dargestellt werden.

Bewegt sich eine `Bullet` mit der Bewegungsrichtung  $0^\circ$  bzw.  $360^\circ$ , so bewegt sie sich waagrecht nach rechts. Ist sie  $180^\circ$ , so bewegt sie sich waagrecht nach links. Ist sie  $90^\circ$ , so bewegt sie sich senkrecht nach oben. Ist sie  $270^\circ$ , so bewegt sie sich senkrecht nach unten. Für andere Gradzahlen bewegt sie sich entsprechend.

- a) Implementieren Sie die Klasse `Bullet` an den markierten Stellen im Quelltext aus und testen Sie Ihr Programm durch Ausführung von `BulletsMover`.

### **Hinweise:**

- Die `Bullets` sollen bei Kollision mit den Rändern der `Canvas` von dieser abprallen.
- Nutzen Sie als Hilfe das Modell und die Kommentare im Quelltext.
- Abweichend vom Modell enthalten die Quelltexte bereits Elemente, die erst später gebraucht werden.

- b) Der Quelltext der Klasse `Bullet` umfasst ein Attribut, Methoden und eine Beziehung, die im Modell von `Bullet` nicht aufgeführt sind. Korrigieren Sie das Modell dahingehend.

Wozu könnten die fehlenden Elemente nützlich sein?

- c) Implementieren Sie in der Klasse `Bullet` die Methode `hits(Bullet b)` aus. Bei deren Aufruf für eine `Bullet bullet`, soll sie `true` zurückliefern, wenn `bullet` mit `b` kollidiert, und andernfalls `false`.

#### Hinweise:

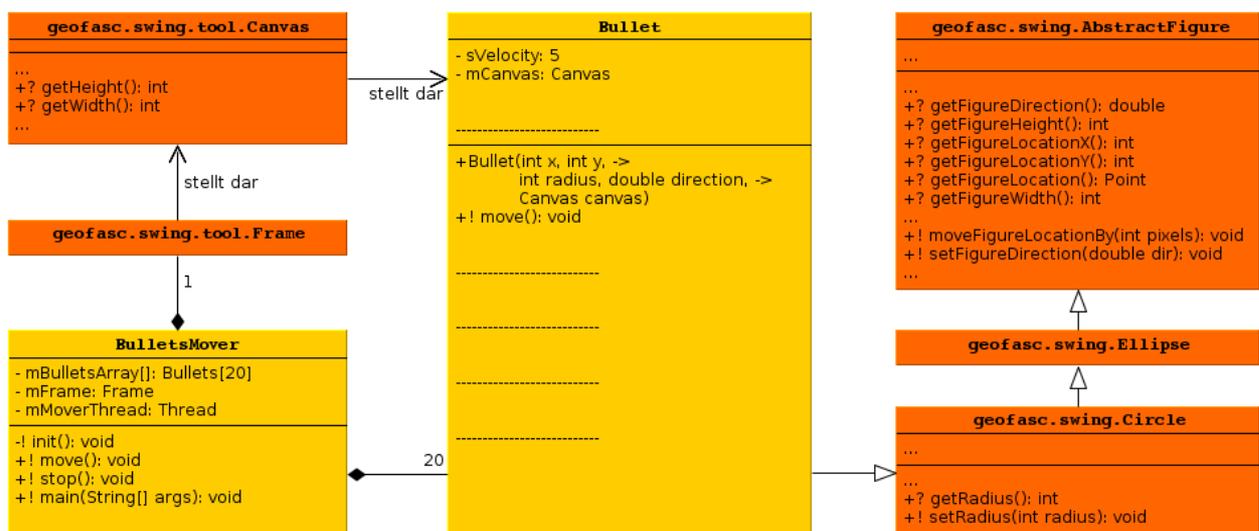
- Berechnen Sie die Abstände zweier `Bullets` über die Abstandsberechnung ihrer Mittelpunkte mit Hilfe des Satzes des Pythagoras.
- Nutzen Sie für mathematische Operationen die Klasse `java.lang.Math` oder `geofasc.swing.tool.Calculator`.

- d) Ergänzen Sie in der Klasse `Bullet` die Methode `move()`, so dass diese zusätzlich eine Kollisionsprüfung und -behandlung mit anderen `Bullets` vornimmt. Das Bekanntmachen einer `Bullet` mit ihren Nachbarn muss in der `init`-Methode der Klasse `BulletsMover` implementiert werden. Testen und korrigieren Sie ggf. ihre Implementierung.

#### Hinweise:

1. Iterieren Sie dazu über alle Nachbarn `b` der aufgerufenen `Bullet bullet`.
2. Prüfen Sie mittels der Methode `hits(Bullet b)`, ob eine Kollision von `bullet` mit `b` vorliegt. Falls ja, ändern Sie die Richtung von `bullet` durch Subtraktion von  $180^\circ$  von ihrer aktuellen Bewegungsrichtung.

### Modell (Klassendiagramm):



[Schriek 2006] : Schriek, Bernhard: Informatik mit Java - Eine Einführung mit BlueJ und der Bibliothek Stifte und Mäuse, Band I. Nili-Verlag, 2005.