

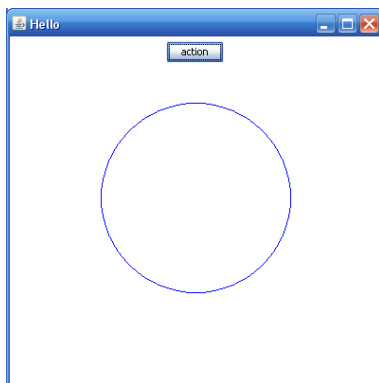
## Arbeitsblatt: INF2

Name:  Kurznamen:

### Vererbung

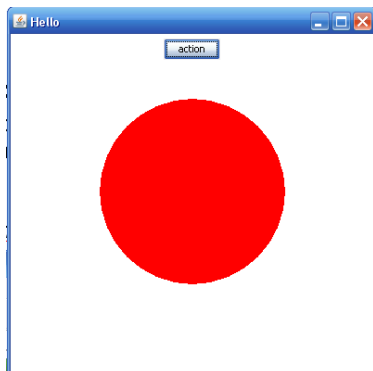
#### Aufgabe 1: Balloon erbt von Ball

Sie haben eine HelloBallon5 Klasse vorgegeben. Darin wird ein Ball in initComponents hinzugefügt. Nach dem Start sollte es wie folgt aussehen.



Wir erstellen nun eine neue Klasse Balloon5, die **von Ball erbt**.

- Die draw Methode soll angepasst werden, dass ausgefüllt gezeichnet wird.
  - Die move Methode soll angepasst werden, dass der Ballon sich 1 Pixel nach oben bewegt.
- Fügen Sie jetzt einen roten Balloon der initComponents Methode hinzu.



### Hinweise:

- Beachten Sie: Obwohl der Ballon in einem Ball Objekt gespeichert wird, wird die draw Methode des Ballons aufgerufen → Polymorphismus Prinzip
- Der Konstruktor von Ball soll im Konstruktor von Balloon aufgerufen werden
- Die Koordinaten sollen von Ball übernommen werden: **keine neuen einführen!**
- `add(new Balloon5(Color.RED, 200, 200, 100));`
- Wenn wir nun den action Knopf drücken, sollte der Ballon langsam steigen
- Wir haben zufällig die japanische Flagge gezeichnet, welches auch als "Land der aufgehenden Sonne" bezeichnet wird, was wir hiermit programmiert haben ;-)

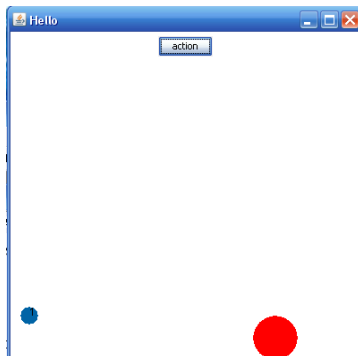
### Abgabe

Praktikum: INF7.1

Filename: Balloon5.java

## Aufgabe 2: Billard erbt von Ball

Wir erstellen nun eine neue Klasse Billard, die wieder **von Ball erbt**.



Billard soll noch ein Geschwindigkeitsvektor (vx,vy) beim Konstruktor mitgegeben werden können (als zwei zusätzliche Parameter am Schluss) und die Bewegung soll in move entsprechend diesem Vektor erfolgen. Wenn Billard an den Rand trifft, soll er abprallen, was man einfach mit einer Umkehrung der Geschwindigkeitsvektorkomponente implementieren kann (wie Ihnen ja aus der Physik geläufig ist). Dies wird in der Methode `testBounce` implementiert.

### Hinweise:

- ```
private void testBounce() {  
    if (mx < r || mx > 400-r) vx = -vx;  
    if (my < r + 25 || my > 400-r) vy = -vy;  
}
```
- `add(new Balloon5(Color.RED, 300, 375, 25));`
- `add(new Billard(ZHAWBLUE, 25, 350, 10, 2, -1));`

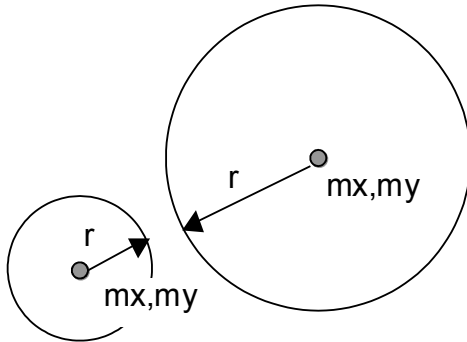
### Abgabe

Praktikum: INF7.2

Filename: Billard.java

### Aufgabe 3 Interaktion verschiedener Objekte

In `testCollision` soll überprüft werden, ob die Billard Kugel auf einen Ballon trifft. Dazu müssen alle andern Bälle (Billiard oder Balloon) paarweise überprüft werden (in `others` übergeben), ob die Distanz der Mittelpunkte kleiner ist als die Summe der Radien. Bei einem Treffer soll der Ballon platzen, was wir dadurch implementieren, dass wir den Radius des Ballons zu 0 setzen. Bei obigen Startwerten sollte die Kugel den Ballon treffen.



#### Hinweise

- in `others` werden alle Bälle übergeben also auch das eigene Objekt. Dieses muss für den Test ausgenommen werden.
- In der Methode `collide` soll überprüft werden, ob eine Kollision zwischen 2 Objekten stattgefunden hat.
- mittels `instanceof Balloon5` kann überprüft werden, ob es sich um einen Balloon handelt

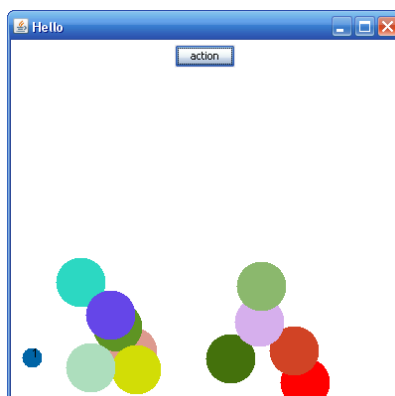
#### Abgabe

Praktikum: INF7.3

Filename: Billard.java

### Aufgabe 4 (optional)

Statt nur einem Balloon können gleich 10 zufällig eingefügt werden, was in `addRandomBalloons` gemacht wird.



Das Platzen kann noch akustisch untermalt werden; folgende Methode erlaubt es ein ".wav" File abzuspielen; verwenden Sie dafür Balloon.wav. Das wav File muss im Verzeichnis, wo die .class File liegen abgelegt sein, damit es gefunden wird.

```
Clip clip = null;
public void playSound(String filename) {
    try {
        if (clip == null) {
            AudioInputStream audioIn = AudioSystem.getAudioInputStream(
                new File(filename));
            clip = AudioSystem.getClip();
            clip.open(audioIn);
        }
        if (!clip.isActive()) {
            clip.setFramePosition(0);
            clip.start();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Das Flickern kann mit sog. *double Buffering* verhindert werden, was in HelloBufferedBalloon5.java gezeigt wird.

Statt den Ballonen könnte man auch mehrere Billard Objekte einfügen, was dann aber zum Problem des *schiefen, zentralen elastischen Stosses* führt

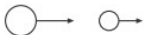



| Stoßart   | Bild                                                                                | Charakteristika                                                                                                      |
|-----------|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| gerade    |  | Die Bahnen beider Schwerpunkte liegen auf einer Geraden.                                                             |
| schief    |  | Die Bahnen beider Schwerpunkte liegen in einer Ebene und schließen einen Winkel ein.                                 |
| zentral   |  | Die Schwerpunkte der Stoßpartner liegen auf der Normalen zur Berührungsebene durch den Berührungspunkt (Stoßnormale) |
| elastisch |  | Die Summen der kinetischen Energien vor und nach dem Stoß sind gleich.                                               |

Abbildung 2.21: Klassifikation der Stoßprozesse.

<https://web.physik.rwth-aachen.de/~fluegge/Vorlesung/PhysIpub/Exscript/4Kapitel/IV6Kapitel.html#IV63>

<https://www.hzdr.de/db/Cms%3FpOid%3D29849>