OOP

Wenn, wenn und noch mal wenn

Henrik Horstmann

23. September 2014

Inhaltsverzeichnis

1 Bedeutung der Symbole....1
2 Das Spiel....2
3 Das Spielfeld....2
4 Jetzt kommt der Ball ins Spiel....3
4.1 Der Ball kommt ins rollen....4
4.2 Ball, bleib auf dem Spielfeld....6
4.3 Tooooooooooor!....7
4.4 Der Spieler betritt das Feld....11
4.5 Training für den Spieler....12
4.6 Die eigene Mannschaft muss auch mal verlieren....13
4.7 Ergänzungen zur Verschönerung....13
5 Übungen....15
5.1 Ab auf den Tennisplatz....15
5.2 Etwas mehr Gerechtigkeit....16
6 Lösungen zu den Aufgaben....17
6.1 Lösung zu Aufgabe 1....17
6.2 Lösung zu Aufgabe 2....18
6.3 Lösung zu Aufgabe 3....19
6.4 Tipp zu Aufgabe 4....20
6.5 Lösung zu Aufgabe 4....21
6.6 Lösung zu Aufgabe 5....23
6.7 Lösung zu Aufgabe 6....24
6.8 Lösung zu Aufgabe 7....26
6.9 Lösung zu Aufgabe 8....28
6.10 Lösung zu Aufgabe 9 (a)....30
6.11 Lösung zu Aufgabe 9 (b)....32
6.12 Lösung zu Aufgabe 9 (c)....34
6.13 Lösung zu Aufgabe 10....36
6.14 Lösung zu Aufgabe 10....38
6.15 Lösung zu Aufgabe 11 (a)....40

6.16 Lösung zu Aufgabe 11 (b).....42
6.17 Lösung zu Aufgabe 11 (c).....44
6.18 Lösung zu Aufgabe 12.....46
6.19 Lösung zu Aufgabe 13.....48
6.20 Lösung zu Aufgabe 14.....51

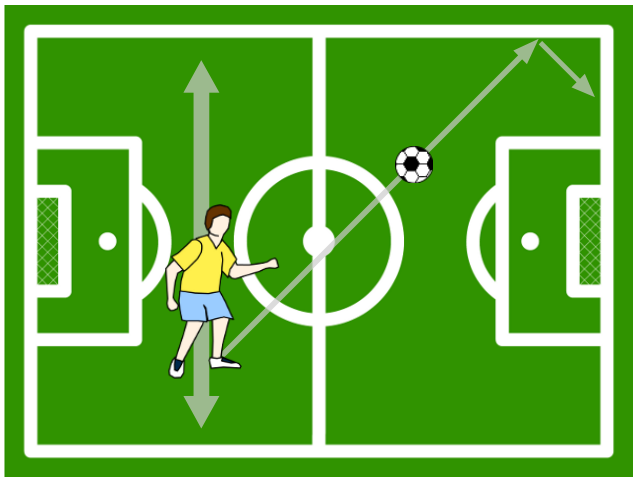
1 Bedeutung der Symbole

Symbol	Beschreibung
	Dieses Symbol kennzeichnet einen Tipp oder Hinweis.
	Aufgepasst, hier passieren leicht Fehler oder es ist mit Schwierigkeiten zu rechnen.
	Hier erfahren Sie, wie es gemacht wird.
	Es folgt eine Aufgabe die zu lösen ist.

© 2014 Henrik Horstmann

2 Das Spiel

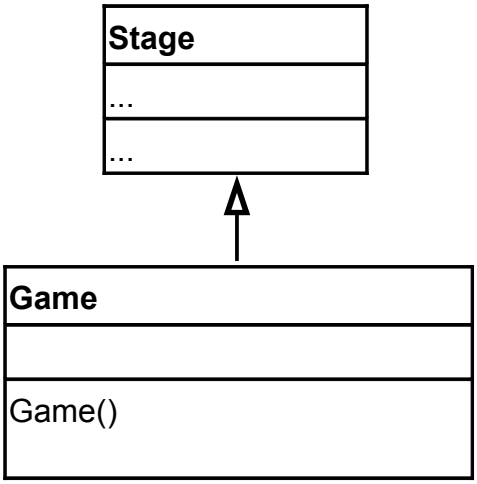
Im folgenden sollen an einem Computerspiel zahlreiche Möglichkeiten von Bedingten Anweisungen gezeigt werden.



Ein Fußballspieler kann mit zwei Tasten auf dem Spielfeld nach oben und unten bewegt werden. Der Ball rollt in einem bestimmten Winkel mit einer bestimmten Geschwindigkeit über das Spielfeld. An den Rändern prallt der Ball ab, so dass er nicht vom Spielfeld rollt. Trifft der Ball das dem Spieler gegenüberliegende Tor, so erhält der Spieler einen Punkt. Rollt der Ball hinter den Spieler, so wird ihm ein Punkt abgezogen.

3 Das Spielfeld

Das ganze Spiel basiert auf einem Fußballfeld, das unsere „Bühne“, bzw. Stage ist.



Als Hintergrund der Stage soll das Bild *Fussballfeld.png* geladen werden. Es hat eine Größe von 640×480 Pixel.

Aufgabe 1:

Erstellen Sie ein Projekt *Fussball.oop*.

Implementieren Sie die Klasse *Game* entsprechend dem UML-Diagramm.

Im Konstruktor von *Game* soll die Größe der Stage festgelegt und das Hintergrundbild geladen werden.

Hier geht es zur Lösung auf Seite 17.

4 Jetzt kommt der Ball ins Spiel

Der Ball wird durch ein *StageObject* dargestellt. Seine aktuelle Postion wird in Attributen festgehalten. Damit erweitert sich die Klasse *Game* wie folgt:

Game
cx:Integer cy:Integer ball:StageObject
Game() resetBall()

Die aktuelle Position des Balls ist (cx,cy) .

Zu Spielbeginn soll der Ball in Spielfeldmitte liegen. Die Methode *resetBall* legt den Ball genau auf Spielfeldmitte.

Aufgabe 2:

Ergänzen Sie die Klasse *Game* um die zusätzlichen Attribute und die Methoden *resetBall*.

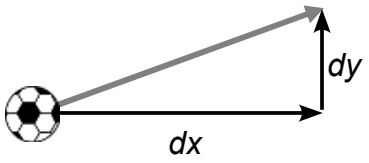
Erweitern Sie den Konstruktor so, dass zunächst die Methode *resetBall* aufgerufen und anschließend ein Objekt für das *StageObject ball* erzeugt und auf der *Stage* platziert wird. Das Bild für den Ball heißt *Fussball.png*.

Hier geht es zur Lösung auf Seite 18.

© 2014 Henrik Horstmann

4.1 Der Ball kommt ins rollen

Zunächst muss eine Richtung und die Geschwindigkeit für den Ball bestimmt werden. Dies geschieht über die zwei Werte *dx* und *dy*. *dx* gibt an, um wie viel Pixel der Ball in x-Richtung (waagerecht) und *dy* gibt an, um wie viel der Ball in y-Reichtung rollt.



Die Klasse muss um zwei weitere Attribute erweitert werden. Außerdem wird eine Methode benötigt, die fortlaufend die neue Position des Balles berechnet und ihn entsprechend auf dem Spielfeld platziert. Dies wird die Methode *run* sein.

Game
cx:Integer cy:Integer dx:Integer dy:Integer ball:StageObject
Game() resetBall() run()

Aufgabe 3:

Erweitern Sie Die Klasse *Game* um die oben beschriebenen Attribute und der Methode *run*.

Die Attribute *dx* und *dy* sollen in der Methode *resetBall* auf 0 gesetzt werden so dass der Ball sich zunächst nicht bewegt.

Hier geht es zur Lösung auf Seite 19.

Abfragen einer Tastatureingabe:

Die Klasse *Stage* besitzt eine Methode *isKeyPressed*, mit der abgefragt werden kann, ob eine Taste der Tastatur betätigt wurde.

```
Boolean Stage.isKeyPressed(String key)
```

Dabei gibt *key* an, welche Taste abgefragt werden soll.

Beisiel:

```
stage.isKeyPressed("s")
```

liefert *TRUE*, wenn die Taste <s> gedrückt wurde.

Sonderfälle:

key=	*Taste*
"UP"	Pfeil nach oben Taste
"DOWN"	Pfeil nach unten Taste
"LEFT"	Pfeil nach links Taste
"RIGHT"	Pfeil nach rechts Taste

StageObject auf der Bühne Verschieben:

Die Klasse *StageObject* besitzt eine Methode, mit der das Objekt an eine beliebige Position auf der *Stage* positioniert werden kann.

```
void StageObject.moveToPosition(Integer x, Integer y)
```

x und *y* sind die Koordinaten des Objekt Mittelpunktes. Der Ursprung des Koordinatensystems liegt in der oberen, linken Ecke der Stage.

Aufgabe 4:

Fragen Sie in der *run* Methode die Taste <s> ab. Wird Sie betätigt, so soll das Attribut *dx* mit einem Zufallswert im Bereich von -10 bis -5 und *dy* mit einem Zufallswert im Bereich von 5 bis 10 versehen werden.

Danach soll in der *run* Methode in Abhängigkeit von *dx* und *dy* die neue Position des Balls berechnet werden (Auf Seite 20 gibt es einen Tipp dazu) und in den Attributen *cx* und *cy* gespeichert werden. Der Ball ist an die neue Position zu setzten.

Erweitern Sie in der Klasse *Program* die Methode *main*, so dass ein Objekt der Klasse *Game* erzeugt wird. Rufen Sie die Methode *startSingle* mit 5 auf. Abschließend soll ein Aufruf von *Computer.waitForExitSignal* verhindern, dass das Programm sofort beendet wird.

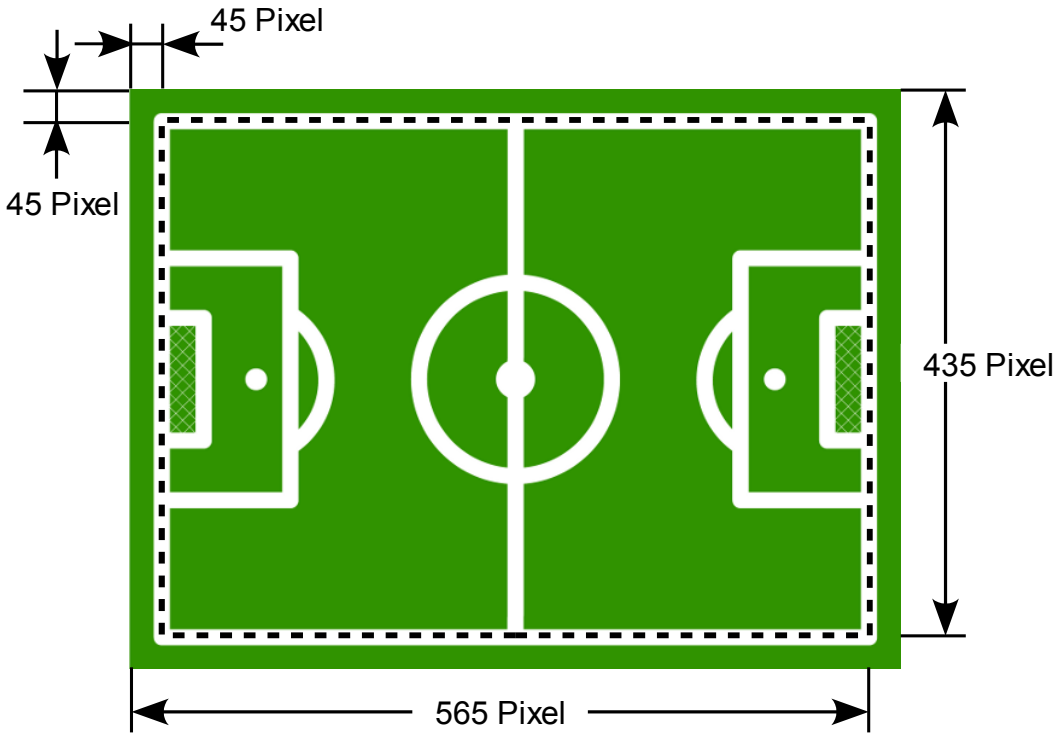
Testen Sie Ihr Programm.

Hier geht es zur Lösung auf Seite 21.

Wenn nach dem Start des Programms die Taste <s> gedrückt wird, rollt der Ball direkt nach links unten aus dem Spielfeld raus.

4.2 Ball, bleib auf dem Spielfeld

Der Ball soll sich nur innerhalb der weißen Begrenzungslinien im Spielfeld bewegen und nicht vom Platz rollen.



Damit der Ball nicht vom Platz rollt, muss geprüft werden, ob er sich auf einer der Begrenzungslinien oder schon dahinter befindet.

Aufgabe 5:

Fügen Sie am Anfang der Methode *run* der Klasse *Game* zwei bedingte Anweisungen ein.

Die erste Bedingte Anweisung prüft, ob sich der Ball links oder rechts außerhalb des markierten Bereichs befindet.

Die zweite Bedingte Anweisung prüft, ob sich der Ball oben oder unten außerhalb des markierten Bereichs befindet.

Tipp: Die Prüfung kann über die aktuelle Position des Balls geschehen. Diese ist in den Attributen *cx* und *cy* gespeichert.

Hier geht es zur Lösung auf Seite 23.

Jetzt ist lediglich bekannt, wann der Ball das Spielfeld verlässt. Der Ball muss noch seine Laufrichtung ändern. Dazu wird nach folgender Strategie verfahren. Rollt der Ball links oder rechts aus dem Spielfeld, so wird die x-Richtung

© 2014 Henrik Horstmann

geändert. Entsprechend wird, wenn der Ball oben oder unten aus dem Bild rolt, die y-Richtung geändert.

Hinweis:

Um die x-Richtung zu ändern muss lediglich das Vorzeichen von *dx* geändert werden. Entsprechend wird durch Vorzeichenänderung von *dy* die y-Richtung geändert.

Das Vorzeichen einer Zahl kann durch Multiplikation mit -1 geändert werden.

Aufgabe 6:

Führen Sie in den bedingten Anweisungen aus Aufgabe 5 die entsprechenden Richtungsänderungen aus.

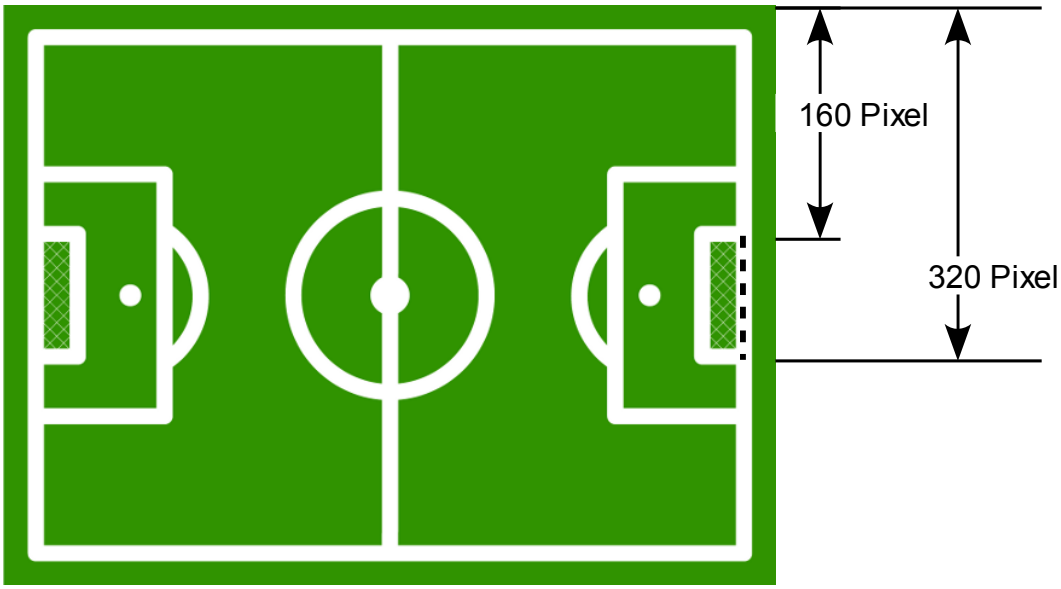
Testen Sie Ihr Programm. Der Ball sollte sich jetzt nur noch innerhalb des Spielfeldes bewegen.

Hier geht es zur Lösung auf Seite 24.

4.3 Tooooooooooor!

Rollt der Ball in das rechte Tor, so soll der Punktezähler um eins erhöht und der Ball wieder in die Spielfeldmitte platziert werden.

Es wird festgelegt, dass der Ball genau dann im Tor ist, wenn er über die Spielfeldbegrenzungslinie im Tor rollt.



Es muss somit zum einen getrennt geprüft werden, ob der Ball links oder rechts aus dem Spielfeld rollt und zum anderen, sollte er rechts aus dem Spielfeld rollen, ob das im Tor geschieht.

Bedingte Anweisungen aneinanderreihen:

Um bedingte Anweisungen aneinander zu reihen, folgt dem *else*-Statement direkt ein *if*-Statement, ein sogenanntes *else-if*.

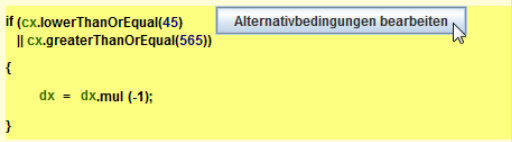
Beispiel:

```
if ( ... )
{
}
else if ( ... )
{
}
else if ( ... )
{
}
else
{
}
```

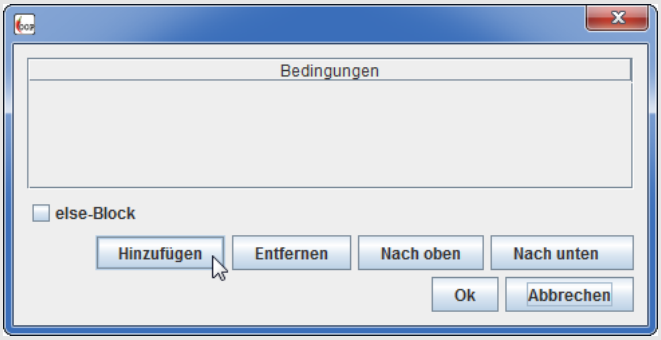
Bei der Programmausführung wird zuerst die oberste Bedingung geprüft. Ist sie *TRUE* so wird der entsprechende Anweisungsblock ausgeführt. Ist sie *FALSE*, so wird die nächste Bedingung geprüft, usw.

Einfügen von weiteren bedingten Anweisungen in HOOPLU:

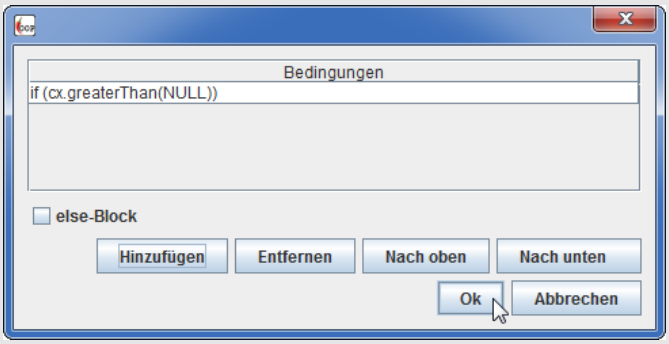
Den Mauszeiger neben die erste Bedingung positioniere. Ein Button mit der Aufschrift *Alternativbedingungen bearbeiten* erscheint.



Ein Mausklick auf den Button öffnet in Dialogfenster. Um weitere Bedingungen einzufügen auf den Button *Hinzufügen* klicken.



Eine zusätzliche Bedingung ist eingefügt. Die Änderungen mit einem Klick auf *Ok* bestätigen.



Im Programm ist ein neuer *if-else* Zweig eingefügt, der auf die gleiche Weise bearbeitet werden kann wie ein *if*-Statement.

Aufgabe 7:

Teilen Sie die bedingte Anweisung für die Prüfung der linken und rechten Spielfeldränder auf, so dass der linke und der rechte Spielfeldrand getrennt geprüft werden.

Hier geht es zur Lösung auf Seite 26.

Rollt der Ball über den rechten Spielfeldrand, so muss geprüft werden, ob es innerhalb des Tor geschieht.

Aufgabe 8:

Fügen Sie innerhalb des *if-else* Block eine weitere bedingte Anweisung ein.

In dieser bedingten Anweisung soll geprüft werden, ob der Ball sich innerhalb des Tores befindet. Diese Prüfung erfolgt ebenfalls über die Koordinaten des Balls und der Position des Tores (siehe Grafik weiter oben).

Ist der Ball im Tor, so soll die Methode *resetBall* aufgerufen werden. Sie legt den Ball zurück in die Spielfeldmitte.

Andernfalls wird die x-Richtung des Balls wie gehabt geändert.

Testen Sie Ihr Programm.

Hier geht es zur Lösung auf Seite 28.

Es fehlt noch der Punktezähler und die Ausgabe des Punktestands. Dazu muss die Klasse *Game* um entsprechende Attribute und Methoden erweitert werden:

© 2014 Henrik Horstmann

Game
cx:Integer cy:Integer dx:Integer dy:Integer ball:StageObject punkte:Integer punkteAnzeige:StageObject
Game() resetBall() run() setzePunkte(aenderung:Integer)

Aufgabe 9:

Ergänzen Sie die Klasse *Game* um die fehlenden Attribute und Methoden.

Erzeugen Sie im Konstruktor ein Objekt für das Attribut *punkteAnzeige* und platzieren Sie es an der Position (320, 100) auf der *Stage*. Es wird kein Bild geladen und die Skalierung soll 100 % betragen. Setzen Sie die Textausrichtung auf zentriert.

Erzeugen Sie im Konstruktor ein Objekt für da Attribut *punkte* und rufen Sie anschließend die Methode *setzePunkte* mit 0 auf.

Hier geht es zur Lösung auf Seite 30.

In der Methode *setzePunkte* soll zum Attribut *punkte* der Parameter *aenderung* addiert werden. Anschließend soll der neue Punktestand angezeigt werden. Dazu muss ein *String* erzeugt werden der "Punkte: [*punkte*]" ([*punkte*] ist der Inhalt des Attributs *punkte*) beinhaltet. Dieser wird dann mit *setText im StageObject punkteAnzeige* angezeigt (Schriftgröße 24).

Hier geht es zur Lösung auf Seite 32.

Rufen Sie jedes Mal, wenn ein Tor fällt die Methode *setzePunkte* mit 1 auf.

Testen Sie Ihr Programm.
Hier geht es zur Lösung auf Seite 34.

4.4 Der Spieler betritt das Feld

Der Spieler hat nun seinen Auftritt. Dazu muss die Klasse um zwei weitere Attribute erweitert werden.

Game
cx:Integer cy:Integer dx:Integer dy:Integer ball:StageObject punkte:Integer punkteAnzeige:StageObject spieler:StageObject sy:Integer
Game() resetBall() run() setzePunkte(aenderung:Integer)

Das Attribut *spieler* ist der Fußballspieler auf der *Stage*. Das Attribut *sy* gibt die vertikale Position des Fußballspielers auf dem Spielfeld an.

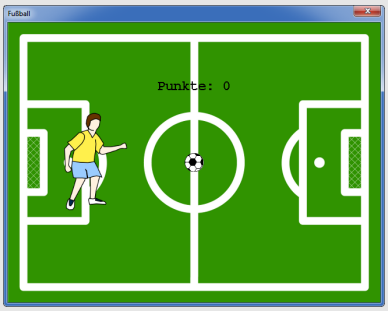
Aufgabe 10:

Erweitern Sie die Klasse *Game* um die neuen Attribute spieler und *sy*.

Erzeugen Sie im Konstruktor für das Attribut *sy* ein neues Objekt mit dem Wert 240.

Anschließend erzeugen Sie für das Attribut *spieler* ein Objekt. Das Bild für den Fußballspieler heißt *Fussballer.png* und soll mit einer Skalierung von 100 % angezeigt werden. Der Spieler ist an der Position (150, *sy*) auf dem Spielfeld zu platzieren.

Testen Sie Ihr Programm. Das Fenster sollte wie folgt aussehen:



Hier geht es zur Lösung auf Seite 36.

Fragen Sie in der *run* Methode der Klasse *Game* die Tasten <↑> (="UP") und <↓> (="DOWN") ab.

Wird die Taste <↑> gedrückt, so soll der Wert im Attribut *sy* um 10 reduziert werden und der Spieler an die Position (150, *sy*) verschoben werden.

Wird die Taste <↓> gedrückt, so soll der Wert im Attribut *sy* um 10 erhöht werden und der Spieler an die Position (150, *sy*) verschoben werden.

Testen Sie Ihr Programm. Der Spieler sollte sich jetzt mit den Tasten <↑> und <↓> nach oben und unten bewegen lassen.

Hier geht es zur Lösung auf Seite 38.

4.5 Training für den Spieler

Zwar kann der Spieler schon über den Platz laufen, aber er kann noch keinen Ball abwehren. Dazu muss geprüft werden, ob Ball und Spieler an der selben Stelle auf der *Stage* sind.

Kollisionsprüfung zweier *StageObject*:

Die Klasse *StageObject* besitzt eine Methode, mit der festgestellt werden kann, ob sich das *StageObject* mit einem anderen überschneidet.

```
Boolean StageObject.checkCollision(StageObject so)
```

Die Methode liefert *TRUE* zurück, wenn sich das *StageObject* mit dem übergebenen *StageObject* überschneidet.

Beispiel:

```
Boolean b;
b = spieler.checkCollision(ball);
```

Treffen Spieler und Ball aufeinander, so ist die x-Richtung (=*dx*) des Balls zu ändern.

Aufgabe 11:

Erweitern Sie in der Methode *run* in der Klasse *Game* die bedingte Anweisung, mit der die linke und rechte Spielfeldbegrenzung überprüft wird, um einen weiteren *if-else* Zweig.

Hier geht es zur Lösung auf Seite 40.

Ändern Sie die Bedingung für den neuen *if-else* Zweig so ab, dass geprüft wird, ob Spieler und Ball aufeinander treffen.

Hier geht es zur Lösung auf Seite 42.

Treffen Spieler und Ball aufeinander, so wird die x-Richtung des Balls geändert.

Testen Sie Ihr Programm.

Hier geht es zur Lösung auf Seite 44.

4.6 Die eigene Mannschaft muss auch mal verlieren

Bis jetzt kann der Spieler nur Punkte gewinnen. Jetzt wird das Spiel etwas schwieriger. Rollt der Ball an dem Spieler nach hinten vorbei, so ist die Runde beendet und der Punktestand wird um 1 reduziert.

Aufgabe 12:

Ändern Sie das Programm so ab, dass wenn der Ball die linke Spielfeldbegrenzung überschreitet der Punktestand um eins reduziert (Methode *setzePunkte*) wird und der Ball wieder in die Spielfeldmitte gelegt wird (Methode *resetBall*).

Testen Sie Ihr Programm.

Hier geht es zur Lösung auf Seite Fehler: Referenz nicht gefunden.

4.7 Ergänzungen zur Verschönerung

Zunächst wäre es schön, wenn sich das Spiel mit einem Tastendruck beenden ließe und nicht nur mit dem Abbrechen Buton in der Entwicklungsumgebung.

Aufgabe 13:

Fragen Sie in der *run* Methode der Klasse *Game* zusätzlich die Taste <q> (=quit) ab. Wird Sie gedrückt so soll die Methode

```
void Computer.sendExitSignal()
```

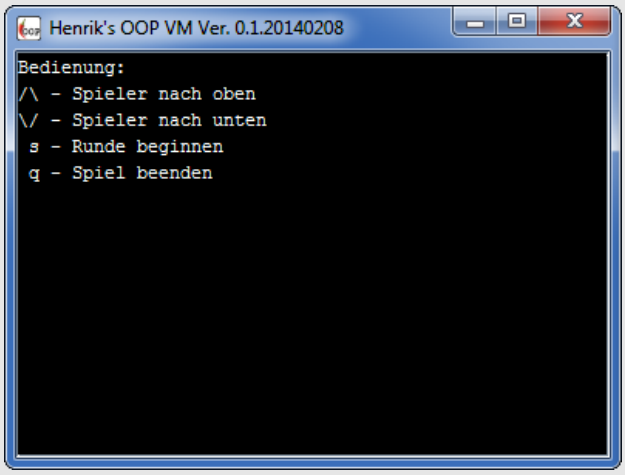
aufgerufen werden. Damit wird das Programm bendet.

Hier geht es zur Lösung auf Seite 48.

Außerdem wäre es schön, wenn auf der Konsole die Tasten zur Bedienung des Spiels beschrieben wären.

Aufgabe 14:

Geben Sie vor dem Aufruf der Methode *waitForExitSignal* in der *main* Methode der Klasse *Program* die Bedienungshilfe auf der Konsole aus. Die Ausgabe soll wie folgt aussehen:



Hier geht es zur Lösung auf Seite 51.

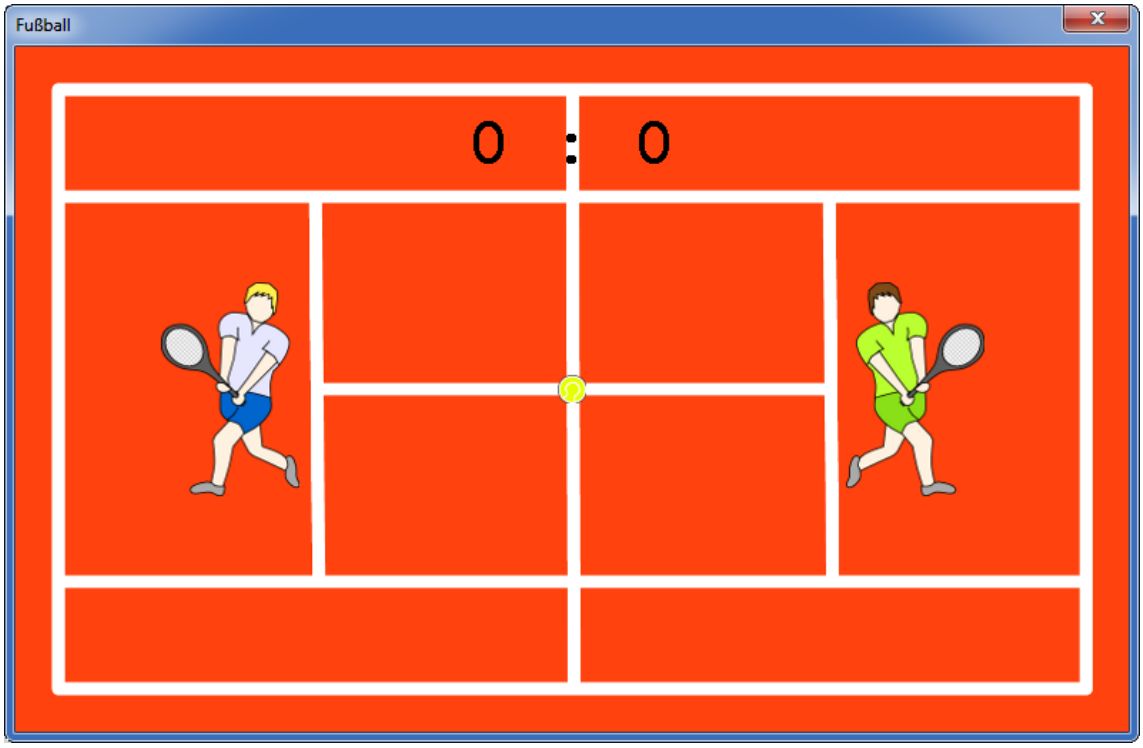
Endlich ist das Spiel fertig.

Spielen Sie ein paar Runden. Danach geht es auf den folgenden Seiten mit Übungen weiter.

5 Übungen

5.1 Ab auf den Tennisplatz

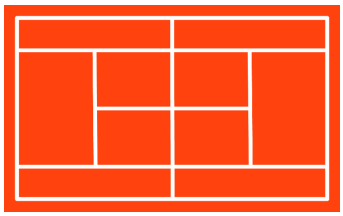
Das nächste Spiel ist für zwei Spieler.



Jeder Spieler muss den Ball abwehren. erreicht der Ball die linke Begrenzungslinie, so erhält der rechte Spieler einen Punkt. entsprechend erhält der linke Spieler einen Punkt wenn der Ball die rechte Begrenzungslinie erreicht.

Sie können das Spiel testen, wenn *Tennis.01.demo.oop* in die Entwicklungsumgebung laden und als Demo starten.

Realisieren Sie das beschriebene Spiel. Folgender Tabelle können die Daten für die Bilddateien entnommen werden.

Grafik	Dateiname	Größe in Pixel
	Tennisplatz.png	780×480

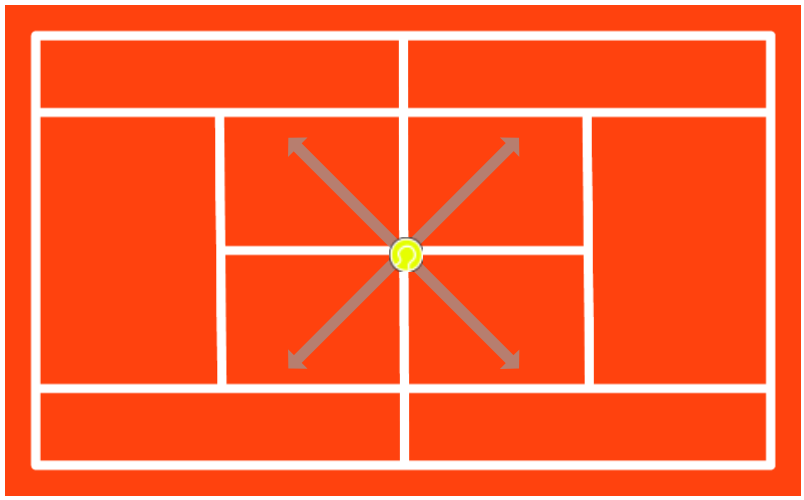
Grafik	Dateiname	Größe in Pixel
	Tennisspieler.links.png	
	Tennisspieler.rechts.png	
	Tennisball.png	

Der rote Rand um die äußere Spielfeldbegrenzung (äußere weiße Linie) hat eine Breite von 45 Pixel.

Der linke Spieler soll 150 Pixel vom rechten Rand der Grafik für den Tennisplatz platziert werden. Entsprechend wird der rechte Spieler 150 Pixel vom rechten Rande der Grafik für den Tennisplatz platziert.

5.2 Etwas mehr Gerechtigkeit

Das Tennisspiel aus Abschnitt 5.1 hat eine Ungerechtigkeit. Beim Start fliegt der Ball im in die Richtung des linken Spielers. Sorgen Sie für mehr Gerechtigkeit, in dem Sie das Programm aus Abschnitt 5.1 so abändern, das zufällig bestimmt wird in welche Richtung der Ball als erstes fliegt.



Sie können das Spiel testen, wenn *Tennis.02.demo.oop* in die Entwicklungsumgebung laden und als Demo starten.

ENDE

6 Lösungen zu den Aufgaben

6.1 Lösung zu Aufgabe 1

```
public class Game extends Stage
{
   // Anfang Attribute
   // Ende Attribute


   public Game ()
   {
      // Anfang Variablen
      // Ende Variablen

      super("Fußball",640,480);
      this.loadBackground("Fussballfeld.png");
   }
}
```

Zurück zur Aufgabe auf Seite 3.

6.2 Lösung zu Aufgabe 2

```
public class Game extends Stage
{
   // Anfang Attribute
   private Integer cx;
   private StageObject ball;
   private Integer cy;
   // Ende Attribute


   public Game ()
   {
      // Anfang Variablen
      // Ende Variablen

      super("Fußball",640,480);
      this.loadBackground("Fussballfeld.png");
      this.resetBall();
      ball = new StageObject("Ball",cx,cy,"Fussball.png",100);
      super.addStageObject(ball);
   }

   public void resetBall ()
   {
      // Anfang Variablen
      // Ende Variablen

      cx = new Integer(320);
      cy = new Integer(240);
   }
}
```

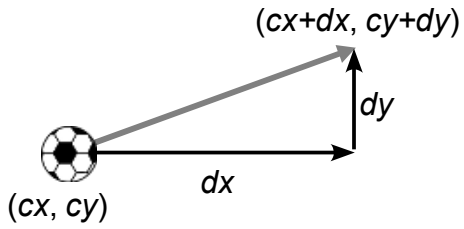
Zurück zur Aufgabe auf Seite 3.

6.3 Lösung zu Aufgabe 3

```
public class Game extends Stage
{
   // Anfang Attribute
   private Integer cx;
   private Integer dy;
   private Integer dx;
   private StageObject ball;
   private Integer cy;
   // Ende Attribute


   public Game ()
   {
      // Anfang Variablen
      // Ende Variablen

      super("Fußball",640,480);
      this.loadBackground("Fussballfeld.png");
      this.resetBall();
      ball = new StageObject("Ball",cx,cy,"Fussball.png",100);
      super.addStageObject(ball);
   }

   public void run ()
   {
      // Anfang Variablen
      // Ende Variablen

   }

   public void resetBall ()
   {
      // Anfang Variablen
      // Ende Variablen

      cx = new Integer(320);
      cy = new Integer(240);
      dx = new Integer();
      dy = new Integer();
   }
}
```

Zurück zur Aufgabe auf Seite 4.

6.4 Tipp zu Aufgabe 4



Die neue Position des Balls ist (*cx*+*dx*, cy+dy).

Zurück zur Aufgabe auf Seite 5.

6.5 Lösung zu Aufgabe 4

```
public class Game extends Stage
{
   // Anfang Attribute
   private Integer cx;
   private Integer dy;
   private Computer comp;
   private Integer dx;
   private StageObject ball;
   private Integer cy;
   // Ende Attribute


   public Game ()
   {
      // Anfang Variablen
      // Ende Variablen

      super("Fußball",640,480);
      this.loadBackground("Fussballfeld.png");
      this.resetBall();
      ball = new StageObject("Ball",cx,cy,"Fussball.png",100);
      super.addStageObject(ball);
      comp = new Computer();
   }

   public void run ()
   {
      // Anfang Variablen
      // Ende Variablen

      if (super.isKeyPressed("s"))
      {
         dx = comp.rand(-10,-5);
         dy = comp.rand(5,10);
      }
      cx = cx.add(dx);
      cy = cy.add(dy);
      ball.moveToPosition(cx,cy);
   }

   public void resetBall ()
   {
      // Anfang Variablen
      // Ende Variablen

      cx = new Integer(320);
      cy = new Integer(240);
      dx = new Integer();
      dy = new Integer();
   }
}
```

```
public class Program
{
   // Anfang Attribute
   // Ende Attribute


   public Program ()
   {
      // Anfang Variablen
      // Ende Variablen

   }

   public void main ()
   {
      // Anfang Variablen
      Game game;
      Computer comp;
      // Ende Variablen

      game = new Game();
      game.startSingle(5);
      comp = new Computer();
      comp.waitForExitSignal();
   }
}
```

Zurück zur Aufgabe auf Seite 5.

6.6 Lösung zu Aufgabe 5

```
public class Game extends Stage
{
   // Anfang Attribute
   private Integer cx;
   private Integer dy;
   private Computer comp;
   private Integer dx;
   private StageObject ball;
   private Integer cy;
   // Ende Attribute

   public Game ()
   {
      // Anfang Variablen
      // Ende Variablen
      super("Fußball",640,480);
      this.loadBackground("Fussballfeld.png");
      this.resetBall();
      ball = new StageObject("Ball",cx,cy,"Fussball.png",100);
      super.addStageObject(ball);
      comp = new Computer();
   }

   public void run ()
   {
      // Anfang Variablen
      // Ende Variablen
      if (cx.lowerThanOrEqual(45) || cx.greaterThanOrEqual(565))
      {
      }
      if (cy.lowerThanOrEqual(45) || cy.greaterThanOrEqual(435))
      {
      }
      if (super.isKeyPressed("s"))
      {
         dx = comp.rand(-10,-5);
         dy = comp.rand(5,10);
      }
      cx = cx.add(dx);
      cy = cy.add(dy);
      ball.moveToPosition(cx,cy);
   }

   public void resetBall ()
   {
      // Anfang Variablen
      // Ende Variablen
      cx = new Integer(320);
      cy = new Integer(240);
      dx = new Integer();
      dy = new Integer();
   }
}
```

Zurück zur Aufgabe auf Seite 6.

6.7 Lösung zu Aufgabe 6

```
public class Game extends Stage
{
   // Anfang Attribute
   private Integer cx;
   private Integer dy;
   private Computer comp;
   private Integer dx;
   private StageObject ball;
   private Integer cy;
   // Ende Attribute


   public Game ()
   {
      // Anfang Variablen
      // Ende Variablen

      super("Fußball",640,480);
      this.loadBackground("Fussballfeld.png");
      this.resetBall();
      ball = new StageObject("Ball",cx,cy,"Fussball.png",100);
      super.addStageObject(ball);
      comp = new Computer();
   }

   public void run ()
   {
      // Anfang Variablen
      // Ende Variablen

      if (cx.lowerThanOrEqual(45) || cx.greaterThanOrEqual(565))
      {
         dx = dx.mul(-1);
      }
      if (cy.lowerThanOrEqual(45) || cy.greaterThanOrEqual(435))
      {
         dy = dy.mul(-1);
      }
      if (super.isKeyPressed("s"))
      {
         dx = comp.rand(-10,-5);
         dy = comp.rand(5,10);
      }
      cx = cx.add(dx);
      cy = cy.add(dy);
      ball.moveToPosition(cx,cy);
   }
```

```
   public void resetBall ()
   {
      // Anfang Variablen
      // Ende Variablen

      cx = new Integer(320);
      cy = new Integer(240);
      dx = new Integer();
      dy = new Integer();
   }
}
```

Zurück zur Aufgabe auf Seite 7.

6.8 Lösung zu Aufgabe 7

```
public class Game extends Stage
{
   // Anfang Attribute
   private Integer cx;
   private Integer dy;
   private Computer comp;
   private Integer dx;
   private StageObject ball;
   private Integer cy;
   // Ende Attribute


   public Game ()
   {
      // Anfang Variablen
      // Ende Variablen

      super("Fußball",640,480);
      this.loadBackground("Fussballfeld.png");
      this.resetBall();
      ball = new StageObject("Ball",cx,cy,"Fussball.png",100);
      super.addStageObject(ball);
      comp = new Computer();
   }

   public void run ()
   {
      // Anfang Variablen
      // Ende Variablen

      if (cx.lowerThanOrEqual(45))
      {
         dx = dx.mul(-1);
      }
      else if (cx.greaterThanOrEqual (565))
      {
      }
      if (cy.lowerThanOrEqual(45) || cy.greaterThanOrEqual(435))
      {
         dy = dy.mul(-1);
      }
      if (super.isKeyPressed("s"))
      {
         dx = comp.rand(-10,-5);
         dy = comp.rand(5,10);
      }
      cx = cx.add(dx);
      cy = cy.add(dy);
      ball.moveToPosition(cx,cy);
   }
```

```
   public void resetBall ()
   {
      // Anfang Variablen
      // Ende Variablen

      cx = new Integer(320);
      cy = new Integer(240);
      dx = new Integer();
      dy = new Integer();
   }
}
```

Zurück zur Aufgabe auf Seite 9.

6.9 Lösung zu Aufgabe 8

```
public class Game extends Stage
{
   // Anfang Attribute
   private Integer cx;
   private Integer dy;
   private Computer comp;
   private Integer dx;
   private StageObject ball;
   private Integer cy;
   // Ende Attribute


   public Game ()
   {
      // Anfang Variablen
      // Ende Variablen

      super("Fußball",640,480);
      this.loadBackground("Fussballfeld.png");
      this.resetBall();
      ball = new StageObject("Ball",cx,cy,"Fussball.png",100);
      super.addStageObject(ball);
      comp = new Computer();
   }

   public void run ()
   {
      // Anfang Variablen
      // Ende Variablen

      if (cx.lowerThanOrEqual(45))
      {
         dx = dx.mul(-1);
      }
      else if (cx.greaterThanOrEqual(565))
      {
         if (cy.greaterThanOrEqual(160) && cy.lowerThanOrEqual(320))
         {
            this.resetBall();
         }
         else
         {
            dx = dx.mul(-1);
         }
      }
```

```
      if (cy.lowerThanOrEqual(45) || cy.greaterThanOrEqual(435))
      {
         dy = dy.mul(-1);
      }
      if (super.isKeyPressed("s"))
      {
         dx = comp.rand(-10,-5);
         dy = comp.rand(5,10);
      }
      cx = cx.add(dx);
      cy = cy.add(dy);
      ball.moveToPosition(cx,cy);
   }

   public void resetBall ()
   {
      // Anfang Variablen
      // Ende Variablen

      cx = new Integer(320);
      cy = new Integer(240);
      dx = new Integer();
      dy = new Integer();
   }
}
```

Zurück zur Aufgabe auf Seite 9.

6.10 Lösung zu Aufgabe 9 (a)

```
public class Game extends Stage
{
   // Anfang Attribute
   private StageObject punkteAnzeige;
   private Integer cx;
   private Integer dy;
   private Computer comp;
   private Integer dx;
   private Integer punkte;
   private StageObject ball;
   private Integer cy;
   // Ende Attribute


   public Game ()
   {
      // Anfang Variablen
      // Ende Variablen

      super("Fußball",640,480);
      this.loadBackground("Fussballfeld.png");
      this.resetBall();
      ball = new StageObject("Ball",cx,cy,"Fussball.png",100);
      super.addStageObject(ball);
      comp = new Computer();
      punkteAnzeige = new StageObject(320,100,100);
      punkteAnzeige.setTextAlignCenter();
      super.addStageObject(punkteAnzeige);
      punkte = new Integer();
      this.setzePunkte(0);
   }

   public void run ()
   {
      // Anfang Variablen
      // Ende Variablen

      if (cx.lowerThanOrEqual(45))
      {
         dx = dx.mul(-1);
      }
      else if (cx.greaterThanOrEqual(565))
      {
         if (cy.greaterThanOrEqual(160) && cy.lowerThanOrEqual(320))
         {
            this.resetBall();
         }
         else
         {
            dx = dx.mul(-1);
         }
      }
```

```
      if (cy.lowerThanOrEqual(45) || cy.greaterThanOrEqual(435))
      {
         dy = dy.mul(-1);
      }
      if (super.isKeyPressed("s"))
      {
         dx = comp.rand(-10,-5);
         dy = comp.rand(5,10);
      }
      cx = cx.add(dx);
      cy = cy.add(dy);
      ball.moveToPosition(cx,cy);
   }

   public void setzePunkte (Integer aenderung)
   {
      // Anfang Variablen
      // Ende Variablen

   }

   public void resetBall ()
   {
      // Anfang Variablen
      // Ende Variablen

      cx = new Integer(320);
      cy = new Integer(240);
      dx = new Integer();
      dy = new Integer();
   }
}
```

Zurück zur Aufgabe auf Seite 10.

6.11 Lösung zu Aufgabe 9 (b)

```
public class Game extends Stage
{
   // Anfang Attribute
   private StageObject punkteAnzeige;
   private Integer cx;
   private Integer dy;
   private Computer comp;
   private Integer dx;
   private Integer punkte;
   private StageObject ball;
   private Integer cy;
   // Ende Attribute


   public Game ()
   {
      // Anfang Variablen
      // Ende Variablen

      super("Fußball",640,480);
      this.loadBackground("Fussballfeld.png");
      this.resetBall();
      ball = new StageObject("Ball",cx,cy,"Fussball.png",100);
      super.addStageObject(ball);
      comp = new Computer();
      punkteAnzeige = new StageObject(320,100,100);
      punkteAnzeige.setTextAlignCenter();
      super.addStageObject(punkteAnzeige);
      punkte = new Integer();
      this.setzePunkte(0);
   }

   public void run ()
   {
      // Anfang Variablen
      // Ende Variablen

      if (cx.lowerThanOrEqual(45))
      {
         dx = dx.mul(-1);
      }
      else if (cx.greaterThanOrEqual(565))
      {
         if (cy.greaterThanOrEqual(160) && cy.lowerThanOrEqual(320))
         {
            this.resetBall();
         }
         else
         {
            dx = dx.mul(-1);
         }
      }
```

```
      if (cy.lowerThanOrEqual(45) || cy.greaterThanOrEqual(435))
      {
         dy = dy.mul(-1);
      }
      if (super.isKeyPressed("s"))
      {
         dx = comp.rand(-10,-5);
         dy = comp.rand(5,10);
      }
      cx = cx.add(dx);
      cy = cy.add(dy);
      ball.moveToPosition(cx,cy);
   }

   public void setzePunkte (Integer aenderung)
   {
      // Anfang Variablen
      String msg;
      // Ende Variablen

      punkte = punkte.add(aenderung);
      msg = new String("Punkte: ");
      msg.append(punkte);
      punkteAnzeige.setText(msg,24);
   }

   public void resetBall ()
   {
      // Anfang Variablen
      // Ende Variablen

      cx = new Integer(320);
      cy = new Integer(240);
      dx = new Integer();
      dy = new Integer();
   }
}
```

Zurück zur Aufgabe auf Seite 10.

6.12 Lösung zu Aufgabe 9 (c)

```
public class Game extends Stage
{
   // Anfang Attribute
   private StageObject punkteAnzeige;
   private Integer cx;
   private Integer dy;
   private Computer comp;
   private Integer dx;
   private Integer punkte;
   private StageObject ball;
   private Integer cy;
   // Ende Attribute


   public Game ()
   {
      // Anfang Variablen
      // Ende Variablen

      super("Fußball",640,480);
      this.loadBackground("Fussballfeld.png");
      this.resetBall();
      ball = new StageObject("Ball",cx,cy,"Fussball.png",100);
      super.addStageObject(ball);
      comp = new Computer();
      punkteAnzeige = new StageObject(320,100,100);
      punkteAnzeige.setTextAlignCenter();
      super.addStageObject(punkteAnzeige);
      punkte = new Integer();
      this.setzePunkte(0);
   }

   public void run ()
   {
      // Anfang Variablen
      // Ende Variablen

      if (cx.lowerThanOrEqual(45))
      {
         dx = dx.mul(-1);
      }
      else if (cx.greaterThanOrEqual(565))
      {
         if (cy.greaterThanOrEqual(160) && cy.lowerThanOrEqual(320))
         {
            this.resetBall();
            this.setzePunkte(1);
         }
         else
         {
            dx = dx.mul(-1);
         }
      }
```

```
      if (cy.lowerThanOrEqual(45) || cy.greaterThanOrEqual(435))
      {
         dy = dy.mul(-1);
      }
      if (super.isKeyPressed("s"))
      {
         dx = comp.rand(-10,-5);
         dy = comp.rand(5,10);
      }
      cx = cx.add(dx);
      cy = cy.add(dy);
      ball.moveToPosition(cx,cy);
   }

   public void setzePunkte (Integer aenderung)
   {
      // Anfang Variablen
      String msg;
      // Ende Variablen

      punkte = punkte.add(aenderung);
      msg = new String("Punkte: ");
      msg.append(punkte);
      punkteAnzeige.setText(msg,24);
   }

   public void resetBall ()
   {
      // Anfang Variablen
      // Ende Variablen

      cx = new Integer(320);
      cy = new Integer(240);
      dx = new Integer();
      dy = new Integer();
   }
}
```

Zurück zur Aufgabe auf Seite 10.

6.13 Lösung zu Aufgabe 10

```
public class Game extends Stage
{
   // Anfang Attribute
   private StageObject ball;
   private Integer cy;
   private Integer cx;
   private Computer comp;
   private Integer sy;
   private StageObject spieler;
   private Integer dy;
   private Integer punkte;
   private Integer dx;
   private StageObject punkteAnzeige;
   // Ende Attribute


   public Game ()
   {
      // Anfang Variablen
      // Ende Variablen

      super("Fußball",640,480);
      this.loadBackground("Fussballfeld.png");
      this.resetBall();
      ball = new StageObject("Ball",cx,cy,"Fussball.png",100);
      super.addStageObject(ball);
      comp = new Computer();
      punkteAnzeige = new StageObject(320,100,100);
      punkteAnzeige.setTextAlignCenter();
      super.addStageObject(punkteAnzeige);
      punkte = new Integer();
      this.setzePunkte(0);
      sy = new Integer(240);
      spieler = new StageObject("Spieler",150,sy,
                                "Fussballer.png",100);
      super.addStageObject(spieler);
   }

   public void run ()
   {
      // Anfang Variablen
      // Ende Variablen
```

```
      if (cx.lowerThanOrEqual(45))
      {
         dx = dx.mul(-1);
      }
      else if (cx.greaterThanOrEqual(565))
      {
         if (cy.greaterThanOrEqual(160) && cy.lowerThanOrEqual(320))
         {
            this.resetBall();
            this.setzePunkte(1);
         }
         else
         {
            dx = dx.mul(-1);
         }
      }
      if (cy.lowerThanOrEqual(45) || cy.greaterThanOrEqual(435))
      {
         dy = dy.mul(-1);
      }
      if (super.isKeyPressed("s"))
      {
         dx = comp.rand(-10,-5);
         dy = comp.rand(5,10);
      }
      cx = cx.add(dx);
      cy = cy.add(dy);
      ball.moveToPosition(cx,cy);
   }

   public void setzePunkte (Integer aenderung)
   {
      // Anfang Variablen
      String msg;
      // Ende Variablen

      punkte = punkte.add(aenderung);
      msg = new String("Punkte: ");
      msg.append(punkte);
      punkteAnzeige.setText(msg,24);
   }

   public void resetBall ()
   {
      // Anfang Variablen
      // Ende Variablen

      cx = new Integer(320);
      cy = new Integer(240);
      dx = new Integer();
      dy = new Integer();
   }
}
```

Zurück zur Aufgabe auf Seite 11.

6.14 Lösung zu Aufgabe 10

```
public class Game extends Stage
{
   // Anfang Attribute
   private StageObject ball;
   private Integer cy;
   private Integer cx;
   private Computer comp;
   private Integer sy;
   private StageObject spieler;
   private Integer dy;
   private Integer punkte;
   private Integer dx;
   private StageObject punkteAnzeige;
   // Ende Attribute


   public Game ()
   {
      // Anfang Variablen
      // Ende Variablen

      super("Fußball",640,480);
      this.loadBackground("Fussballfeld.png");
      this.resetBall();
      ball = new StageObject("Ball",cx,cy,"Fussball.png",100);
      super.addStageObject(ball);
      comp = new Computer();
      punkteAnzeige = new StageObject(320,100,100);
      punkteAnzeige.setTextAlignCenter();
      super.addStageObject(punkteAnzeige);
      punkte = new Integer();
      this.setzePunkte(0);
      sy = new Integer(240);
      spieler = new StageObject("Spieler",150,sy,
                                "Fussballer.png",100);
      super.addStageObject(spieler);
   }

   public void run ()
   {
      // Anfang Variablen
      // Ende Variablen

      if (cx.lowerThanOrEqual(45))
      {
         dx = dx.mul(-1);
      }
      else if (cx.greaterThanOrEqual(565))
      {
         if (cy.greaterThanOrEqual(160) && cy.lowerThanOrEqual(320))
         {
            this.resetBall();
            this.setzePunkte(1);
         }
```

```
         else
         {
            dx = dx.mul(-1);
         }
      }
      if (cy.lowerThanOrEqual(45) || cy.greaterThanOrEqual(435))
      {
         dy = dy.mul(-1);
      }

      if (super.isKeyPressed("s"))
      {
         dx = comp.rand(-10,-5);
         dy = comp.rand(5,10);
      }
      else if (super.isKeyPressed("UP"))
      {
         sy = sy.sub(10);
         spieler.moveToPosition(150,sy);
      }
      else if (super.isKeyPressed("DOWN"))
      {
         sy = sy.add(10);
         spieler.moveToPosition(150,sy);
      }

      cx = cx.add(dx);
      cy = cy.add(dy);
      ball.moveToPosition(cx,cy);
   }

   public void setzePunkte (Integer aenderung)
   {
      // Anfang Variablen
      String msg;
      // Ende Variablen

      punkte = punkte.add(aenderung);
      msg = new String("Punkte: ");
      msg.append(punkte);
      punkteAnzeige.setText(msg,24);
   }

   public void resetBall ()
   {
      // Anfang Variablen
      // Ende Variablen

      cx = new Integer(320);
      cy = new Integer(240);
      dx = new Integer();
      dy = new Integer();
   }
}
```

Zurück zur Aufgabe auf Seite 11.

6.15 Lösung zu Aufgabe 11 (a)

```
public class Game extends Stage
{
   // Anfang Attribute
   private StageObject ball;
   private Integer cy;
   private Integer cx;
   private Computer comp;
   private Integer sy;
   private StageObject spieler;
   private Integer dy;
   private Integer punkte;
   private Integer dx;
   private StageObject punkteAnzeige;
   // Ende Attribute


   public Game ()
   {
      // Anfang Variablen
      // Ende Variablen

      super("Fußball",640,480);
      this.loadBackground("Fussballfeld.png");
      this.resetBall();
      ball = new StageObject("Ball",cx,cy,"Fussball.png",100);
      super.addStageObject(ball);
      comp = new Computer();
      punkteAnzeige = new StageObject(320,100,100);
      punkteAnzeige.setTextAlignCenter();
      super.addStageObject(punkteAnzeige);
      punkte = new Integer();
      this.setzePunkte(0);
      sy = new Integer(240);
      spieler = new StageObject("Spieler",150,sy,
                                "Fussballer.png",100);
      super.addStageObject(spieler);
   }

   public void run ()
   {
      // Anfang Variablen
      // Ende Variablen

      if (cx.lowerThanOrEqual(45))
      {
         dx = dx.mul(-1);
      }
      else if (cx.greaterThanOrEqual(565))
      {
         if (cy.greaterThanOrEqual(160) && cy.lowerThanOrEqual(320))
         {
            this.resetBall();
            this.setzePunkte(1);
         }
         else
```

```
            {
               dx = dx.mul(-1);
            }
         }
         else if (ball.checkCollision(NULL))
         {
         }

         if (cy.lowerThanOrEqual(45) || cy.greaterThanOrEqual(435))
         {
            dy = dy.mul(-1);
         }
         if (super.isKeyPressed("s"))
         {
            dx = comp.rand(-10,-5);
            dy = comp.rand(5,10);
         }
         else if (super.isKeyPressed("UP"))
         {
            sy = sy.sub(10);
            spieler.moveToPosition(150,sy);
         }
         else if (super.isKeyPressed("DOWN"))
         {
            sy = sy.add(10);
            spieler.moveToPosition(150,sy);
         }
         cx = cx.add(dx);
         cy = cy.add(dy);
         ball.moveToPosition(cx,cy);
      }

      public void setzePunkte (Integer aenderung)
      {
         // Anfang Variablen
         String msg;
         // Ende Variablen

         punkte = punkte.add(aenderung);
         msg = new String("Punkte: ");
         msg.append(punkte);
         punkteAnzeige.setText(msg,24);
      }

      public void resetBall ()
      {
         // Anfang Variablen
         // Ende Variablen

         cx = new Integer(320);
         cy = new Integer(240);
         dx = new Integer();
         dy = new Integer();
      }
   }
```

Zurück zur Aufgabe auf Seite 12.

6.16 Lösung zu Aufgabe 11 (b)

```
public class Game extends Stage
{
   // Anfang Attribute
   private StageObject ball;
   private Integer cy;
   private Integer cx;
   private Computer comp;
   private Integer sy;
   private StageObject spieler;
   private Integer dy;
   private Integer punkte;
   private Integer dx;
   private StageObject punkteAnzeige;
   // Ende Attribute


   public Game ()
   {
      // Anfang Variablen
      // Ende Variablen

      super("Fußball",640,480);
      this.loadBackground("Fussballfeld.png");
      this.resetBall();
      ball = new StageObject("Ball",cx,cy,"Fussball.png",100);
      super.addStageObject(ball);
      comp = new Computer();
      punkteAnzeige = new StageObject(320,100,100);
      punkteAnzeige.setTextAlignCenter();
      super.addStageObject(punkteAnzeige);
      punkte = new Integer();
      this.setzePunkte(0);
      sy = new Integer(240);
      spieler = new StageObject("Spieler",150,sy,
                                "Fussballer.png",100);
      super.addStageObject(spieler);
   }

   public void run ()
   {
      // Anfang Variablen
      // Ende Variablen

      if (cx.lowerThanOrEqual(45))
      {
         dx = dx.mul(-1);
      }
      else if (cx.greaterThanOrEqual(565))
      {
         if (cy.greaterThanOrEqual(160) && cy.lowerThanOrEqual(320))
         {
            this.resetBall();
            this.setzePunkte(1);
         }
         else
```

```
            {
               dx = dx.mul(-1);
            }
         }
         else if (spieler.checkCollision(ball))
         {
         }

         if (cy.lowerThanOrEqual(45) || cy.greaterThanOrEqual(435))
         {
            dy = dy.mul(-1);
         }
         if (super.isKeyPressed("s"))
         {
            dx = comp.rand(-10,-5);
            dy = comp.rand(5,10);
         }
         else if (super.isKeyPressed("UP"))
         {
            sy = sy.sub(10);
            spieler.moveToPosition(150,sy);
         }
         else if (super.isKeyPressed("DOWN"))
         {
            sy = sy.add(10);
            spieler.moveToPosition(150,sy);
         }
         cx = cx.add(dx);
         cy = cy.add(dy);
         ball.moveToPosition(cx,cy);
      }

      public void setzePunkte (Integer aenderung)
      {
         // Anfang Variablen
         String msg;
         // Ende Variablen

         punkte = punkte.add(aenderung);
         msg = new String("Punkte: ");
         msg.append(punkte);
         punkteAnzeige.setText(msg,24);
      }

      public void resetBall ()
      {
         // Anfang Variablen
         // Ende Variablen

         cx = new Integer(320);
         cy = new Integer(240);
         dx = new Integer();
         dy = new Integer();
      }
   }
```

Zurück zur Aufgabe auf Seite 12.

6.17 Lösung zu Aufgabe 11 (c)

```
public class Game extends Stage
{
   // Anfang Attribute
   private StageObject ball;
   private Integer cy;
   private Integer cx;
   private Computer comp;
   private Integer sy;
   private StageObject spieler;
   private Integer dy;
   private Integer punkte;
   private Integer dx;
   private StageObject punkteAnzeige;
   // Ende Attribute


   public Game ()
   {
      // Anfang Variablen
      // Ende Variablen

      super("Fußball",640,480);
      this.loadBackground("Fussballfeld.png");
      this.resetBall();
      ball = new StageObject("Ball",cx,cy,"Fussball.png",100);
      super.addStageObject(ball);
      comp = new Computer();
      punkteAnzeige = new StageObject(320,100,100);
      punkteAnzeige.setTextAlignCenter();
      super.addStageObject(punkteAnzeige);
      punkte = new Integer();
      this.setzePunkte(0);
      sy = new Integer(240);
      spieler = new StageObject("Spieler",150,sy,
                                "Fussballer.png",100);
      super.addStageObject(spieler);
   }

   public void run ()
   {
      // Anfang Variablen
      // Ende Variablen

      if (cx.lowerThanOrEqual(45))
      {
         dx = dx.mul(-1);
      }
      else if (cx.greaterThanOrEqual(565))
      {
         if (cy.greaterThanOrEqual(160) && cy.lowerThanOrEqual(320))
         {
            this.resetBall();
            this.setzePunkte(1);
         }
         else
```

```
            {
               dx = dx.mul(-1);
            }
         }
         else if (spieler.checkCollision(ball))
         {
            dx = dx.mul(-1);
         }

         if (cy.lowerThanOrEqual(45) || cy.greaterThanOrEqual(435))
         {
            dy = dy.mul(-1);
         }
         if (super.isKeyPressed("s"))
         {
            dx = comp.rand(-10,-5);
            dy = comp.rand(5,10);
         }
         else if (super.isKeyPressed("UP"))
         {
            sy = sy.sub(10);
            spieler.moveToPosition(150,sy);
         }
         else if (super.isKeyPressed("DOWN"))
         {
            sy = sy.add(10);
            spieler.moveToPosition(150,sy);
         }
         cx = cx.add(dx);
         cy = cy.add(dy);
         ball.moveToPosition(cx,cy);
      }

      public void setzePunkte (Integer aenderung)
      {
         // Anfang Variablen
         String msg;
         // Ende Variablen

         punkte = punkte.add(aenderung);
         msg = new String("Punkte: ");
         msg.append(punkte);
         punkteAnzeige.setText(msg,24);
      }

      public void resetBall ()
      {
         // Anfang Variablen
         // Ende Variablen

         cx = new Integer(320);
         cy = new Integer(240);
         dx = new Integer();
         dy = new Integer();
      }
   }
```

Zurück zur Aufgabe auf Seite 12.

6.18 Lösung zu Aufgabe 12

```
public class Game extends Stage
{
   // Anfang Attribute
   private StageObject ball;
   private Integer cy;
   private Integer cx;
   private Computer comp;
   private Integer sy;
   private StageObject spieler;
   private Integer dy;
   private Integer punkte;
   private Integer dx;
   private StageObject punkteAnzeige;
   // Ende Attribute


   public Game ()
   {
      // Anfang Variablen
      // Ende Variablen

      super("Fußball",640,480);
      this.loadBackground("Fussballfeld.png");
      this.resetBall();
      ball = new StageObject("Ball",cx,cy,"Fussball.png",100);
      super.addStageObject(ball);
      comp = new Computer();
      punkteAnzeige = new StageObject(320,100,100);
      punkteAnzeige.setTextAlignCenter();
      super.addStageObject(punkteAnzeige);
      punkte = new Integer();
      this.setzePunkte(0);
      sy = new Integer(240);
      spieler = new StageObject("Spieler",150,sy,
                                "Fussballer.png",100);
      super.addStageObject(spieler);
   }

   public void run ()
   {
      // Anfang Variablen
      // Ende Variablen

      if (cx.lowerThanOrEqual(45))
      {
         this.setzePunkte(-1);
         this.resetBall();
      }
      else if (cx.greaterThanOrEqual(565))
      {
         if (cy.greaterThanOrEqual(160) && cy.lowerThanOrEqual(320))
         {
            this.resetBall();
            this.setzePunkte(1);
         }
```

```
            else
            {
               dx = dx.mul(-1);
            }
         }
         else if (spieler.checkCollision(ball))
         {
            dx = dx.mul(-1);
         }
         if (cy.lowerThanOrEqual(45) || cy.greaterThanOrEqual(435))
         {
            dy = dy.mul(-1);
         }
         if (super.isKeyPressed("s"))
         {
            dx = comp.rand(-10,-5);
            dy = comp.rand(5,10);
         }
         else if (super.isKeyPressed("UP"))
         {
            sy = sy.sub(10);
            spieler.moveToPosition(150,sy);
         }
         else if (super.isKeyPressed("DOWN"))
         {
            sy = sy.add(10);
            spieler.moveToPosition(150,sy);
         }
         cx = cx.add(dx);
         cy = cy.add(dy);
         ball.moveToPosition(cx,cy);
      }

      public void setzePunkte (Integer aenderung)
      {
         // Anfang Variablen
         String msg;
         // Ende Variablen

         punkte = punkte.add(aenderung);
         msg = new String("Punkte: ");
         msg.append(punkte);
         punkteAnzeige.setText(msg,24);
      }

      public void resetBall ()
      {
         // Anfang Variablen
         // Ende Variablen

         cx = new Integer(320);
         cy = new Integer(240);
         dx = new Integer();
         dy = new Integer();
      }
   }
```

Zurück zur Aufgabe auf Seite 13.

6.19 Lösung zu Aufgabe 13

```
public class Game extends Stage
{
   // Anfang Attribute
   private StageObject ball;
   private Integer cy;
   private Integer cx;
   private Computer comp;
   private Integer sy;
   private StageObject spieler;
   private Integer dy;
   private Integer punkte;
   private Integer dx;
   private StageObject punkteAnzeige;
   // Ende Attribute


   public Game ()
   {
      // Anfang Variablen
      // Ende Variablen

      super("Fußball",640,480);
      this.loadBackground("Fussballfeld.png");
      this.resetBall();
      ball = new StageObject("Ball",cx,cy,"Fussball.png",100);
      super.addStageObject(ball);
      comp = new Computer();
      punkteAnzeige = new StageObject(320,100,100);
      punkteAnzeige.setTextAlignCenter();
      super.addStageObject(punkteAnzeige);
      punkte = new Integer();
      this.setzePunkte(0);
      sy = new Integer(240);
      spieler = new StageObject("Spieler",150,sy,
                                "Fussballer.png",100);
      super.addStageObject(spieler);
   }

   public void run ()
   {
      // Anfang Variablen
      // Ende Variablen

      if (cx.lowerThanOrEqual(45))
      {
         this.setzePunkte(-1);
         this.resetBall();
      }
      else if (cx.greaterThanOrEqual(565))
      {
         if (cy.greaterThanOrEqual(160) && cy.lowerThanOrEqual(320))
         {
            this.resetBall();
            this.setzePunkte(1);
         }
```

```
            else
            {
               dx = dx.mul(-1);
            }
         }
         else if (spieler.checkCollision(ball))
         {
            dx = dx.mul(-1);
         }
         if (cy.lowerThanOrEqual(45) || cy.greaterThanOrEqual(435))
         {
            dy = dy.mul(-1);
         }

         if (super.isKeyPressed("s"))
         {
            dx = comp.rand(-10,-5);
            dy = comp.rand(5,10);
         }
         else if (super.isKeyPressed("UP"))
         {
            sy = sy.sub(10);
            spieler.moveToPosition(150,sy);
         }
         else if (super.isKeyPressed("DOWN"))
         {
            sy = sy.add(10);
            spieler.moveToPosition(150,sy);
         }
         else if (super.isKeyPressed("q"))
         {
            comp.sendExitSignal();
         }

         cx = cx.add(dx);
         cy = cy.add(dy);
         ball.moveToPosition(cx,cy);
      }

      public void setzePunkte (Integer aenderung)
      {
         // Anfang Variablen
         String msg;
         // Ende Variablen

         punkte = punkte.add(aenderung);
         msg = new String("Punkte: ");
         msg.append(punkte);
         punkteAnzeige.setText(msg,24);
      }
```

```
   public void resetBall ()
   {
      // Anfang Variablen
      // Ende Variablen

      cx = new Integer(320);
      cy = new Integer(240);
      dx = new Integer();
      dy = new Integer();
   }
}
```

Zurück zur Aufgabe auf Seite 13.

6.20 Lösung zu Aufgabe 14

```
public class Program
{
   // Anfang Attribute
   // Ende Attribute


   public Program ()
   {
      // Anfang Variablen
      // Ende Variablen

   }

   public void main ()
   {
      // Anfang Variablen
      Game game;
      Computer comp;
      // Ende Variablen

      game = new Game();
      game.startSingle(5);
      comp = new Computer();
      comp.print("Bedienung:");
      comp.println();
      comp.print("/\ - Spieler nach oben");
      comp.println();
      comp.print("\/ - Spieler nach unten");
      comp.println();
      comp.print(" s - Runde beginnen");
      comp.println();
      comp.print(" q - Spiel beenden");
      comp.waitForExitSignal();
   }
}
```

Zurück zur Aufgabe auf Seite 14.