

OOP

Felder von Objekten

Henrik Horstmann

23. September 2014

Inhaltsverzeichnis

1 Bedeutung der Symbole.....	1
2 Was ist eine Feldvariable?.....	2
2.1 Feldvariablen deklarieren.....	2
2.2 Anlegen einer Feldvariable.....	3
2.3 Schubladen öffnen.....	4
3 Mäuse fangen.....	5
3.1 Das Spielfeld.....	5
3.2 Die Katze betritt die Bühne.....	6
3.3 Die Katze wird hungrig.....	8
3.4 Der Erfolg der Katze.....	12
3.5 Das Spiel beenden.....	13
4 Übungen.....	14
4.1 Flieger Rallye.....	14
4.2 Wo ist die Münze?.....	18
5 Lösungen zu den Aufgaben.....	24
5.1 Lösung zu Aufgabe 1.....	24
5.2 Lösung zu Aufgabe 2 (a).....	25
5.3 Lösung zu Aufgabe 2 (b).....	26
5.4 Lösung zu Aufgabe 2 (c).....	27
5.5 Lösung zu Aufgabe 3 (a).....	28
5.6 Lösung zu Aufgabe 3 (b).....	29
5.7 Lösung zu Aufgabe 3 (c).....	30
5.8 Lösung zu Aufgabe 3 (d).....	32
5.9 Lösung zu Aufgabe 4 (a).....	33
5.10 Lösung zu Aufgabe 4 (b).....	34
5.11 Lösung zu Aufgabe 4 (c).....	35
5.12 Lösung zu Aufgabe 5.....	36
5.13 Lösung zu Aufgabe 6 (a).....	38
5.14 Lösung zu Aufgabe 6 (b).....	40
5.15 Lösung zu Aufgabe 7.....	42

5.16 Lösung zu Aufgabe 8 (a).....	44
5.17 Lösung zu Aufgabe 8 (b).....	46
5.18 Lösung zu Aufgabe 8 (c).....	48
5.19 Lösung zu Aufgabe 8 (d).....	50
5.20 Lösung zu Aufgabe 8 (e).....	52
5.21 Lösung zu Aufgabe 9 (a).....	54
5.22 Lösung zu Aufgabe 9 (b).....	56
5.23 Lösung zu Aufgabe 9 (c).....	58
5.24 Lösung zu Aufgabe 9 (d).....	60
5.25 Lösung zu Aufgabe 9 (e).....	62
5.26 Lösung zu Aufgabe 10 (a).....	65
5.27 Lösung zu Aufgabe 10 (b).....	68
5.28 Lösung zu Aufgabe 11 (a).....	69
5.29 Lösung zu Aufgabe 11 (b).....	71
5.30 Lösung zu Aufgabe 11 (c).....	73
5.31 Lösung zu Aufgabe 12.....	75
5.32 Lösung zu Aufgabe 13.....	77
5.33 Lösung zu Aufgabe 14.....	80

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.

2 Was ist eine Feldvariable?

Unter einer Feldvariable (in der Fachsprache Array genannt) kann man sich einen Schrank mit Schubladen vorstellen. Jede Schublade kann einen Wert aufnehmen. Die Schubladen sind durchnummeriert.



Die Nummerierung der einzelnen Felder beginnt bei 0!

Beispiel: Sei *feld* eine Feldvariable mit 4 Feldern. Das erste Feld hat die Nummer 0, das letzte Feld hat die Nummer 3.



2.1 Feldvariablen deklarieren

Feldvariable können Attribute, Parameter oder Variable einer Methode sein. Jeder Feldvariable ist ein Typ (Klasse) zugeordnet, er gibt an von welchem Type (Klasse) die Objekte in dem Feld sein dürfen.

Feldvariablen werden genauso deklariert, wie jede andere Variable auch. Es wird lediglich durch "[]" angegeben, dass es sich um eine Feldvariable handelt.



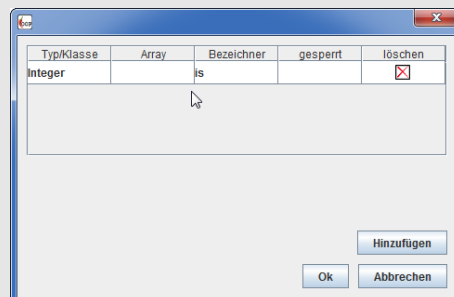
Deklaration einer Feldvariablen:

```
int [] is;
```

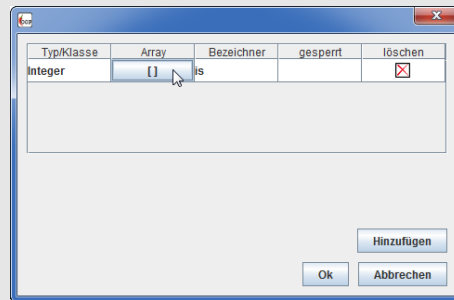
Das Feld hat den Namen *is* und kann Objekte der Klasse *Integer* aufnehmen.

Deklaration einer Feldvariable in HOOPLU:

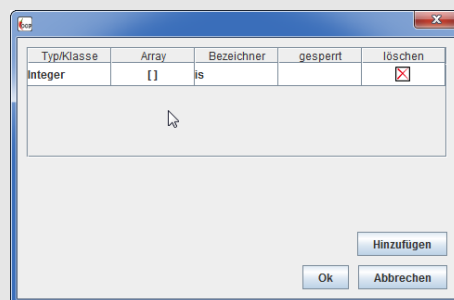
Zunächst wird ein Attribut, Parameter oder Variable vom entsprechenden Typ deklariert.



Anschließend ist der Mauszeiger in die Deklaration in der Spalte **Array** zu positionieren. Ein Button mit der Aufschrift **[]** erscheint.



Mit einem Mausklick auf den Button ist die Deklaration einer Feldvariable abgeschlossen:



2.2 Anlegen einer Feldvariable

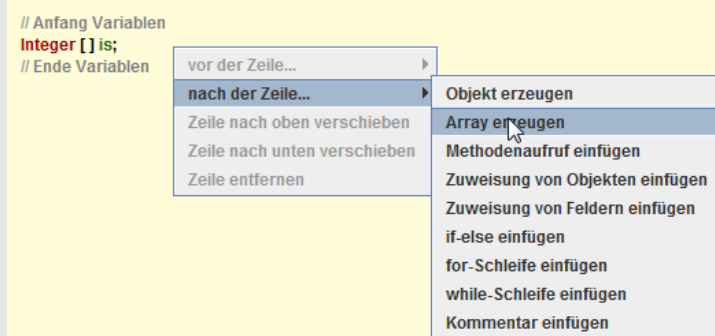
Um eine Feldvariable verwenden zu können muss diese zunächst angelegt werden. Da Felder wie Objekte behandelt werden, müssen Felder vor ihrer Verwendung mit *new* erzeugt werden.



Anlegen einer Feldvariable:

Den Mauszeiger an die Stelle im Code positionieren, an der die Feldvariable angelegt werden soll. Mit der rechten Maustaste öffnet sich ein Kontextmenü zum Bearbeiten des Codes. Den Eintrag

vor der Zeile → Array erzeugen, bzw.
nach der Zeile → Array erzeugen auswählen



```
// Anfang Variablen
Integer [] is;
// Ende Variablen

is = new Integer [1];
```

In der neuen Codezeile können nun sowohl die Feldvariable ausgewählt werden, für die ein Feld angelegt werden soll, als auch die Größe des Feldes bestimmt werden. Dazu wird der Mauszeiger auf die entsprechenden Stellen in der Codezeile positioniert. Es erscheinen entsprechende Eingabefelder, in denen die Änderungen vorgenommen werden können.

Auswahl des Feldes ändern:

```
// Anfang Variablen
Integer [] is;
// Ende Variablen

is = new Integer [1];
```

Feldgröße festlegen (hier 5 Elemente):

```
// Anfang Variablen
Integer [] is;
// Ende Variablen

is = new Integer [5];
```

2.3 Schubladen öffnen

Um auf die einzelnen Felder einer Feldvariable zugreifen zu können wird eine spezielle Syntax benötigt: `is[<index>]`



Sobald sie eine Feldvariable in Ihrem Code verwenden (z.B. für einen Methodenaufruf), so wird hinter die Variable automatisch ein eckiges Klammerpaar angezeigt. Wird der Mauszeiger innerhalb des eckigen Klammerpaares positioniert, so kann der Index bearbeitet werden.

Beispiel:

```
Integer idx;
String [] namen;

idx          = new Integer(1);
namen        = new String [3];

// Schreibender Zugriff
namen[0]     = "Paul"; // Adressierung mit einer Zahl
namen[idx]   = "Ines"; // Adressierung mit einer Variable

if ( namen[2].equal("Gert") ) // Lesender Zugriff
{
    ...
}
```

3 Mäuse fangen

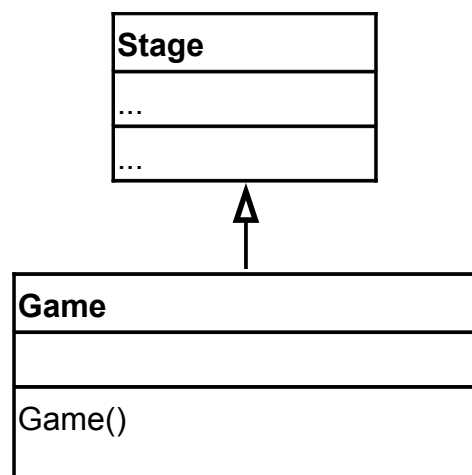
Im folgenden soll an einem Computerspiel der Einsatz von Feldvariablen gezeigt werden.



Eine Katze steht mit einem Kescher vor einem Feld von Mäuselöchern. Immer wenn eine Maus aus einem Mausloch schaut muss die Katze versuchen die Maus mit dem Kescher zu fangen. Dazu ist jedem Mausloch eine Taste auf der Tastatur zugeordnet.

3.1 Das Spielfeld

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



Als Hintergrund der *Stage* soll das Bild *Hintergrund_gelb.png* geladen werden. Es hat eine Größe von 661×472 Pixel.

**Aufgabe 1:**

Erstellen Sie ein Projekt *MaeuseFangen.oop*.

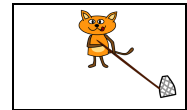
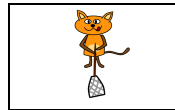
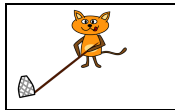
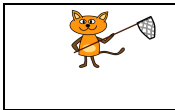
Implementieren Sie die Klasse *Game* entsprechend dem UML-Diagramm.

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

[Hier geht es zur Lösung auf Seite 24.](#)

3.2 Die Katze betritt die Bühne

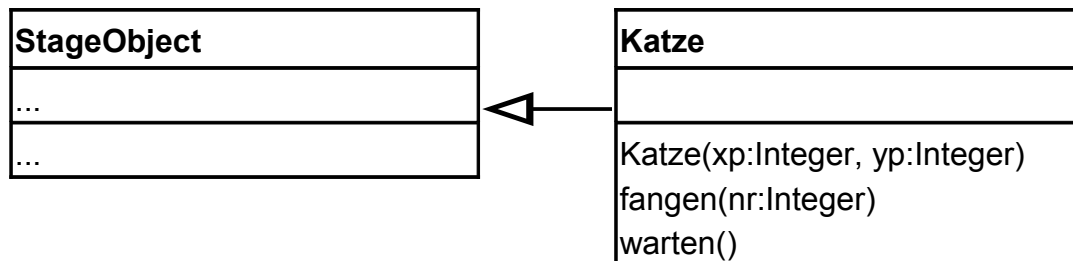
Die Katze wird als *StageObject* die Bühne (*Stage*) betreten. Für jede Haltung der Katze (Auf Mäuse warten, Maus 1 fangen, Maus 2 fangen, etc.) gibt es eine eigene Grafik.



Katze01.00.png Katze01.01.png Katze01.02.png Katze01.03.png Katze01.04.png Katze01.05.png

Diese Grafiken werden als Sprites in das *StageObject* Katze eingebunden und je nach Aktion der Katze angezeigt.

UML-Diagramm der Klasse Katze



Beschreibung der Klasse Katze

Katze(xp:Integer, yp:Integer)	
xp	Horizontale Position der Katze auf der "Bühne".
yp	Vertikale Position der Katze auf der "Bühne".
<p>Das <i>StageObject</i> wird an der Position (xp,yp) auf der Bühne platziert. Alle Grafiken werden auf 100 % skaliert. Es werden die 6 Grafiken aus der oben angegebenen Tabelle in genau der angegebenen Reihenfolge als Sprite dem <i>StageObject</i> hinzugefügt.</p>	

fangen(nr:Integer)	
nr	Nummer der Maus, die gefangen werden soll ($1 \leq nr \leq 5$)
Es wird zu der übergebenen Nummer die entsprechende Grafik in dem <i>StageObject</i> angezeigt.	

warten()
In dem <i>StageObject</i> wird die Grafik mit der Katze in Ruheposition angezeigt.

**Aufgabe 2:**

Legen Sie eine neue Klasse *Katze* entsprechend dem oben angegebenen UML-Diagramm an.

[Hier geht es zur Lösung auf Seite 25.](#)

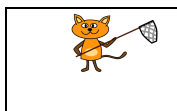
Implementieren Sie den Konstruktor gemäß obiger Beschreibung.

[Hier geht es zur Lösung auf Seite 26.](#)

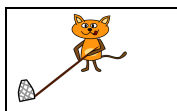
Implementieren Sie die Methoden gemäß obiger Beschreibung.

[Hier geht es zur Lösung auf Seite 27.](#)

Wie die Katze aussehen soll ist damit bestimmt. Jetzt muss sie noch zum "Leben" erweckt werden. Das "Leben" der Katze ist wie folgt festgelegt. Normal befindet sich die Katze in Ruheposition. Wird jedoch eine bestimmte Taste gedrückt, so soll sie die entsprechende Maus fangen.



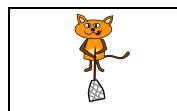
Ruheposition



Maus 1 fangen



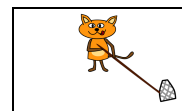
Maus 2 fangen



Maus 3 fangen



Maus 4 fangen



Maus 5 fangen

Taste <y>

Taste <x>

Taste <c>

Taste <v>

Taste

**Aufgabe 3:**

Erweitern Sie Die Klasse *Game* um ein Attribut *katze* der Klasse *Katze*.

Erzeugen Sie im Konstruktor von *Game* ein Objekt für das Attribut *katze* und platzieren sie auf der *Stage* an der Position (330,236).

[Hier geht es zur Lösung auf Seite 28.](#)

Erweitern Sie die Klasse *Game* um eine Methode

```
void mausFangen(Integer nr)
```

In dieser Methode soll die Methode *fangen* des Objektes *katze* mit dem Parameter *nr* aufgerufen werden.

[Hier geht es zur Lösung auf Seite 29.](#)

Überschreiben Sie die Methode *run*. In der *run* Methode soll eine Bedingte Anweisung eingefügt werden: Wird eine der oben angegebenen Tasten gedrückt, so soll die Methode *mausFangen* aufgerufen werden. Als Parameter wird die Nummer der Maus übergeben (Beispiel: Taste <x> wird gedrückt → *this.mausFangen(2)* wird aufgerufen). Wird keine Taste gedrückt, so wird *katze.warten()* aufgerufen.

[Hier geht es zur Lösung auf Seite 30.](#)

Fügen Sie in die *main* Methode der Klasse *Program* zwei Variablen ein. Eine vom Typ *Game* und die andere vom Typ *Computer*.

Erzeugen Sie ein neues *Game* Objekt und fügen Sie den Methodenaufruf *startSingle(200)* ein. Danach muss ein Aufruf von *waitForExitSignal()* des Objektes *Computer* eingefügt werden, um zu verhindern, dass das Programm sofort beendet wird.

Testen Sie ihr Programm.

[Hier geht es zur Lösung auf Seite 32.](#)

3.3 Die Katze wird hungrig

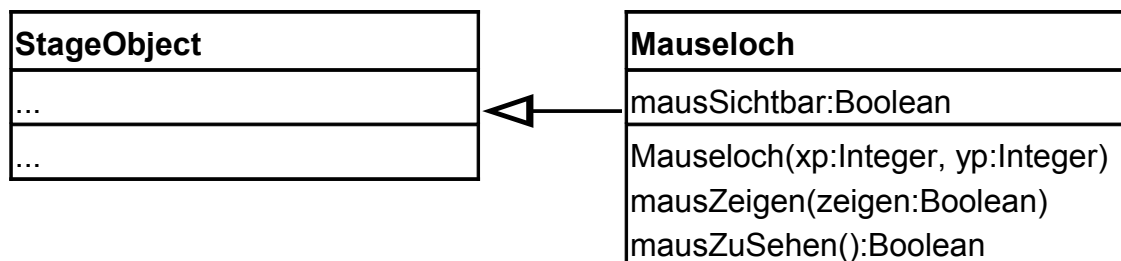


Die Katze greift immer ins Leer, egal welche Taste gedrückt wird. Es müssen Mäuse ins Spiel kommen.

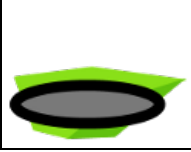

Fünf Mäuselöcher soll die Bühne bekommen, aus denen nach einem Zufallsprinzip eine Maus herauschaut.

Das Mauseloch inklusive Maus wird durch eine eigene Klasse repräsentiert

UML-Diagramm der Klasse Mauseloch



Beschreibung der Klasse Mausloch

Mausloch(xp:Integer, yp:Integer)	
xp	Horizontale Position des Mauslochs auf der "Bühne".
yp	Vertikale Position des Mauslochs auf der "Bühne".
<p>Das <i>StageObject</i> wird an der Position (xp,yp) auf der Bühne platziert. Alle Grafiken werden auf 100 % skaliert. Es werden die folgende Grafiken in der angegebenen Reihenfolge als Sprite dem <i>StageObject</i> hinzugefügt:</p> <div style="display: flex; align-items: center;">  <div style="margin-left: 10px;">Mausloch.00.png</div> </div> <div style="display: flex; align-items: center;">  <div style="margin-left: 10px;">Mausloch.01.png</div> </div> <p>Das Attribut <i>mausSichtbar</i> wird auf FALSE gesetzt.</p>	

mausZeigen(zeigen:Boolean)	
zeigen	=TRUE bedeutet, dass die Maus aus dem Loch herauschaut. =FALSE bedeutet, es ist keine Maus zu sehen.
<p>Das Attribut <i>mausSichtbar</i> wird auf den Wert von <i>zeigen</i> gesetzt. Ist <i>zeigen</i>=TRUE, dann wird die Grafik <i>Mausloch.01.png</i> angezeigt, im anderen Fall wird <i>Mausloch.00.png</i> angezeigt.</p>	

mausZuSehen():Boolean	
Liefert den Wert des Attributs <i>mausSichtbar</i> zurück.	

**Aufgabe 4:**

Legen Sie eine neue Klasse *Mausloch* entsprechend dem oben angegebenen UML-Diagramm an.

[Hier geht es zur Lösung auf Seite 33.](#)

Implementieren Sie den Konstruktor gemäß obiger Beschreibung.

[Hier geht es zur Lösung auf Seite 34.](#)

Implementieren Sie die Methoden gemäß obiger Beschreibung.

[Hier geht es zur Lösung auf Seite 35.](#)

Im folgenden werden fünf Mäuselöcher auf der Bühne platziert. Grundsätzlich ist es nötig, für jedes Mauseloch ein Attribut in der Klasse *Game* anzulegen. Um den damit Verbundenen Aufwand zu verringern und die weitere Programmlogik zu vereinfachen soll statt mehreren Attributen eine Feldvariable zur Speicherung der Mäuselöcher verwendet werden.

**Aufgabe 5:**

Fügen Sie in der Klasse *Game* eine Feldvariable für Objekte des Typs *Mauseloch* als Attribut mit dem Namen *m/s* ein ([Wie Feldvariablen deklariert werden ist auf Seite 2 beschrieben](#)).

[Hier geht es zur Lösung auf Seite 36.](#)

Im nächsten Schritt muss die Feldvariable angelegt, die Mauselöcher erzeugt und auf der Bühne platziert werden.

**Aufgabe 6:**

Hinweis: Fügen Sie Ihren Code vor dem Statement ein, mit dem das Objekt der Klasse *Katze* erzeugt wird:

```
// --> Code an dieser Stelle einfüegen  
katze = new Katze(330,236);
```

Legen Sie im Konstruktor eine Feldvariable für das Attribut *m/s* an ([Wie eine Feldvariable angelegt wird ist auf Seite 3 beschrieben](#)). Die Länge des Feldes soll 5 sein.

Erzeugen Sie ein Objekt der Klasse *Mauseloch* und speichern Sie es im 1. Feld von *m/s*.

Platzieren Sie das Mauseloch auf der Bühne an der Position (100,380).

[Hier geht es zur Lösung auf Seite 38.](#)

Erzeugen Sie auf die gleiche Weise die restlichen vier Mauselöcher und speichern sie im Feld *m/s*. Platzieren Sie die Mauselöcher an den folgenden Positionen auf der Bühne: (218,380), (336,380), (454,380), (572,380)

[Hier geht es zur Lösung auf Seite 40.](#)

Wird das Programm jetzt gestartet, sind zwar Mauselöcher zu sehen, aber es kommen keine Mäuse zum Vorschein.

Nach dem Zufallsprinzip soll ein Mauseloch ausgewählt werden, aus dem eine Maus hervorschaut. Dazu wird ein weiteres Attribut benötigt, in dem die Nummer des Mauselochs gespeichert wird, aus dem die Maus heraus schaut.

Außerdem soll ein "Zeitzähler" darüber entscheiden, wie lange eine Maus aus ihrem Loch herausschaut. Der Zeitzähler muss ebenfalls als Attribut angelegt werden.



Aufgabe 7:

Legen Sie in der Klasse *Game* zwei Attribute *maus* und *zeit* vom Typ *Integer* an.

Im Konstruktor soll *maus* der Wert 0 und *zeit* der Wert 15 zugewiesen werden.

[Hier geht es zur Lösung auf Seite 42.](#)

In der *run* Methode der Klasse *Game* muss nun entschieden werden, welche Maus sich aus ihrem Mauseloch wagt. Allerdings zeigt sich eine Maus erst, wenn keine andere Maus mehr zu sehen ist. Das bedeutet, dass erst eine neue Maus ausgewählt wird, wenn der Zeitzähler auf 0 ist. Es muss somit auch der Zeitzähler herunter gezählt werden.



Aufgabe 8:

Fügen Sie am Ende der *run* Methode der Klasse *Game* Code ein, der den Zähler im Attribut *zeit* um 1 verringert.

[Hier geht es zur Lösung auf Seite 44.](#)

Ist der Zähler im Attribut *zeit* ≤ 0 , dann soll die Maus im Mauseloch mit der Nummer *maus* wieder verschwinden. Verwenden Sie dazu die Methode *mausZeigen(FALSE)* der Klasse *Mauseloch*.

[Hier geht es zur Lösung auf Seite 46.](#)

Weisen Sie dem Attribut *maus* eine neue Zufallszahl zwischen 0 und 4 zu.

[Hier geht es zur Lösung auf Seite 48.](#)

Lassen Sie im Mauseloch mit der Nummer *maus* eine Maus herausschauen. Verwenden Sie dazu die Methode *mausZeigen(TRUE)* der Klasse *Mauseloch*.

[Hier geht es zur Lösung auf Seite 50.](#)

Setzen Sie den Zeitzähler im Attribut *zeit* auf den Wert 5.

[Hier geht es zur Lösung auf Seite 52.](#)

Testen Sie Ihr Programm.

3.4 Der Erfolg der Katze

Die Katze kann versuchen die Mäuse zu fangen. Allerdings ist nicht zu sehen, ob sie dabei erfolgreich ist oder am Ende verhungert.

Es müssen Zähler her, die zum einen festhalten, wie viel Mäuse sich zeigen und zum anderen zeigen wie viele von der Katze gefangen werden.



Aufgabe 9:

Fügen Sie in die Klasse *Game* zwei Attribute *anzahlMaeuse* und *anzahlGefangen* ein, die als Zähler dienen. Im Konstruktor sollen beide Werte mit 0 initialisiert werden.

[Hier geht es zur Lösung auf Seite 54.](#)

Fügen Sie in die Klasse *Game* ein Attribut *punkteAnzeige* vom Typ *StageObject* ein. Erzeugen Sie im Konstruktor ein Objekt für das Attribut und platzieren Sie es an der Position (330,25) (Skalierung 100 %). Der Text soll zentriert angezeigt werden. Die *punkteAnzeige* soll noch vor allen anderen *StageObjects* die *Stage* betreten.

[Hier geht es zur Lösung auf Seite 56.](#)

Erstellen Sie in der Klasse *Game* eine Methode
void refreshAnzeige()

Die Methode soll den String "x von y Mäusen gefangen" in dem *StageObject* *punkteAnzeige* mit einer Schriftgröße von 30 ausgeben. Dabei ist x der aktuelle Wert des Attributs *anzahlGefangen* und y der aktuelle Wert des Attributs *anzahlMaeuse*.

[Hier geht es zur Lösung auf Seite 58.](#)

Zeigt sich eine Maus in ihrem Mauselloch, so soll der Zähler im Attribut *anzahlMaeuse* um 1 erhöht werden und die Anzeige aktualisiert werden (verwenden Sie dazu die zuvor erstellte Methode *refreshAnzeige()*).

[Hier geht es zur Lösung auf Seite 60.](#)

Die Methode *mausFangen* der Klasse *Game* muss um eine bedingte Anweisung erweitert werden. Als Bedingung wird geprüft, ob aus dem Mauselloch, auf das die Katze ihren Kescher richtet, eine Maus herauschaut. Der Parameter *nr* gibt die Nummer des Mausellochs an, auf das die Katze ihren Kescher richtet. Ob eine Maus aus ihrem Mauselloch schaut, kann mit der Methode *mausZuSehen()* der Klasse *Mauselloch* geprüft werden.

Ist die Bedingung erfüllt, dann hat die Katze eine Maus gefangen und der Zähler im Attribut *anzahlGefangen* muss um 1 erhöht und die *punkteAnzeige* aktualisiert werden.

[Hier geht es zur Lösung auf Seite 62.](#)

3.5 Das Spiel beenden

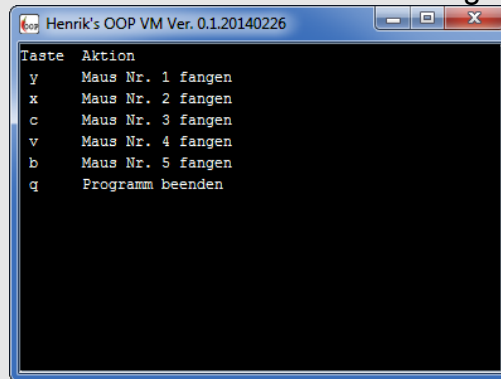


Aufgabe 10:

Erweitern Sie die Klasse *Game* so, dass bei einem Tastendruck von <q> das Spiel beendet wird.

[Hier geht es zur Lösung auf Seite 65.](#)

Geben Sie auf der Konsole eine kurze Bedienungsanleitung aus:



```
Henrik's OOP VM Ver. 0.1.20140226
Taste  Aktion
y      Maus Nr. 1 fangen
x      Maus Nr. 2 fangen
c      Maus Nr. 3 fangen
v      Maus Nr. 4 fangen
b      Maus Nr. 5 fangen
q      Programm beenden
```

Der Code dazu soll in der Methode *main* der Klasse *Program* eingefügt werden.

[Hier geht es zur Lösung auf Seite 68.](#)

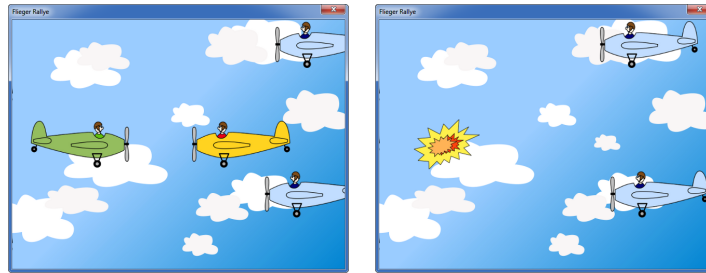
Das Spiel ist nun fertig. Lehnen Sie sich zurück, spielen Sie ein paar Runden um sich dann an die Übungen auf den folgenden Seiten zu machen.

4 Übungen

4.1 Flieger Rallye

In diesem Spiel fliegt der Spieler im Gegenverkehr und muss den Entgegenkommenden Flugzeugen ausweichen. Ausweichen kann der Spieler, in dem er sein Flugzeug nach oben oder unten steuert.

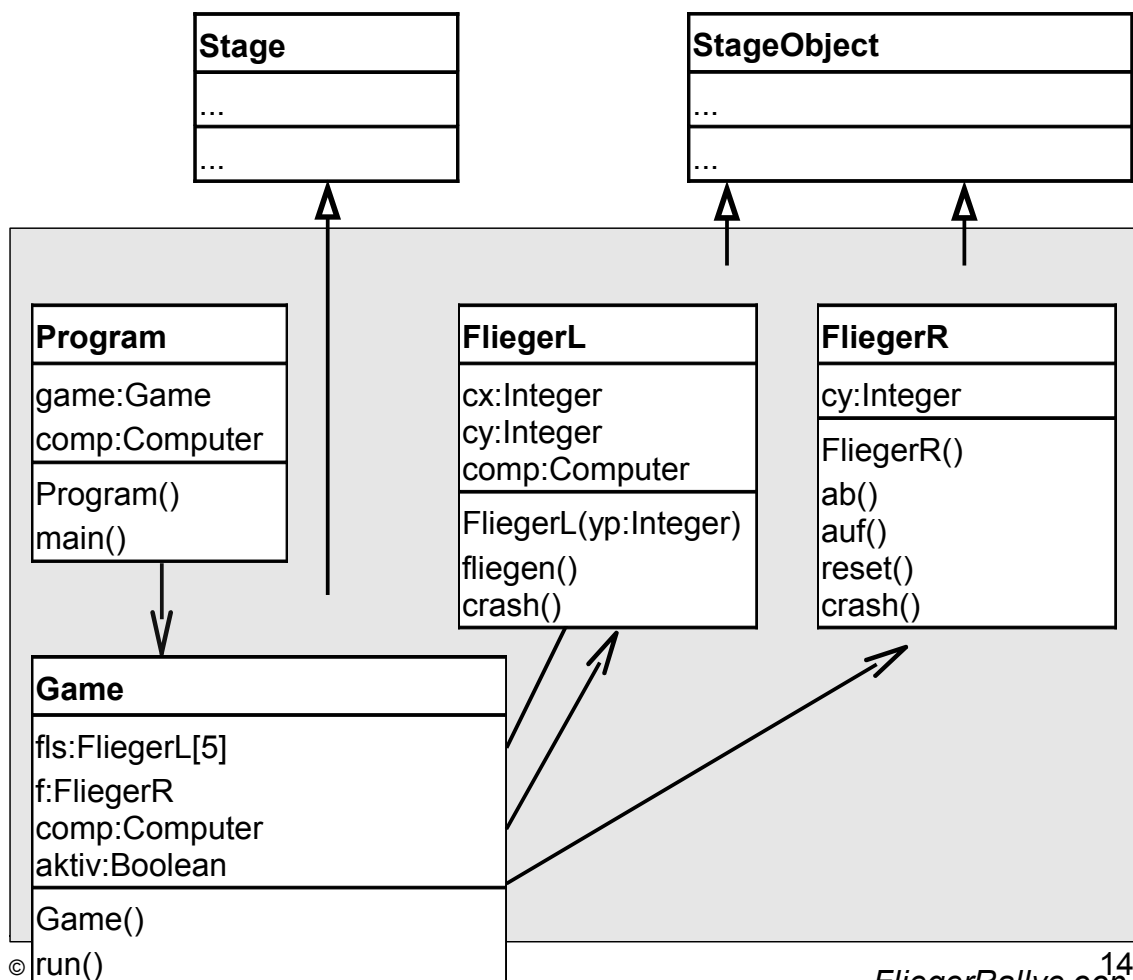
Kommt es zur Kollision, so gehen die kollidierenden Flugzeuge in einem Feuerball auf.



Die Datei *FliegerRallye.demo.oop* enthält das Programm zum Testen.

Entwickeln Sie ein Programm, dass oben beschriebenes Spiel realisiert. Berücksichtigen Sie dabei folgendes UML-Diagramm und die Klassenbeschreibungen.

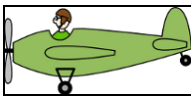
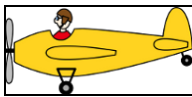
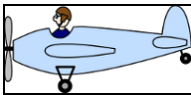
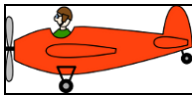
UML-Diagramm



Beschreibung der Klasse FliegerL

FliegerL beschreibt Flugzeuge die von rechts nach links fliegen.

Attribute	
cx	Aktuelle horizontale Position des Flugzeugs.
cy	Aktuelle vertikale Position des Flugzeugs.
comp	Wird für das Generieren von Zufallszahlen benötigt.

FliegerL(yp:Integer)	
yp	Vertikale Position des Flugzeugs.
<p>Zunächst wird das <i>StageObject</i> an der Position (0,0) auf der Bühne platziert. Alle Grafiken werden auf 100 % skaliert. Es werden die folgende Grafiken in der angegebenen Reihenfolge als Sprite dem <i>StageObject</i> hinzugefügt:</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;">  Flieger01.l.png </div> <div style="text-align: center;">  Flieger02.l.png </div> </div> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;">  Flieger03.l.png </div> <div style="text-align: center;">  Flieger04.l.png </div> </div> <p>Das Attribut <i>cx</i> bekommt den Wert -100. Das Attribut <i>cy</i> wird auf den Parameter <i>yp</i> gesetzt. Anschließend wird das <i>StageObject</i> an die Position (<i>cx</i>,<i>cy</i>) verschoben. Als letztes wird für das Attribut <i>comp</i> ein Objekt erzeugt.</p>	

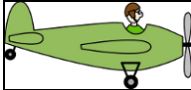

fliegen()
<p>Von dem Wert im Attribut <i>cx</i> werden 20 abgezogen. Wenn der Wert in <i>cx</i> kleiner als 0 ist, dann</p> <ul style="list-style-type: none"> • erzeuge eine Zufallszahl im Bereich von 0 bis 9. • Ist die Zufallszahl größer als 8, dann <ul style="list-style-type: none"> • bekommt das Attribut <i>cx</i> den Wert 700. • Es wird eine Zufallszahl im Bereich von 0 bis 3 erzeugt. • In Abhängigkeit von der zuvor erzeugten Zufallszahl wird das anzuzeigende Flugzeug ausgewählt (0≐grün, 1≐gelb, 2≐blau und 3≐rot). <p>Das <i>StageObject</i> wird an die Position (<i>cx</i>,<i>cy</i>) verschoben.</p>

crash()
<p>Das Attribut <i>cx</i> bekommt den Wert -100. Anschließend wird das <i>StageObject</i> an die Position (<i>cx</i>,<i>cy</i>) verschoben.</p>

Beschreibung der Klasse FliegerR

FliegerR beschreibt Flugzeuge die von links nach rechts fliegen.

Attribute	
cy	Aktuelle vertikale Position des Flugzeugs.

FliegerR()
<p>Zunächst wird das <i>StageObject</i> an der Position (130,0) auf der Bühne platziert. Alle Grafiken werden auf 100 % skaliert. Es werden die folgende Grafiken in der angegebenen Reihenfolge als Sprite dem <i>StageObject</i> hinzugefügt:</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  <p>Flieger01.r.png</p> </div> <div style="text-align: center;">  <p>Explosion01.png</p> </div> </div> <p>Die <i>reset</i> Methode der Klasse <i>FliegerR</i> wird aufgerufen um das Flugzeug ist Startposition zu bringen.</p>

ab()
<p>Wenn der Wert im Attribut <i>cy</i> kleiner als 432 ist, dann</p> <ul style="list-style-type: none"> • addiere zu dem Wert in <i>cy</i> den Wert 24 dazu • und verschiebe <i>StageObject</i> an die Position (130,<i>cy</i>).

auf()
<p>Wenn der Wert im Attribut <i>cy</i> größer als 48 ist, dann</p> <ul style="list-style-type: none"> • subtrahiere von dem Wert in <i>cy</i> den Wert 24 • und verschiebe <i>StageObject</i> an die Position (130,<i>cy</i>).

reset()
<p>Es wird das Bild mit dem Flugzeug angezeigt. Das Attribut <i>cy</i> bekommt den Wert 240. Das <i>StageObject</i> wird an die Position (130,<i>cy</i>) verschoben.</p>

crash()
<p>Es wird das Bild mit dem Feuerball angezeigt.</p>

Beschreibung der Klasse Game

<i>Attribute</i>	
<i>fls</i>	Verwaltet 5 Objekte mit Flugzeugen, die von rechts nach links fliegen.
<i>f</i>	Verwaltet das Flugzeug des Spielers.
<i>comp</i>	Wird für das Generieren von Zufallszahlen und zur Steuerung des Programms benötigt.
<i>aktiv</i>	Gibt an, ob das Spiel gerade aktiv ist.

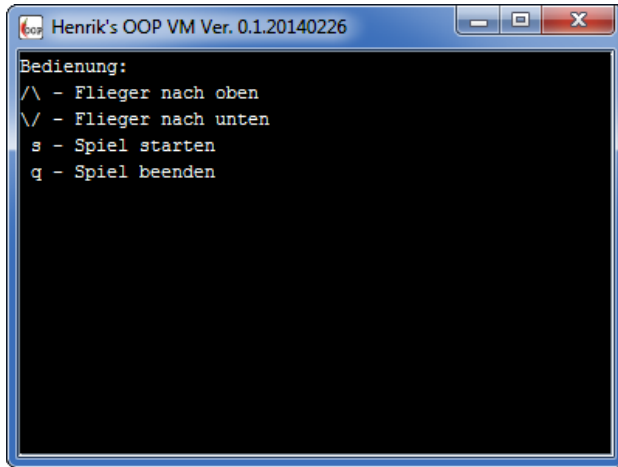
<i>Game()</i>
<p>Das Spiel soll "Flieger Rallye" heißen. Das Spielfenster hat eine Größe von 640×480 Pixel. Das Hintergrundbild befindet sich in der Datei <i>Himmel01.png</i>. Im Attribut <i>fls</i> soll ein Feld für 5 Einträge angelegt werden. Für jeden Eintrag im Feld <i>fls</i> ist ein Objekt vom Type <i>FliegerL</i> zu erzeugen. Die vertikalen Positionen der einzelnen Objekte sind wie folgt: 48, 144, 240, 336 und 432. Alle Objekte im Feld <i>fls</i> werden auf der <i>Stage</i> platziert.</p> <p>Das Attribut <i>f</i> erhält ein neues Objekt des Typs <i>FliegerR</i>, das ebenfalls auf der <i>Stage</i> platziert wird.</p> <p>Abschließend wird dem Attribut <i>comp</i> ein neues Objekt zugewiesen und erhält das Attribut <i>aktiv</i> den Wert FALSE.</p>

<i>run()</i>
<p>Wenn das Attribut <i>aktiv</i> =TRUE ist, dann</p> <ul style="list-style-type: none"> • erzeuge eine Zufallszahl <i>fliegerNr</i> im Bereich von 0 bis 4. • Das Flugzeug im Feld <i>fls</i> an der Position <i>fliegerNr</i> fliegt ein Stück (verwenden Sie die Methode <i>fliegen</i> der Klasse <i>FliegerL</i>). • Kollidiert danach das Flugzeug mit dem des Spielers, dann <ul style="list-style-type: none"> • wird von beiden Flugzeugen die <i>crash</i> Methode aufgerufen • und das Attribut <i>aktiv</i> auf FALSE gesetzt. • sonst prüfe, ob die Taste <↑> gedrückt wurde. Wenn ja, dann <ul style="list-style-type: none"> • lasse das Flugzeug des Spielers aufsteigen. • sonst prüfe, ob die Taste <↓> gedrückt wurde. Wenn ja, dann <ul style="list-style-type: none"> • lasse das Flugzeug des Spielers absteigen. <p>sonst prüfe, ob die Taste <s> gedrückt wurde. Wenn ja, dann</p> <ul style="list-style-type: none"> • resette das Flugzeug des Spielers. • Setze das Attribut <i>aktiv</i> auf TRUE. <p>Prüfe, ob , ob die Taste <q> gedrückt wurde. Wenn ja, dann</p> <ul style="list-style-type: none"> • setze das Attribut <i>aktiv</i> auf FALSE. • Rufe die <i>sendExitSignal</i> Methode der Klasse <i>Computer</i> auf.

Beschreibung der Klasse Program

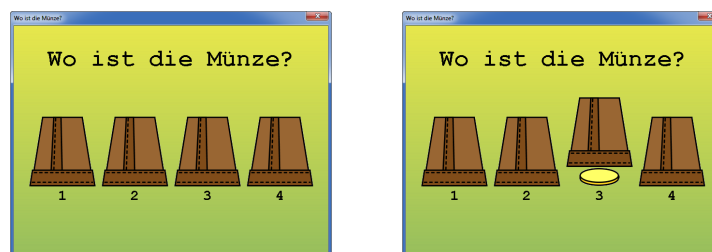
<i>Attribute</i>	
game	Repräsentiert das Spiel.
comp	Wird für die Textausgabe und zur Steuerung des Programms benötigt.

Program()
-

main()
<p>Dem Attribut <i>game</i> wird ein neues Objekt vom Type <i>Game</i> zugewiesen. Das Spiel im Attribut <i>game</i> wird mit einem Aufruf der Methode <i>startSingle(50)</i> gestartet.</p> <p>Auf der Konsole wird folgende Bedienungsanleitung ausgegeben:</p>  <p>Abschließend erfolgt ein Aufruf der Methode <i>waitForExitSignal</i> der Klasse <i>Computer</i>.</p>

4.2 Wo ist die Münze?

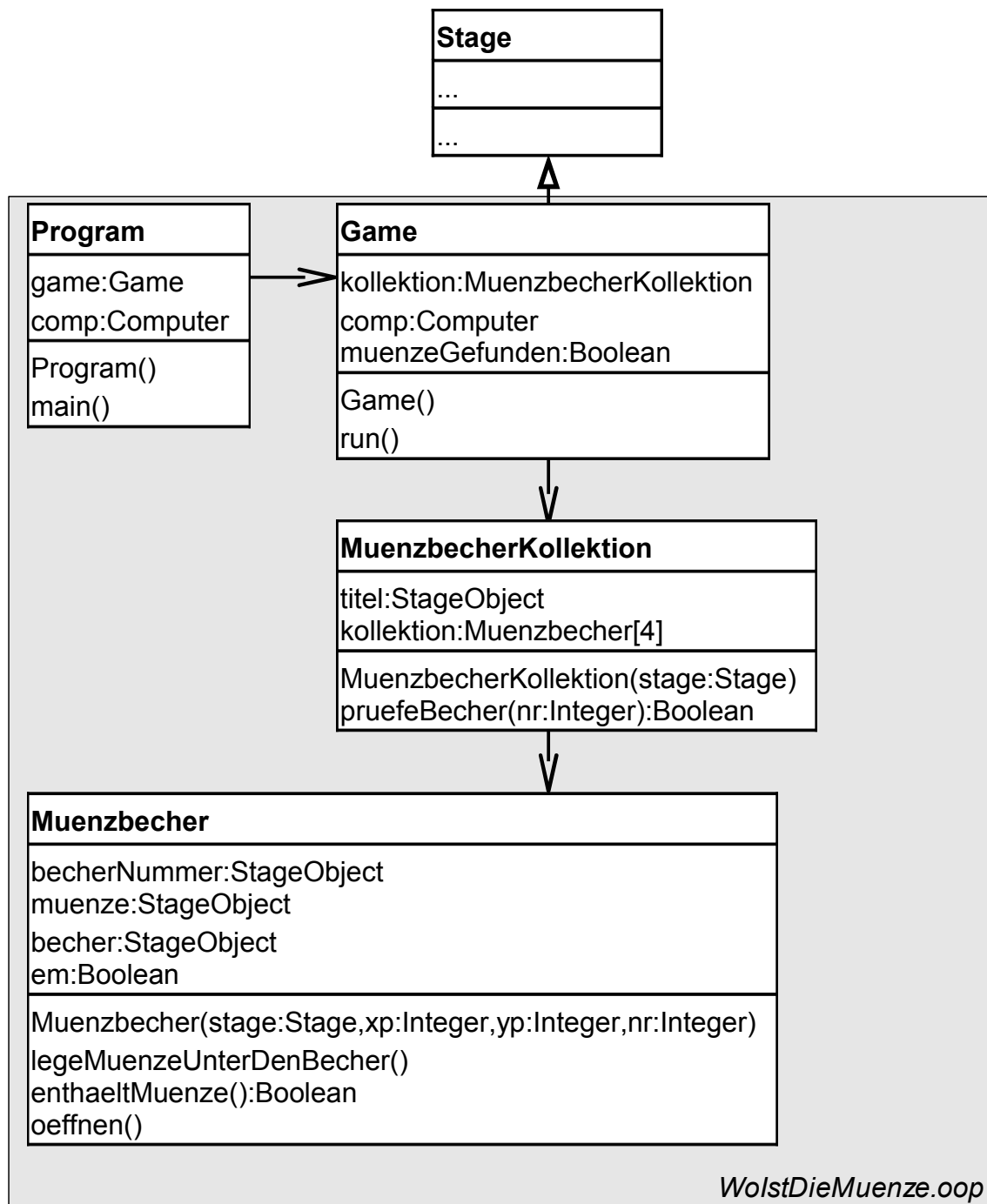
In diesem Spiel soll der Spieler Raten unter welchen der 4 Becher sich die Münze befindet. Der Spieler darf so lange aufdecken, bis er die Münze gefunden hat.



Die Datei *WolstDieMuenze.demo.oop* enthält das Programm zum Testen.

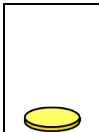
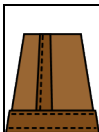
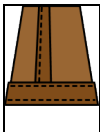
Entwickeln Sie ein Programm, dass oben beschriebenes Spiel realisiert. Berücksichtigen Sie dabei folgendes UML-Diagramm und die Klassenbeschreibungen.

UML-Diagramm



Beschreibung der Klasse Muenzbecher

<i>Attribute</i>	
becherNummer	Label zur Darstellung der Becher Nummerierung.
muenze	Grafisches Element zur Darstellung der Münze.
becher	Grafisches Element zur Darstellung des Bechers.
em	Wen die Münze unter dem Becher liegt ist <i>em</i> =TRUE, sonst ist <i>em</i> =FALSE.

Muenzbecher(stage:Stage,xp:Integer,yp:Integer,nr:Integer)	
stage	Bühne, auf der der Münzbecher platziert werden soll.
xp	Horizontale Position des Münzbechers auf der Bühne.
yp	Vertikale Position des Münzbechers auf der Bühne.
nr	Nummer des Münzbechers.
<p>Zunächst wird ein <i>StageObject</i> für das Attribut <i>muenze</i> erzeugt und auf der Bühne <i>stage</i> platziert. Die Grafik für die Münze ist in der Datei</p> <div style="display: flex; align-items: center;">  <div style="margin-left: 10px;">Muenze.png (Skallierung 100 %).</div> </div> <p>Positionen der Münze ist (xp,yp). Außerdem soll die Münze ausgeblendet werden (<i>StageObject.hide()</i>).</p> <p>Es wird ein <i>StageObject</i> für das Attribut <i>becher</i> erzeugt und auf der Bühne an der Position (xp,yp) platziert. Dem <i>StageObject</i> werden folgende Bilder (Skallierung 100 %) als Sprite in der angegebenen Reihenfolge hinzugefügt.</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  <div>Becher.geschlossen.png</div> </div> <div style="text-align: center;">  <div>Becher.offen.png</div> </div> </div> <p>Schließlich wird noch <i>StageObject</i> für das Attribut <i>becher</i> erzeugt und auf der Bühne an der Position $(xp,yp+100)$ platziert. Der Text soll zentriert ausgerichtet werden. Außerdem soll das <i>StageObject</i> den Wert des Parameters <i>nr</i> mit einer Schriftgröße von 30 anzeigen.</p> <p>Das Attribut <i>em</i> wird FALSE gesetzt.</p>	

legeMuenzeUnterDenBecher()	
<p>Das Attribut <i>em</i> wird TRUE gesetzt.</p> <p>Die Münze wird eingeblendet (<i>StageObject.show()</i>).</p>	

enthaeltMuenze():Boolean

Gibt den Wert des Attributs <i>em</i> zurück.

oeffnen()

Im Attribut <i>becher</i> wird das Sprite <i>Becher.offen.png</i> angezeigt.
--

Beschreibung der Klasse MuenzbecherKollektion

Attribute	
-----------	--

kollektion	Feld mit Münzbechern.
------------	-----------------------

titel	Titel des Spiels.
-------	-------------------

MuenzbecherKollektion(stage:Stage)

Als erstes wird für das Attribut *titel* ein *StageObject* erzeugt und auf der *stage* an der Position (325,50) platziert. In dem *StageObject* soll der Text "Wo ist die Münze?" zentriert und mit einer Schriftgröße von 50 dargestellt werden.

Als nächstes wird für das Attribut *kollektion* ein Feld mit 4 Einträgen angelegt. Für jeden Eintrag wird ein Objekt vom Typ *Muenzbecher* erzeugt. Die Positionen der 4 Münzbecher sind (100,240), (250,240), (40,240) und (550,240).

Mit einer Zufallszahl im Bereich von 0 bis 3 wird festgelegt, unter welchem Becher die Münze liegt (*Muenzbecher.legeMuenzeUnterDenBecher()*).

pruefeBecher(nr:Integer):Boolean

Es wird der Becher <i>nr</i> geöffnet.
--

Wenn der Becher <i>nr</i> die Münze enthält, liefert die Methode TRUE zurück. Sonst ist der Rückgabewert FALSE.

Beschreibung der Klasse Game

Attribute	
-----------	--

kollektion	Münzbecher Kollektion von 4 Münzbechern für das Spiel.
------------	--

muenzeGefunden	Ist =TRUE, wenn die Münze gefunden wurde. Sonst ist das Attribut =FALSE.
----------------	--

comp	Wird für die Textausgabe auf der Konsole und zur Steuerung des Programms benötigt.
------	--

Game()

Das Spiel soll "Wo ist die Münze?" heißen. Das Spielfenster hat eine Größe von 650×480 Pixel. Das Hintergrundbild befindet sich in der Datei *BG_green.png*.

Das Attribut *muenzeGefunden* wird FALSE gesetzt.

run()

Prüfe, ob die Taste <a> gedrückt wird. Wenn ja,

- öffnen und prüfe den ersten Becher in der Kollektion
- Ist die Münze unter dem Becher, so wird das Attribut *muenzeGefunden* auf TRUE gesetzt.

sonst prüfe, ob die Taste <s> gedrückt wird. Wenn ja,

- öffnen und prüfe den zweite Becher in der Kollektion
- Ist die Münze unter dem Becher, so wird das Attribut *muenzeGefunden* auf TRUE gesetzt.

sonst prüfe, ob die Taste <d> gedrückt wird. Wenn ja,

- öffnen und prüfe den dritte Becher in der Kollektion
- Ist die Münze unter dem Becher, so wird das Attribut *muenzeGefunden* auf TRUE gesetzt.

sonst prüfe, ob die Taste <f> gedrückt wird. Wenn ja,

- öffnen und prüfe den vierten Becher in der Kollektion
- Ist die Münze unter dem Becher, so wird das Attribut *muenzeGefunden* auf TRUE gesetzt.

Ist das Attribut *muenzeGefunden* =TRUE, dann

- soll auf der Konsole die Nachricht "Glückwunsch, Sie haben die Münze gefunden!" ausgegeben werden.
- Um das Programm zu beenden wird die Methode *sendExitSignal* der Klasse *Computer* aufgerufen.
- Das Attribut *muenzeGefunden* wird auf FALSE gesetzt.

Beschreibung der Klasse Program**Attribute**

game	Repräsentiert das Spiel.
comp	Wird für die Textausgabe und zur Steuerung des Programms benötigt.

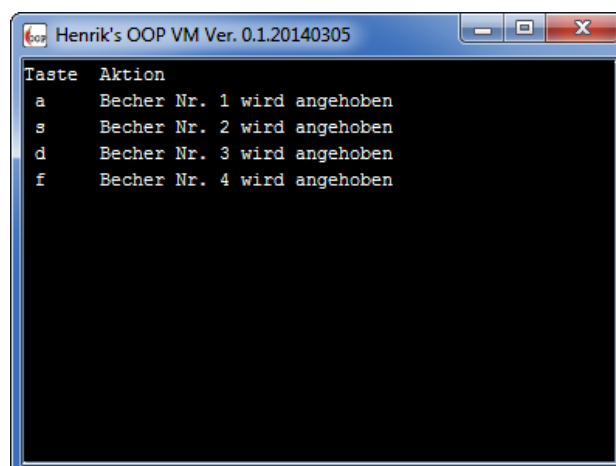
Program()

-

main()

Dem Attribut *game* wird ein neues Objekt vom Type *Game* zugewiesen.
Das Spiel im Attribut *game* wird mit einem Aufruf der Methode *startSingle(200)* gestartet.

Auf der Konsole wird folgende Bedienungsanleitung ausgegeben:



Abschließend erfolgt ein Aufruf der Methode *waitForExitSignal* der Klasse *Computer*.

ENDE

5 Lösungen zu den Aufgaben

5.1 Lösung zu Aufgabe 1

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

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

        super("Mäuse fangen", 661, 472);
        super.loadBackground("Hintergrund_gelb.png");    }
}
```

[Zurück zur Aufgabe auf Seite 6.](#)

5.2 Lösung zu Aufgabe 2 (a)

```
public class Katze extends StageObject
{
    // Anfang Attribute
    // Ende Attribute

    public Katze (Integer xp,Integer yp)
    {
        // Anfang Variablen
        // Ende Variablen

        super (NULL,NULL,NULL) ;
    }

    public void fangen (Integer nr)
    {
        // Anfang Variablen
        // Ende Variablen
    }

    public void warten ()
    {
        // Anfang Variablen
        // Ende Variablen
    }
}
```

[Zurück zur Aufgabe auf Seite 7.](#)

5.3 Lösung zu Aufgabe 2 (b)

```
public class Katze extends StageObject
{
    // Anfang Attribute
    // Ende Attribute

    public Katze (Integer xp,Integer yp)
    {
        // Anfang Variablen
        // Ende Variablen

        super(xp, yp, 100);
        super.addSprite("Ruheposition", "Katze01.00.png");
        super.addSprite("Maus 1 gefangen", "Katze01.01.png");
        super.addSprite("Maus 2 gefangen", "Katze01.02.png");
        super.addSprite("Maus 3 gefangen", "Katze01.03.png");
        super.addSprite("Maus 4 gefangen", "Katze01.04.png");
        super.addSprite("Maus 5 gefangen", "Katze01.05.png");
    }

    public void fangen (Integer nr)
    {
        // Anfang Variablen
        // Ende Variablen
    }

    public void warten ()
    {
        // Anfang Variablen
        // Ende Variablen
    }
}
```

[Zurück zur Aufgabe auf Seite 7.](#)

5.4 Lösung zu Aufgabe 2 (c)

```
public class Katze extends StageObject
{
    // Anfang Attribute
    // Ende Attribute

    public Katze (Integer xp,Integer yp)
    {
        // Anfang Variablen
        // Ende Variablen

        super(xp,yp,100);
        super.addSprite("Ruheposition","Katze01.00.png");
        super.addSprite("Maus 1 gefangen","Katze01.01.png");
        super.addSprite("Maus 2 gefangen","Katze01.02.png");
        super.addSprite("Maus 3 gefangen","Katze01.03.png");
        super.addSprite("Maus 4 gefangen","Katze01.04.png");
        super.addSprite("Maus 5 gefangen","Katze01.05.png");
    }

    public void fangen (Integer nr)
    {
        // Anfang Variablen
        // Ende Variablen

        super.setSprite(nr);
    }

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

        super.setSprite(0);
    }
}
```

[Zurück zur Aufgabe auf Seite 7.](#)

5.5 Lösung zu Aufgabe 3 (a)

```
public class Game extends Stage
{
    // Anfang Attribute
    private Katze katze;
    // Ende Attribute

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

        super("Mäuse fangen", 661, 472);
        super.loadBackground("Hintergrund_gelb.png");
        katze = new Katze(330, 236);
        super.addStageObject(katze);
    }
}
```

[Zurück zur Aufgabe auf Seite 7.](#)

5.6 Lösung zu Aufgabe 3 (b)

```
public class Game extends Stage
{
    // Anfang Attribute
    private Katze katze;
    // Ende Attribute

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

        super("Mäuse fangen", 661, 472);
        super.loadBackground("Hintergrund_gelb.png");
        katze = new Katze(330, 236);
        super.addStageObject(katze);
    }

    public void mausFangen (Integer nr)
    {
        // Anfang Variablen
        // Ende Variablen

        katze.fangen(nr);
    }
}
```

[Zurück zur Aufgabe auf Seite 7.](#)

5.7 Lösung zu Aufgabe 3 (c)

```
public class Game extends Stage
{
    // Anfang Attribute
    private Katze katze;
    // Ende Attribute

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

        super("Mäuse fangen", 661, 472);
        super.loadBackground("Hintergrund_gelb.png");
        katze = new Katze(330, 236);
        super.addStageObject(katze);
    }

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

        if (super.isKeyPressed("y"))
        {
            this.mausFangen(1);
        }
        else if (super.isKeyPressed("x"))
        {
            this.mausFangen(2);
        }
        else if (super.isKeyPressed("c"))
        {
            this.mausFangen(3);
        }
        else if (super.isKeyPressed("v"))
        {
            this.mausFangen(4);
        }
        else if (super.isKeyPressed("b"))
        {
            this.mausFangen(5);
        }
        else
        {
            katze.warten();
        }
    }
}
```

```
public void mausFangen (Integer nr)
{
    // Anfang Variablen
    // Ende Variablen

    katze.fangen(nr);
}
}
```

[Zurück zur Aufgabe auf Seite 7.](#)

5.8 Lösung zu Aufgabe 3 (d)

```
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(200);
        comp = new Computer();
        comp.waitForExitSignal();
    }
}
```

[Zurück zur Aufgabe auf Seite 7.](#)

5.9 Lösung zu Aufgabe 4 (a)

```
public class Mauseloch extends StageObject
{
    // Anfang Attribute
    private Boolean mausSichtbar;
    // Ende Attribute

    public Mauseloch (Integer xp,Integer yp)
    {
        // Anfang Variablen
        // Ende Variablen

        super (NULL,NULL,NULL) ;
    }

    public Boolean mausZuSehen ()
    {
        // Anfang Variablen
        // Ende Variablen

        return NULL;
    }

    public void mausZeigen (Boolean zeigen)
    {
        // Anfang Variablen
        // Ende Variablen
    }
}
```

[Zurück zur Aufgabe auf Seite 9.](#)

5.10 Lösung zu Aufgabe 4 (b)

```
public class Mauselloch extends StageObject
{
    // Anfang Attribute
    private Boolean mausSichtbar;
    // Ende Attribute

    public Mauselloch (Integer xp,Integer yp)
    {
        // Anfang Variablen
        // Ende Variablen

        super(xp, yp, 100);
        super.addSprite("Mauselloch", "Mauselloch.00.png");
        super.addSprite("Mauselloch mit Maus", "Mauselloch.01.png");
        mausSichtbar = new Boolean(FALSE);
    }

    public Boolean mausZuSehen ()
    {
        // Anfang Variablen
        // Ende Variablen

        return NULL;
    }

    public void mausZeigen (Boolean zeigen)
    {
        // Anfang Variablen
        // Ende Variablen
    }
}
```

[Zurück zur Aufgabe auf Seite 9.](#)

5.11 Lösung zu Aufgabe 4 (c)

```
public class Mauseloch extends StageObject
{
    // Anfang Attribute
    private Boolean mausSichtbar;
    // Ende Attribute

    public Mauseloch (Integer xp,Integer yp)
    {
        // Anfang Variablen
        // Ende Variablen

        super(xp,yp,100);
        super.addSprite("Mauseloch","Mauseloch.00.png");
        super.addSprite("Mauseloch mit Maus","Mauseloch.01.png");
    }

    public Boolean mausZuSehen ()
    {
        // Anfang Variablen
        // Ende Variablen

        return mausSichtbar;
    }

    public void mausZeigen (Boolean zeigen)
    {
        // Anfang Variablen
        // Ende Variablen

        mausSichtbar = zeigen;
        if (zeigen)
        {
            super.setSprite(1);
        }
        else
        {
            super.setSprite(0);
        }
    }
}
```

[Zurück zur Aufgabe auf Seite 9.](#)

5.12 Lösung zu Aufgabe 5

```
public class Game extends Stage
{
    // Anfang Attribute
    private Mauseloch[] mls;
    private Katze katze;
    // Ende Attribute

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

        super("Mäuse fangen", 661, 472);
        super.loadBackground("Hintergrund_gelb.png");
        katze = new Katze(330, 236);
        super.addStageObject(katze);
    }

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

        if (super.isKeyPressed("y"))
        {
            this.mausFangen(1);
        }
        else if (super.isKeyPressed("x"))
        {
            this.mausFangen(2);
        }
        else if (super.isKeyPressed("c"))
        {
            this.mausFangen(3);
        }
        else if (super.isKeyPressed("v"))
        {
            this.mausFangen(4);
        }
        else if (super.isKeyPressed("b"))
        {
            this.mausFangen(5);
        }
        else if (super.isKeyPressed("q"))
        {
            comp.sendExitSignal();
        }
        else
        {
            katze.warten();
        }
    }
}
```

```
public void mausFangen (Integer nr)
{
    // Anfang Variablen
    // Ende Variablen

    katze.fangen(nr);
}
}
```

[Zurück zur Aufgabe auf Seite 10.](#)

5.13 Lösung zu Aufgabe 6 (a)

```
public class Game extends Stage
{
    // Anfang Attribute
    private Mauseloch[] mls;
    private Katze katze;
    // Ende Attribute

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

        super("Mäuse fangen",661,472);
        super.loadBackground("Hintergrund_gelb.png");
        mls = new Mauseloch[5];
        mls[0] = new Mauseloch(100,380);
        super.addStageObject(mls[0]);
        katze = new Katze(330,236);
        super.addStageObject(katze);
    }

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

        if (super.isKeyPressed("y"))
        {
            this.mausFangen(1);
        }
        else if (super.isKeyPressed("x"))
        {
            this.mausFangen(2);
        }
        else if (super.isKeyPressed("c"))
        {
            this.mausFangen(3);
        }
        else if (super.isKeyPressed("v"))
        {
            this.mausFangen(4);
        }
        else if (super.isKeyPressed("b"))
        {
            this.mausFangen(5);
        }
        else if (super.isKeyPressed("q"))
        {
            comp.sendExitSignal();
        }
        else
        {
            katze.warten();
        }
    }
}
```

```
public void mausFangen (Integer nr)
{
    // Anfang Variablen
    // Ende Variablen

    katze.fangen(nr);
}
}
```

[Zurück zur Aufgabe auf Seite 10.](#)

5.14 Lösung zu Aufgabe 6 (b)

```
public class Game extends Stage
{
    // Anfang Attribute
    private Mauseloch[] mls;
    private Katze katze;
    // Ende Attribute

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

        super("Mäuse fangen", 661, 472);
        super.loadBackground("Hintergrund_gelb.png");
        super.addStageObject(punkteAnzeige);
        mls = new Mauseloch[5];
        mls[0] = new Mauseloch(100, 380);
        super.addStageObject(mls[0]);
        mls[1] = new Mauseloch(218, 380);
        super.addStageObject(mls[1]);
        mls[2] = new Mauseloch(336, 380);
        super.addStageObject(mls[2]);
        mls[3] = new Mauseloch(454, 380);
        super.addStageObject(mls[3]);
        mls[4] = new Mauseloch(572, 380);
        super.addStageObject(mls[4]);
        katze = new Katze(330, 236);
        super.addStageObject(katze);
    }

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

        if (super.isKeyPressed("y"))
        {
            this.mausFangen(1);
        }
        else if (super.isKeyPressed("x"))
        {
            this.mausFangen(2);
        }
        else if (super.isKeyPressed("c"))
        {
            this.mausFangen(3);
        }
        else if (super.isKeyPressed("v"))
        {
            this.mausFangen(4);
        }
        else if (super.isKeyPressed("b"))
        {
            this.mausFangen(5);
        }
    }
}
```

```
        else if (super.isKeyPressed("q"))
        {
            comp.sendExitSignal();
        }
        else
        {
            katze.warten();
        }
    }

    public void mausFangen (Integer nr)
    {
        // Anfang Variablen
        // Ende Variablen

        katze.fangen(nr);
    }
}
```

[Zurück zur Aufgabe auf Seite 10.](#)

5.15 Lösung zu Aufgabe 7

```
public class Game extends Stage
{
    // Anfang Attribute
    private Mauselloch[] mls;
    private Katze katze;
    private Integer maus;
    private Integer zeit;
    // Ende Attribute

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

        super("Mäuse fangen", 661, 472);
        super.loadBackground("Hintergrund_gelb.png");
        mls = new Mauselloch[5];
        mls[0] = new Mauselloch(100, 380);
        super.addStageObject(mls[0]);
        mls[1] = new Mauselloch(218, 380);
        super.addStageObject(mls[1]);
        mls[2] = new Mauselloch(336, 380);
        super.addStageObject(mls[2]);
        mls[3] = new Mauselloch(454, 380);
        super.addStageObject(mls[3]);
        mls[4] = new Mauselloch(572, 380);
        super.addStageObject(mls[4]);
        katze = new Katze(330, 236);
        super.addStageObject(katze);
        zeit = new Integer(15);
        maus = new Integer();
    }

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

        if (super.isKeyPressed("y"))
        {
            this.mausFangen(1);
        }
        else if (super.isKeyPressed("x"))
        {
            this.mausFangen(2);
        }
        else if (super.isKeyPressed("c"))
        {
            this.mausFangen(3);
        }
        else if (super.isKeyPressed("v"))
        {
            this.mausFangen(4);
        }
    }
}
```

```
        else if (super.isKeyPressed("b"))
        {
            this.mausFangen(5);
        }
        else if (super.isKeyPressed("q"))
        {
            comp.sendExitSignal();
        }
        else
        {
            katze.warten();
        }
    }

    public void mausFangen (Integer nr)
    {
        // Anfang Variablen
        // Ende Variablen

        katze.fangen(nr);
    }
}
```

[Zurück zur Aufgabe auf Seite 11.](#)

5.16 Lösung zu Aufgabe 8 (a)

```
public class Game extends Stage
{
    // Anfang Attribute
    private Mauselloch[] mls;
    private Integer zeit;
    private Katze katze;
    private Integer maus;
    // Ende Attribute

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

        super("Mäuse fangen", 661, 472);
        super.loadBackground("Hintergrund_gelb.png");
        mls = new Mauselloch[5];
        mls[0] = new Mauselloch(100, 380);
        super.addStageObject(mls[0]);
        mls[1] = new Mauselloch(218, 380);
        super.addStageObject(mls[1]);
        mls[2] = new Mauselloch(336, 380);
        super.addStageObject(mls[2]);
        mls[3] = new Mauselloch(454, 380);
        super.addStageObject(mls[3]);
        mls[4] = new Mauselloch(572, 380);
        super.addStageObject(mls[4]);
        katze = new Katze(330, 236);
        super.addStageObject(katze);
        zeit = new Integer(15);
        maus = new Integer();
    }

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

        if (super.isKeyPressed("y"))
        {
            this.mausFangen(1);
        }
        else if (super.isKeyPressed("x"))
        {
            this.mausFangen(2);
        }
        else if (super.isKeyPressed("c"))
        {
            this.mausFangen(3);
        }
        else if (super.isKeyPressed("v"))
        {
            this.mausFangen(4);
        }
    }
}
```

```
        else if (super.isKeyPressed("b"))
        {
            this.mausFangen(5);
        }
        else
        {
            katze.warten();
        }
        zeit = zeit.sub(1);
    }

    public void mausFangen (Integer nr)
    {
        // Anfang Variablen
        // Ende Variablen

        katze.fangen(nr);
    }
}
```

[Zurück zur Aufgabe auf Seite 11.](#)

5.17 Lösung zu Aufgabe 8 (b)

```
public class Game extends Stage
{
    // Anfang Attribute
    private Mauselloch[] mls;
    private Integer zeit;
    private Katze katze;
    private Integer maus;
    // Ende Attribute

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

        super("Mäuse fangen", 661, 472);
        super.loadBackground("Hintergrund_gelb.png");
        mls = new Mauselloch[5];
        mls[0] = new Mauselloch(100, 380);
        super.addStageObject(mls[0]);
        mls[1] = new Mauselloch(218, 380);
        super.addStageObject(mls[1]);
        mls[2] = new Mauselloch(336, 380);
        super.addStageObject(mls[2]);
        mls[3] = new Mauselloch(454, 380);
        super.addStageObject(mls[3]);
        mls[4] = new Mauselloch(572, 380);
        super.addStageObject(mls[4]);
        katze = new Katze(330, 236);
        super.addStageObject(katze);
        zeit = new Integer(15);
        maus = new Integer();
    }

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

        if (super.isKeyPressed("y"))
        {
            this.mausFangen(1);
        }
        else if (super.isKeyPressed("x"))
        {
            this.mausFangen(2);
        }
        else if (super.isKeyPressed("c"))
        {
            this.mausFangen(3);
        }
        else if (super.isKeyPressed("v"))
        {
            this.mausFangen(4);
        }
    }
}
```

```
        else if (super.isKeyPressed("b"))
        {
            this.mausFangen(5);
        }
        else
        {
            katze.warten();
        }
        zeit = zeit.sub(1);
        if (zeit.lowerThanOrEqual(0))
        {
            mls[maus].mausZeigen(FALSE);
        }
    }

    public void mausFangen (Integer nr)
    {
        // Anfang Variablen
        // Ende Variablen

        katze.fangen(nr);
    }
}
```

[Zurück zur Aufgabe auf Seite 11.](#)

5.18 Lösung zu Aufgabe 8 (c)

```
public class Game extends Stage
{
    // Anfang Attribute
    private Mauseloch[] mls;
    private Integer zeit;
    private Katze katze;
    private Integer maus;
    private Computer comp;
    // Ende Attribute

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

        super("Mäuse fangen",661,472);
        super.loadBackground("Hintergrund_gelb.png");
        mls = new Mauseloch[5];
        mls[0] = new Mauseloch(100,380);
        super.addStageObject(mls[0]);
        mls[1] = new Mauseloch(218,380);
        super.addStageObject(mls[1]);
        mls[2] = new Mauseloch(336,380);
        super.addStageObject(mls[2]);
        mls[3] = new Mauseloch(454,380);
        super.addStageObject(mls[3]);
        mls[4] = new Mauseloch(572,380);
        super.addStageObject(mls[4]);
        katze = new Katze(330,236);
        super.addStageObject(katze);
        zeit = new Integer(15);
        maus = new Integer();
        comp = new Computer();
    }

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

        if (super.isKeyPressed("y"))
        {
            this.mausFangen(1);
        }
        else if (super.isKeyPressed("x"))
        {
            this.mausFangen(2);
        }
        else if (super.isKeyPressed("c"))
        {
            this.mausFangen(3);
        }
    }
}
```

```
        else if (super.isKeyPressed("v"))
        {
            this.mausFangen(4);
        }
        else if (super.isKeyPressed("b"))
        {
            this.mausFangen(5);
        }
        else
        {
            katze.warten();
        }
        zeit = zeit.sub(1);
        if (zeit.lowerThanOrEqual(0))
        {
            mls[maus].mausZeigen(FALSE);
            maus = comp.rand(0,4);
        }
    }

    public void mausFangen (Integer nr)
    {
        // Anfang Variablen
        // Ende Variablen

        katze.fangen(nr);
    }
}
```

[Zurück zur Aufgabe auf Seite 11.](#)

5.19 Lösung zu Aufgabe 8 (d)

```
public class Game extends Stage
{
    // Anfang Attribute
    private Mauseloch[] mls;
    private Integer zeit;
    private Katze katze;
    private Integer maus;
    private Computer comp;
    // Ende Attribute

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

        super("Mäuse fangen",661,472);
        super.loadBackground("Hintergrund_gelb.png");
        mls = new Mauseloch[5];
        mls[0] = new Mauseloch(100,380);
        super.addStageObject(mls[0]);
        mls[1] = new Mauseloch(218,380);
        super.addStageObject(mls[1]);
        mls[2] = new Mauseloch(336,380);
        super.addStageObject(mls[2]);
        mls[3] = new Mauseloch(454,380);
        super.addStageObject(mls[3]);
        mls[4] = new Mauseloch(572,380);
        super.addStageObject(mls[4]);
        katze = new Katze(330,236);
        super.addStageObject(katze);
        zeit = new Integer(15);
        maus = new Integer();
        comp = new Computer();
    }

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

        if (super.isKeyPressed("y"))
        {
            this.mausFangen(1);
        }
        else if (super.isKeyPressed("x"))
        {
            this.mausFangen(2);
        }
        else if (super.isKeyPressed("c"))
        {
            this.mausFangen(3);
        }
    }
}
```

```
        else if (super.isKeyPressed("v"))
        {
            this.mausFangen(4);
        }
        else if (super.isKeyPressed("b"))
        {
            this.mausFangen(5);
        }
        else
        {
            katze.warten();
        }
        zeit = zeit.sub(1);
        if (zeit.lowerThanOrEqual(0))
        {
            mls[maus].mausZeigen(FALSE);
            maus = comp.rand(0,4);
            mls[maus].mausZeigen(TRUE);
        }
    }

    public void mausFangen (Integer nr)
    {
        // Anfang Variablen
        // Ende Variablen

        katze.fangen(nr);
    }
}
```

[Zurück zur Aufgabe auf Seite 11.](#)

5.20 Lösung zu Aufgabe 8 (e)

```
public class Game extends Stage
{
    // Anfang Attribute
    private Mauseloch[] mls;
    private Integer zeit;
    private Katze katze;
    private Integer maus;
    private Computer comp;
    // Ende Attribute

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

        super("Mäuse fangen",661,472);
        super.loadBackground("Hintergrund_gelb.png");
        mls = new Mauseloch[5];
        mls[0] = new Mauseloch(100,380);
        super.addStageObject(mls[0]);
        mls[1] = new Mauseloch(218,380);
        super.addStageObject(mls[1]);
        mls[2] = new Mauseloch(336,380);
        super.addStageObject(mls[2]);
        mls[3] = new Mauseloch(454,380);
        super.addStageObject(mls[3]);
        mls[4] = new Mauseloch(572,380);
        super.addStageObject(mls[4]);
        katze = new Katze(330,236);
        super.addStageObject(katze);
        zeit = new Integer(15);
        maus = new Integer();
        comp = new Computer();
    }

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

        if (super.isKeyPressed("y"))
        {
            this.mausFangen(1);
        }
        else if (super.isKeyPressed("x"))
        {
            this.mausFangen(2);
        }
        else if (super.isKeyPressed("c"))
        {
            this.mausFangen(3);
        }
    }
}
```

```
        else if (super.isKeyPressed("v"))
        {
            this.mausFangen(4);
        }
        else if (super.isKeyPressed("b"))
        {
            this.mausFangen(5);
        }
        else
        {
            katze.warten();
        }
        zeit = zeit.sub(1);
        if (zeit.lowerThanOrEqual(0))
        {
            mls[maus].mausZeigen(FALSE);
            maus = comp.rand(0,4);
            mls[maus].mausZeigen(TRUE);
            zeit = new Integer(5);
        }
    }

    public void mausFangen (Integer nr)
    {
        // Anfang Variablen
        // Ende Variablen

        katze.fangen(nr);
    }
}
```

[Zurück zur Aufgabe auf Seite 11.](#)

5.21 Lösung zu Aufgabe 9 (a)

```
public class Game extends Stage
{
    // Anfang Attribute
    private Computer comp;
    private Integer anzahlGefangen;
    private Mauselloch[] mls;
    private Integer zeit;
    private Integer anzahlMaeuse;
    private Katze katze;
    private Integer maus;
    // Ende Attribute

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

        super("Mäuse fangen",661,472);
        super.loadBackground("Hintergrund_gelb.png");
        mls = new Mauselloch[5];
        mls[0] = new Mauselloch(100,380);
        super.addStageObject(mls[0]);
        mls[1] = new Mauselloch(218,380);
        super.addStageObject(mls[1]);
        mls[2] = new Mauselloch(336,380);
        super.addStageObject(mls[2]);
        mls[3] = new Mauselloch(454,380);
        super.addStageObject(mls[3]);
        mls[4] = new Mauselloch(572,380);
        super.addStageObject(mls[4]);
        katze = new Katze(330,236);
        super.addStageObject(katze);
        zeit = new Integer(15);
        maus = new Integer();
        comp = new Computer();
        anzahlMaeuse = new Integer();
        anzahlGefangen = new Integer();
    }

    public void run ()
    {
        // Anfang Variablen
        Integer neueMaus;
        // Ende Variablen

        if (super.isKeyPressed("y"))
        {
            this.mausFangen(1);
        }
        else if (super.isKeyPressed("x"))
        {
            this.mausFangen(2);
        }
    }
}
```

```
        else if (super.isKeyPressed("c"))
        {
            this.mausFangen(3);
        }
        else if (super.isKeyPressed("v"))
        {
            this.mausFangen(4);
        }
        else if (super.isKeyPressed("b"))
        {
            this.mausFangen(5);
        }
        else
        {
            katze.warten();
        }
        zeit = zeit.sub(1);
        if (zeit.lowerThanOrEqual(0))
        {
            mls[maus].mausZeigen(FALSE);
            maus = comp.rand(0,5);
            mls[maus].mausZeigen(TRUE);
            zeit = new Integer(5);
        }
    }

    public void mausFangen (Integer nr)
    {
        // Anfang Variablen
        // Ende Variablen

        katze.fangen(nr);
    }
}
```

[Zurück zur Aufgabe auf Seite 12.](#)

5.22 Lösung zu Aufgabe 9 (b)

```
public class Game extends Stage
{
    // Anfang Attribute
    private StageObject punkteAnzeige;
    private Computer comp;
    private Integer anzahlGefangen;
    private Mauselloch[] mls;
    private Integer zeit;
    private Integer anzahlMaeuse;
    private Katze katze;
    private Integer maus;
    // Ende Attribute

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

        super("Mäuse fangen",661,472);
        super.loadBackground("Hintergrund_gelb.png");
        punkteAnzeige = new StageObject(330,25,100);
        punkteAnzeige.setTextAlignCenter();
        super.addStageObject(punkteAnzeige);
        mls = new Mauselloch[5];
        mls[0] = new Mauselloch(100,380);
        super.addStageObject(mls[0]);
        mls[1] = new Mauselloch(218,380);
        super.addStageObject(mls[1]);
        mls[2] = new Mauselloch(336,380);
        super.addStageObject(mls[2]);
        mls[3] = new Mauselloch(454,380);
        super.addStageObject(mls[3]);
        mls[4] = new Mauselloch(572,380);
        super.addStageObject(mls[4]);
        katze = new Katze(330,236);
        super.addStageObject(katze);
        zeit = new Integer(15);
        maus = new Integer();
        comp = new Computer();
        anzahlMaeuse = new Integer();
        anzahlGefangen = new Integer();
    }

    public void run ()
    {
        // Anfang Variablen
        Integer neueMaus;
        // Ende Variablen

        if (super.isKeyPressed("y"))
        {
            this.mausFangen(1);
        }
    }
}
```

```
        else if (super.isKeyPressed("x"))
        {
            this.mausFangen(2);
        }
        else if (super.isKeyPressed("c"))
        {
            this.mausFangen(3);
        }
        else if (super.isKeyPressed("v"))
        {
            this.mausFangen(4);
        }
        else if (super.isKeyPressed("b"))
        {
            this.mausFangen(5);
        }
        else
        {
            katze.warten();
        }
        zeit = zeit.sub(1);
        if (zeit.lowerThanOrEqual(0))
        {
            mls[maus].mausZeigen(FALSE);
            maus = comp.rand(0,5);
            mls[maus].mausZeigen(TRUE);
            zeit = new Integer(5);
        }
    }

    public void mausFangen (Integer nr)
    {
        // Anfang Variablen
        // Ende Variablen

        katze.fangen(nr);
    }
}
```

[Zurück zur Aufgabe auf Seite 12.](#)

5.23 Lösung zu Aufgabe 9 (c)

```
public class Game extends Stage
{
    // Anfang Attribute
    private StageObject punkteAnzeige;
    private Computer comp;
    private Integer anzahlGefangen;
    private Mauselloch[] mls;
    private Integer zeit;
    private Integer anzahlMaeuse;
    private Katze katze;
    private Integer maus;
    // Ende Attribute

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

        super("Mäuse fangen",661,472);
        super.loadBackground("Hintergrund_gelb.png");
        punkteAnzeige = new StageObject(330,25,100);
        punkteAnzeige.setTextAlignCenter();
        super.addStageObject(punkteAnzeige);
        mls = new Mauselloch[5];
        mls[0] = new Mauselloch(100,380);
        super.addStageObject(mls[0]);
        mls[1] = new Mauselloch(218,380);
        super.addStageObject(mls[1]);
        mls[2] = new Mauselloch(336,380);
        super.addStageObject(mls[2]);
        mls[3] = new Mauselloch(454,380);
        super.addStageObject(mls[3]);
        mls[4] = new Mauselloch(572,380);
        super.addStageObject(mls[4]);
        katze = new Katze(330,236);
        super.addStageObject(katze);
        zeit = new Integer(15);
        maus = new Integer();
        comp = new Computer();
        anzahlMaeuse = new Integer();
        anzahlGefangen = new Integer();
    }

    public void run ()
    {
        // Anfang Variablen
        Integer neueMaus;
        // Ende Variablen

        if (super.isKeyPressed("y"))
        {
            this.mausFangen(1);
        }
    }
}
```

```
        else if (super.isKeyPressed("x"))
        {
            this.mausFangen(2);
        }
        else if (super.isKeyPressed("c"))
        {
            this.mausFangen(3);
        }
        else if (super.isKeyPressed("v"))
        {
            this.mausFangen(4);
        }
        else if (super.isKeyPressed("b"))
        {
            this.mausFangen(5);
        }
        else
        {
            katze.warten();
        }
        zeit = zeit.sub(1);
        if (zeit.lowerThanOrEqual(0))
        {
            mls[maus].mausZeigen(FALSE);
            maus = comp.rand(0,5);
            mls[maus].mausZeigen(TRUE);
            zeit = new Integer(5);
        }
    }

    public void refreshAnzeige ()
    {
        // Anfang Variablen
        String msg;
        // Ende Variablen

        msg = new String();
        msg.append(anzahlGefangen);
        msg.append(" von ");
        msg.append(anzahlMaeuse);
        msg.append(" Mäusen gefangen.");
        punkteAnzeige.setText(msg,30);
    }

    public void mausFangen (Integer nr)
    {
        // Anfang Variablen
        // Ende Variablen

        katze.fangen(nr);
    }
}
```

[Zurück zur Aufgabe auf Seite 12.](#)

5.24 Lösung zu Aufgabe 9 (d)

```
public class Game extends Stage
{
    // Anfang Attribute
    private StageObject punkteAnzeige;
    private Computer comp;
    private Integer anzahlGefangen;
    private Mauselloch[] mls;
    private Integer zeit;
    private Integer anzahlMaeuse;
    private Katze katze;
    private Integer maus;
    // Ende Attribute

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

        super("Mäuse fangen", 661, 472);
        super.loadBackground("Hintergrund_gelb.png");
        punkteAnzeige = new StageObject(330, 25, 100);
        punkteAnzeige.setTextAlignCenter();
        super.addStageObject(punkteAnzeige);
        mls = new Mauselloch[5];
        mls[0] = new Mauselloch(100, 380);
        super.addStageObject(mls[0]);
        mls[1] = new Mauselloch(218, 380);
        super.addStageObject(mls[1]);
        mls[2] = new Mauselloch(336, 380);
        super.addStageObject(mls[2]);
        mls[3] = new Mauselloch(454, 380);
        super.addStageObject(mls[3]);
        mls[4] = new Mauselloch(572, 380);
        super.addStageObject(mls[4]);
        katze = new Katze(330, 236);
        super.addStageObject(katze);
        zeit = new Integer(15);
        maus = new Integer();
        comp = new Computer();
        anzahlMaeuse = new Integer();
        anzahlGefangen = new Integer();
    }

    public void run ()
    {
        // Anfang Variablen
        Integer neueMaus;
        // Ende Variablen

        if (super.isKeyPressed("y"))
        {
            this.mausFangen(1);
        }
    }
}
```

```
        else if (super.isKeyPressed("x"))
        {
            this.mausFangen(2);
        }
        else if (super.isKeyPressed("c"))
        {
            this.mausFangen(3);
        }
        else if (super.isKeyPressed("v"))
        {
            this.mausFangen(4);
        }
        else if (super.isKeyPressed("b"))
        {
            this.mausFangen(5);
        }
        else
        {
            katze.warten();
        }
        zeit = zeit.sub(1);
        if (zeit.lowerThanOrEqual(0))
        {
            mls[maus].mausZeigen(FALSE);
            maus = comp.rand(0,5);
            mls[maus].mausZeigen(TRUE);
            zeit = new Integer(5);
            anzahlMaeuse = anzahlMaeuse.add(1);
            this.refreshAnzeige();
        }
    }

    public void refreshAnzeige ()
    {
        // Anfang Variablen
        String msg;
        // Ende Variablen

        msg = new String();
        msg.append(anzahlGefangen);
        msg.append(" von ");
        msg.append(anzahlMaeuse);
        msg.append(" Mäusen gefangen.");
        punkteAnzeige.setText(msg,30);
    }

    public void mausFangen (Integer nr)
    {
        // Anfang Variablen
        // Ende Variablen

        katze.fangen(nr);
    }
}
```

[Zurück zur Aufgabe auf Seite 12.](#)

5.25 Lösung zu Aufgabe 9 (e)

```
public class Game extends Stage
{
    // Anfang Attribute
    private StageObject punkteAnzeige;
    private Computer comp;
    private Integer anzahlGefangen;
    private Mauselloch[] mls;
    private Integer zeit;
    private Integer anzahlMaeuse;
    private Katze katze;
    private Integer maus;
    // Ende Attribute

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

        super("Mäuse fangen",661,472);
        super.loadBackground("Hintergrund_gelb.png");
        punkteAnzeige = new StageObject(330,25,100);
        punkteAnzeige.setTextAlignCenter();
        super.addStageObject(punkteAnzeige);
        mls = new Mauselloch[5];
        mls[0] = new Mauselloch(100,380);
        super.addStageObject(mls[0]);
        mls[1] = new Mauselloch(218,380);
        super.addStageObject(mls[1]);
        mls[2] = new Mauselloch(336,380);
        super.addStageObject(mls[2]);
        mls[3] = new Mauselloch(454,380);
        super.addStageObject(mls[3]);
        mls[4] = new Mauselloch(572,380);
        super.addStageObject(mls[4]);
        katze = new Katze(330,236);
        super.addStageObject(katze);
        zeit = new Integer(15);
        maus = new Integer();
        comp = new Computer();
        anzahlMaeuse = new Integer();
        anzahlGefangen = new Integer();
    }

    public void run ()
    {
        // Anfang Variablen
        Integer neueMaus;
        // Ende Variablen

        if (super.isKeyPressed("y"))
        {
            this.mausFangen(1);
        }
    }
}
```

```
        else if (super.isKeyPressed("x"))
        {
            this.mausFangen(2);
        }
        else if (super.isKeyPressed("c"))
        {
            this.mausFangen(3);
        }
        else if (super.isKeyPressed("v"))
        {
            this.mausFangen(4);
        }
        else if (super.isKeyPressed("b"))
        {
            this.mausFangen(5);
        }
        else
        {
            katze.warten();
        }
        zeit = zeit.sub(1);
        if (zeit.lowerThanOrEqual(0))
        {
            mls[maus].mausZeigen(FALSE);
            maus = comp.rand(0,5);
            mls[maus].mausZeigen(TRUE);
            zeit = new Integer(5);
            anzahlMaeuse = anzahlMaeuse.add(1);
            this.refreshAnzeige();
        }
    }

    public void refreshAnzeige ()
    {
        // Anfang Variablen
        String msg;
        // Ende Variablen

        msg = new String();
        msg.append(anzahlGefangen);
        msg.append(" von ");
        msg.append(anzahlMaeuse);
        msg.append(" Mäusen gefangen.");
        punkteAnzeige.setText(msg,30);
    }
```

```
public void mausFangen (Integer nr)
{
    // Anfang Variablen
    // Ende Variablen

    katze.fangen(nr);
    nr = nr.sub(1);
    if (mls[nr].mausZuSehen())
    {
        anzahlGefangen = anzahlGefangen.add(1);
        this.refreshAnzeige();
    }
}
```

[Zurück zur Aufgabe auf Seite 12.](#)

5.26 Lösung zu Aufgabe 10 (a)

```
public class Game extends Stage
{
    // Anfang Attribute
    private StageObject punkteAnzeige;
    private Computer comp;
    private Integer anzahlGefangen;
    private Mauselloch[] mls;
    private Integer zeit;
    private Integer anzahlMaeuse;
    private Katze katze;
    private Integer maus;
    // Ende Attribute

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

        super("Mäuse fangen", 661, 472);
        super.loadBackground("Hintergrund_gelb.png");
        punkteAnzeige = new StageObject(330, 25, 100);
        punkteAnzeige.setTextAlignCenter();
        super.addStageObject(punkteAnzeige);
        mls = new Mauselloch[5];
        mls[0] = new Mauselloch(100, 380);
        super.addStageObject(mls[0]);
        mls[1] = new Mauselloch(218, 380);
        super.addStageObject(mls[1]);
        mls[2] = new Mauselloch(336, 380);
        super.addStageObject(mls[2]);
        mls[3] = new Mauselloch(454, 380);
        super.addStageObject(mls[3]);
        mls[4] = new Mauselloch(572, 380);
        super.addStageObject(mls[4]);
        katze = new Katze(330, 236);
        super.addStageObject(katze);
        zeit = new Integer(15);
        maus = new Integer();
        comp = new Computer();
        anzahlMaeuse = new Integer();
        anzahlGefangen = new Integer();
    }

    public void run ()
    {
        // Anfang Variablen
        Integer neueMaus;
        // Ende Variablen

        if (super.isKeyPressed("y"))
        {
            this.mausFangen(1);
        }
    }
}
```

```
        else if (super.isKeyPressed("x"))
        {
            this.mausFangen(2);
        }
        else if (super.isKeyPressed("c"))
        {
            this.mausFangen(3);
        }
        else if (super.isKeyPressed("v"))
        {
            this.mausFangen(4);
        }
        else if (super.isKeyPressed("b"))
        {
            this.mausFangen(5);
        }
        else if (super.isKeyPressed("q"))
        {
            comp.sendExitSignal();
        }
        else
        {
            katze.warten();
        }
        zeit = zeit.sub(1);
        if (zeit.lowerThanOrEqual(0))
        {
            mls[maus].mausZeigen(FALSE);
            maus = comp.rand(0,5);
            mls[maus].mausZeigen(TRUE);
            zeit = new Integer(5);
            anzahlMaeuse = anzahlMaeuse.add(1);
            this.refreshAnzeige();
        }
    }

    public void refreshAnzeige ()
    {
        // Anfang Variablen
        String msg;
        // Ende Variablen

        msg = new String();
        msg.append(anzahlGefangen);
        msg.append(" von ");
        msg.append(anzahlMaeuse);
        msg.append(" Mäusen gefangen.");
        punkteAnzeige.setText(msg,30);
    }
```

```
public void mausFangen (Integer nr)
{
    // Anfang Variablen
    // Ende Variablen

    katze.fangen(nr);
    nr = nr.sub(1);
    if (mls[nr].mausZuSehen())
    {
        anzahlGefangen = anzahlGefangen.add(1);
        this.refreshAnzeige();
    }
}
```

[Zurück zur Aufgabe auf Seite 13.](#)

5.27 Lösung zu Aufgabe 10 (b)

```
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(200);
        comp = new Computer();
        comp.print("Taste  Aktion");
        comp.println();
        comp.print(" y      Maus Nr. 1 fangen");
        comp.println();
        comp.print(" x      Maus Nr. 2 fangen");
        comp.println();
        comp.print(" c      Maus Nr. 3 fangen");
        comp.println();
        comp.print(" v      Maus Nr. 4 fangen");
        comp.println();
        comp.print(" b      Maus Nr. 5 fangen");
        comp.println();
        comp.print(" q      Programm beenden");
        comp.waitForExitSignal();
    }
}
```

[Zurück zur Aufgabe auf Seite 13.](#)

5.28 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 Fehler: Referenz nicht gefunden.](#)

5.29 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 Fehler: Referenz nicht gefunden.](#)

5.30 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 Fehler: Referenz nicht gefunden.](#)

5.31 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 Fehler: Referenz nicht gefunden.](#)

5.32 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 Fehler: Referenz nicht gefunden.](#)

5.33 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 Fehler: Referenz nicht gefunden.](#)