Dipl.-Inform. Annett Thüring
M.Sc. Janek Bevendorff
M.Sc. Vaibhav Kasturia

Martin-Luther-Universität Halle-Wittenberg
Institut für Informatik
Wintersemester 2018/19

Objektorientierte Programmierung: Hausaufgabenblatt 12

Abgabe: 28.01.2019, 11:00

Das Ziel dieses Übungsblattes ist es, dass Sie Ihre Kenntnisse zu Klassenvererbung und dem Überschreiben von Methoden festigen.

Hausaufgabe 1: (15 Punkte)

Bitte beachten Sie: die Einsendung der Aufgabe erfolgt nur im YAPEX.
Open exercises via code → Freigabecode: `5d0fbokkpfh2-bc40` (Beutespiel)

In dieser Aufgabe soll ein auf dem Schachspiel basierendes kleines Beute-Jäger-Spiel umgesetzt werden. Implementieren Sie dazu die folgenden Klassen.

Hinweis: Denken Sie grundsätzlich daran die korrekten `@`-Annotationen beim Überschreiben zu verwenden und Klassenattribute als `private` und ggf. `final` zu markieren!

SpielFigur (5 Punkte) Objekte dieser Klassen besitzen insgesamt drei von außen nicht zugreifbare Attribute: `xPos`, `yPos` und `farbe`. Es muss sichergestellt werden, dass zu jeder Zeit `xPos` nur Werte zwischen `A` und `H`, `yPos` nur Werte zwischen 1 und 8 und `farbe` nur die Werte `"weiss"` oder `"schwarz"` annehmen (daraus ergeben sich auch die entsprechenden Datentypen). Der Konstruktor erhält die `xPos` und `yPos` als Parameter des entsprechenden Datentyps sowie einen boolean-Wert (`true` falls die Farbe schwarz ist und `false` falls die Farbe weiß ist). Sollten bezüglich der Position falsche Werte übergeben worden sein, so wird der am nächsten liegende erlaubte Wert angenommen. Implementieren Sie nun noch die folgenden Methoden:

- `getXPos`, `getYPos` und `getFarbe`, welche den Wert des entsprechenden Attributes zurückgeben.
- `toString`, welche eine Zeichenkette im Format `"schwarze Figur (Feld A1)"` zurückgibt (ohne Zeilenumbruch am Ende, wobei schwarz die Farbe ist und A1 sich aus der x- und y-Position zusammensetzt).
- `istNeuePositionKorrekt`, welche als Parameter die neue einzunehmende x- und y-Position übergeben bekommt sowie `true` zurückgibt, falls sich die neue Position auf dem Spielfeld befindet und `false` zurückgibt, falls sich die neue Position außerhalb des Spielfeldes befindet. Diese Methode darf nur von dieser Klasse bzw. von Unterklassen aufrufbar sein!
- `ziehe`, welche als Parameter die Anzahl der zu ziehenden Felder in x- und y-Richtung erhält (positive Werte stehen für eine Bewegung nach rechts bzw. oben). Die Methode berechnet nun die neue Position und überprüft sie mit der Methode `istNeuePositionKorrekt`. Falls die neue Position legitim ist, nimmt die Figur auch die Position ein und es wird `true` zurückgegeben. Andernfalls wird die neue Position nicht eingenommen und `false` zurückgegeben.
- `trifft` erhält als Parameter ein Objekt vom Typ `SpielFigur` und gibt `true`, falls die eigene Position mit der Position des übergebenen Objektes übereinstimmt und in allen anderen Fällen `false` zurück.

DameFigur (5 Punkte) Diese Klasse erbt von der Klasse `SpielFigur`. Als zusätzliches Attribut erhält sie `name`, welches als Wert `"Dame"` enthält. Es muss über die Deklaration sichergestellt werden, dass dieses von außen nicht zugreifbare Attribut nicht mehr geändert werden kann! Des Weiteren sollen sie

- die Methode `istNeuePositionKorrekt` überschreiben. Für die Überprüfung, ob sich die Position außerhalb des Spielfeldes befindet, muss auf die überschriebene Methode (siehe Oberklasse) zurückgegriffen werden. Wenn die Figur sich mit der neuen Position nicht außerhalb des Spielfeldes befindet, muss überprüft werden, ob die Figur sich korrekt bewegt hat (die Dame darf in jede Richtung linear oder diagonal ziehen[1]). Sie gibt dann entsprechend `true` oder `false` zurück.
- die Methode `ziehe` überladen. Die überladende Methode erhält als Parameter die Richtung (`-` und `|` für eine lineare sowie `/` und `\` für eine diagonale Bewegung) und Anzahl der zu ziehenden Felder (ein positiver Wert steht für rechts bzw. oben bei der linearen Bewegung und bei der diagonalen Bewegung für eine Bewegung nach rechts unten/oben). Anhand der beiden Parameter muss sie die überladene Methode (siehe Oberklasse) aufrufen.
- die Methode `toString` überschreiben, welche dann eine Zeichenkette im Format `"weisse Dame (Feld D4)"` zurückgibt.

BeuteSpiel (5 Punkte) Hierbei handelt es sich um das eigentliche kleine Jäger-Beute-Spiel. Implementieren Sie in dieser Klasse einen Konstruktor, der als Parameter einen x- und einen y-Wert übergeben bekommt und daraus ein `DameFigur`-Objekt (Farbe `"weiss"`) erzeugt. Dieses Objekt ist die „Beute“. Implementieren Sie dazu eine `start`-Methode (die automatisch von der vorgegebenen `Main`-Klasse aufgerufen wird). In dieser darf der Spieler per Eingabe ein Objekt ebenfalls dieser Klasse erzeugen und die entsprechende Position angeben. Dieses Objekt ist somit der Jäger. Wurde bei der Positionierung bereits die Beute erwischt, erfolgt eine entsprechende Ausgabe und das Programm wird beendet. Andernfalls hat der Spieler bis zu 10 mögliche Versuche, seine Spielfigur zu bewegen, wobei zu Beginn jeweils die Spielfigur mit ihrer aktuellen Position ausgegeben wird. Als Benutzereingabe muss der Spieler eins der Symbole `-`, `|`, `/` oder `\` sowie die Anzahl der Felder eingeben. Bei einer ungültigen Bewegung wird der Benutzer erneut zur Eingabe aufgefordert! Sobald der Spieler die Beute fängt, erfolgt eine entsprechende Ausgabe und das Programm wird beendet. Endet das Spiel, ohne dass die Beute gefangen wurde, erfolgt ebenfalls eine entsprechende Ausgabe, wobei zusätzlich die Position der Beute mit ausgegeben wird. Ein *möglicher* Durchlauf könnte folgendermaßen aussehen:

[1] Sie darf an sich auch keine anderen Figuren überspringen, dies wird hier aber bewusst außer Acht gelassen!

Beispieldurchlauf

```
Die Beute steht. Positionieren Sie den Jaeger
Spalte (A bis H) Ihrer Figur? E
Zeile (1 bis 8) Ihrer Figur? 4
Die Beute-Figur steht woanders!
Sie haben nun 10 Dame-Zuege, um die Beute-Figur zu treffen.
Bewegen Sie Ihre schwarze Dame (Feld E4)
Wollen Sie waagrecht (-), senkrecht (|) oder diagonal (/, \) ziehen? \
Wie viele Felder ziehen? (< 0 nach links oben, > 0 nach rechts unten) -3
Leider kein Treffer!
Bewegen Sie Ihre schwarze Dame (Feld B7)
Wollen Sie waagrecht (-), senkrecht (|) oder diagonal (/, \) ziehen? -
Wie viele Felder ziehen? (< 0 nach links, > 0 nach rechts) 4
Leider kein Treffer!
Bewegen Sie Ihre schwarze Dame (Feld F7)
Wollen Sie waagrecht (-), senkrecht (|) oder diagonal (/, \) ziehen? /
Wie viele Felder ziehen? (< 0 nach links unten, > 0 nach rechts oben) -2
Leider kein Treffer!
Bewegen Sie Ihre schwarze Dame (Feld D5)
Wollen Sie waagrecht (-), senkrecht (|) oder diagonal (/, \) ziehen? |
Wie viele Felder ziehen? (< 0 nach oben, > 0 nach unten) -4
Leider kein Treffer!
Bewegen Sie Ihre schwarze Dame (Feld D1)
Wollen Sie waagrecht (-), senkrecht (|) oder diagonal (/, \) ziehen? |
Wie viele Felder ziehen? (< 0 nach oben, > 0 nach unten) 8
Ungueltige Bewegung!
Bewegen Sie Ihre schwarze Dame (Feld D1)
Wollen Sie waagrecht (-), senkrecht (|) oder diagonal (/, \) ziehen? |
Wie viele Felder ziehen? (< 0 nach oben, > 0 nach unten) 7
Treffer! Sie (als Jaeger) haben gewonnen
```