Grundlagen der Programmierung

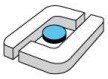
Aufgabenblatt 9 (Eigenes OO-Programm, Vererbung)

Prof. Dr.-Ing. Heiko Tapken / Programmier-Team

Testat: KW 50

Wintersemester 2018/19

Bestehensgrenze: 10 Punkte



Hochschule Osnabrück
University of Applied Sciences

Aufgabe 1 (RushHour) [10 Punkte]

Rushhour ist ein Spiel für einen einzelnen Spieler. Auf einem rechteckigen Spielbrett, das aus einzelnen Kacheln besteht, wird initial eine Menge an Autos platziert. Jedes Auto überdeckt mindestens zwei Kacheln. Ein Auto mit mehr als zwei Kacheln darf dabei keine Ecken aufweisen. Liegen die Kacheln, die ein Auto überdeckt, nebeneinander, ist es ein horizontal verschiebbares Auto. Liegen die Kacheln übereinander, ist es ein vertikal verschiebbares Auto. Ein Auto wird als Hauptauto gekennzeichnet. Ziel des Spiels ist es nun, die Autos so zu verschieben, bis irgendwann das Hauptauto (im Bild bspw. das rote Auto) den rechten Rand berührt.

Horizontale Autos dürfen dabei nur nach links und rechts, vertikale Autos nur nach oben und unten verschoben werden. Ein Auto darf nicht über den Rand hinweg oder wenn ein anderes Auto im Wege steht, verschoben werden.

Entwickeln Sie ein Java-Programm, mit dem ein Benutzer Rushhour spielen kann. Die Ausgangsstellung der Autos sei dabei in einem von Ihnen zu wählenden Datenformat festgelegt. Die Autoteile werden dabei durch jeweils gleiche Zeichen repräsentiert; das Hauptauto durch das Zeichen ‚*'. Die Stellung der Autos in der oberen Abbildung entspräche **z.B.** folgendem Kodierung:

```
122_3_
145_36
145**6
7778_6
__98aa
bb9cc_
```

Ein * steht dabei für das Hauptauto; _ kennzeichnen leere Felder.

Der Benutzer soll nun solange Autos verschieben können (durch Angabe von Auto und Richtung) bis das Hauptauto den rechten Rand erreicht. Im Folgenden wird ein beispielhafter Programmablauf skizziert (Benutzereingaben in <>):

Grundlagen der Programmierung

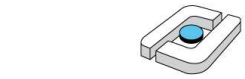
Aufgabenblatt 9 (Eigenes OO-Programm, Vererbung)

Prof. Dr.-Ing. Heiko Tapken / Programmier-Team

Testat: KW 50

Wintersemester 2018/19

Bestehensgrenze: 10 Punkte



Hochschule Osnabrück
University of Applied Sciences

```
+----+----+----+----+----+----+
| 1  | 2  | 2  |    | 3  |    |
+----+----+----+----+----+----+
| 1  | 4  | 5  |    | 3  | 6  |
+----+----+----+----+----+----+
| 1  | 4  | 5  | *  | *  | 6
+----+----+----+----+----+----+
| 7  | 7  | 7  | 8  |    | 6  |
+----+----+----+----+----+----+
|    |    | 9  | 8  | a  | a  |
+----+----+----+----+----+----+
| b  | b  | 9  | c  | c  |    |
+----+----+----+----+----+----+

Wahl eines Autos (Buchstabe angeben): <c>
Richtung (l=links, r=rechts, u=hoch, d=runter, q=Autowechsel): <r>

+----+----+----+----+----+----+
| 1  | 2  | 2  |    | 3  |    |
+----+----+----+----+----+----+
| 1  | 4  | 5  |    | 3  | 6  |
+----+----+----+----+----+----+
| 1  | 4  | 5  | *  | *  | 6
+----+----+----+----+----+----+
| 7  | 7  | 7  | 8  |    | 6  |
+----+----+----+----+----+----+
|    |    | 9  | 8  | a  | a  |
+----+----+----+----+----+----+
| b  | b  | 9  |    | c  | c  |
+----+----+----+----+----+----+

Richtung (l=links, r=rechts, u=hoch, d=runter, q=Autowechsel): <q>
Wahl eines Autos (Buchstabe angeben): 8
Richtung (l=links, r=rechts, u=hoch, d=runter, q=Autowechsel): <d>
+----+----+----+----+----+----+
| 1  | 2  | 2  |    | 3  |    |
+----+----+----+----+----+----+
| 1  | 4  | 5  |    | 3  | 6  |
+----+----+----+----+----+----+
| 1  | 4  | 5  | *  | *  | 6
+----+----+----+----+----+----+
| 7  | 7  | 7  |    |    | 6  |
+----+----+----+----+----+----+
|    |    | 9  | 8  | a  | a  |
+----+----+----+----+----+----+
| b  | b  | 9  | 8  | c  | c  |
+----+----+----+----+----+----+

Richtung (l=links, r=rechts, u=hoch, d=runter, q=Autowechsel): <q>
Wahl eines Autos (Buchstabe angeben): 7
Richtung (l=links, r=rechts, u=hoch, d=runter, q=Autowechsel): <r>
```

```
+----+----+----+----+----+----+
| 1  | 2  | 2  |    | 3  |    |
+----+----+----+----+----+----+
| 1  | 4  | 5  |    | 3  | 6  |
+----+----+----+----+----+----+
| 1  | 4  | 5  | *  | *  | 6
+----+----+----+----+----+----+
|    | 7  | 7  | 7  |    | 6  |
+----+----+----+----+----+----+
|    |    | 9  | 8  | a  | a  |
+----+----+----+----+----+----+
| b  | b  | 9  | 8  | c  | c  |
+----+----+----+----+----+----+
```

…

Bis das Auto rausgefahren wird.

Um einen Eindruck für das Spiel zu bekommen: http://www.spielen.com/spiel/rush_hour_2.html

Aufgabe 2 (Vererbung 1) [3 Punkte]

```
class Grossvater {
  int x = 3;
  int y = -4;
}

class Vater extends Grossvater {
  float x = 4.5F;
  int z;
  public Vater(int z) {
   this.z = z;
  }
}

class Sohn extends Vater {
  long a;
  double x = -18.5;
  public Sohn(long a) {
    super(5);
    this.a = a;
  }
}
```

Leiten Sie von der Klasse Sohn eine Klasse Enkel ab, die eine Methode besitzt, in der die Werte aller Attribute, die ein Objekte der Klasse Enkel besitzt (auch die geerbten!), addiert werden und die die Summe auf den Bildschirm ausgibt. Schreiben Sie ein kleines Testprogramm!