

Übung 11: Entwurfsmuster

Abgabetermin: 18/JUN/2009, 13:00

Name: _____ Matrikelnummer: _____

Aufgabe	Punkte	abzugeben schriftlich	abzugeben elektronisch	korr.	Pkte
Übung 11	24	Java-Programm, Screenshot des Programms	Java-Programm	<input type="checkbox"/>	

Animierte Figuren

Schreiben Sie ein Programm, das in einem Panel Bewegungen von unterschiedlichen Figuren animiert. Die Anwendung soll wie folgt aufgebaut sein:

Figuren:

Eine Figur hat eine Position und ist aus einer Reihe von graphischen Objekten (z.B. Rectangles) aufgebaut. Realisieren Sie mehrere Typen von unterschiedlichen Figuren (mindestens 2), die unterschiedlichen Aufbau haben.

Dekoratoren für Figuren:

Realisieren Sie Dekoratoren, mit denen man Figuren mit „visuellen Zusätzen“ versehen kann. Überlegen Sie sich mehrere Dekoratoren (mindestens 2). Verwenden Sie

- das *Decorator-Pattern*, um für die Figuren Dekoratoren zu realisieren.

Darstellung der Figuren:

Die Figuren sollen in einer graphischen Oberfläche dargestellt werden. Realisieren Sie dazu ein eigenes Panel, welches eine Reihe von Figuren darstellen kann.

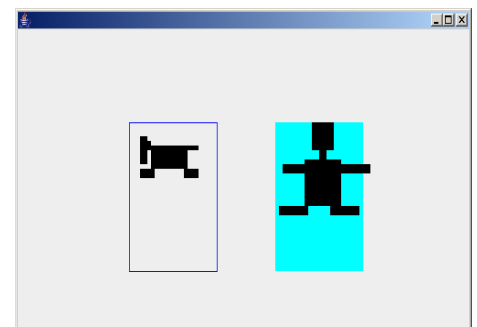
Bewegungen:

Bei den Figuren soll es nun möglich sein, Bewegungen durchzuführen. Die Bewegungen werden dabei als Veränderungen der graphischen Objekte der Figur realisiert. Allerdings sollen die Bewegungen nicht durch die Figuren selbst, sondern in Visitor-Objekten durchgeführt werden (dadurch ist es möglich, immer neue Bewegungen bei den Figuren zu realisieren). Verwenden Sie

- das *Visitor-Pattern*, um unterschiedliche Bewegungen für die Figuren zu realisieren.
- das *Chain-of-Responsibility-Pattern*, um die Anwendung des Visitors von Dekoratoren an die eigentlichen Figuren weiterzuleiten.

Animation:

Stellen Sie die Bewegungen in der graphischen Oberfläche dar. Realisieren Sie dazu eine Animationsroutine (siehe unten), welche in einer Endlosschleife für alle Figuren die gleichen Bewegungen (= Anwendung des gleichen Visitors) durchführt, dann eine bestimmte Zeit wartet (Methode `Thread.sleep(time)`) und dann mit einer neuen Bewegung fortfährt.



Hinweise zur Animation:

Animierte Bewegungen realisiert man am besten unter Verwendung eines eigenen Threads. Threads werden als Objekte der Klasse `Thread` angelegt, wobei ein `Runnable`-Objekt übergeben wird, d.h. ein Objekt, welches das Interface `Runnable` implementiert und somit eine `void run()`-Methode hat.

```
Thread thread = new Thread(new SimpleAnimator());
```

Mit

```
thread.start();
```

wird der Thread gestartet und die Methode `run` des `Runnable`-Objekts ausgeführt.

Im folgenden Beispiel wird `SimpleAnimator` als ein eigene `JComponent` realisiert, die ein Rechteck verschiebt und vergrößert und diese Bewegung animiert. Dazu wird

- die Veränderung des Rechtecks durchgeführt
- mit einem Aufruf von `repaint()` das Neuzeichnen des Panels veranlasst
- dann mit `Thread.sleep(10)` der Threads für 10 ms passiviert (`Thread.sleep` kann eine `InterruptedException` werfen und muss daher mit einem `try-catch`-Block gekapselt werden).

```
public class SimpleAnimator extends JComponent implements Runnable {  
  
    Rectangle rect;  
  
    public SimpleAnimator() {  
        rect = new Rectangle(0, 0, 50, 50);  
    }  
  
    public void paint(Graphics g) {  
        g.setColor(getBackground());  
        g.fillRect(0, 0, getWidth(), getHeight());  
        g.setColor(getForeground());  
        g.drawRect(rect.x, rect.y, rect.w, rect.h);  
    }  
  
    public void run() {  
        while (true) {  
            rect.move(1, 1);  
            rect.enlarge(1);  
  
            repaint();  
  
            try {  
                Thread.sleep(10);  
            } catch (InterruptedException e) {}  
        }  
    }  
}
```