

## Übung 20: GoF Patterns *Decorator für Klasse Car* Testatübung 3: Abgabe per KW49 – 7.12.14 23 Uhr

### Hinweise

- Diese Übung ist individuell zu lösen.
- Die Lösung ist spätestens per 7.12. 23:00 per Mail als Export eines Eclipse-Projektes in eine Zip-Datei mit Name *U20-NameVorname.zip* an Ihren Übungsbetreuer abzugeben.
- Für Ihre Lösung erhalten Sie ein individuelles Feedback.
- Die Testatübung gilt als erfüllt, wenn der Betreuer die Lösung als genügend beurteilt.

### Aufgabe

Eine Klasse Car sieht wie folgt aus:

```
public class Car {  
    private String description;  
    private int price;  
  
    public Car(String description, int price) {  
        this.description = description;  
        this.price = price;  
    }  
  
    public String getDescription() {  
        return description;  
    }  
  
    public int getPrice() {  
        return price;  
    }  
}
```

Car besitzt also eine Beschreibung und einen Preis. Nun soll es mindestens drei Zusatzoptionen wie zum Beispiel Klimaanlage usw. (Ihnen fällt schon etwas ein) geben. Jede Zusatzoptionen (durch eine separate Klasse implementiert<sup>1</sup>) hat auch eine Beschreibung, welche die Beschreibung des Cars ergänzt, und einen Preis, der sich zum Preis des Cars addiert.

Cars mit mehreren Zusatzoptionen (d.h. mit mehreren Decorators) können in einer *Simple Factory* erzeugt werden. Die Factory soll konfiguriert werden können, damit sie Cars mit bestimmten Zusatzoptionen liefert.

Schreiben Sie JUnit-Tests um zu testen, ob Ihre Lösung korrekt funktioniert.

---

<sup>1</sup> Natürlich könnte man in diesem einfachen Beispiel auch eine einzige Decorator-Klasse *Zusatzoption* verwenden und die unterschiedlichen Optionen einfach in unterschiedlichen Attributwerten abbilden. Verwenden Sie aber für jede Zusatzoption eine eigene Klasse, in welcher Beschreibung und Preis fest als Literale codiert werden.