



Problem

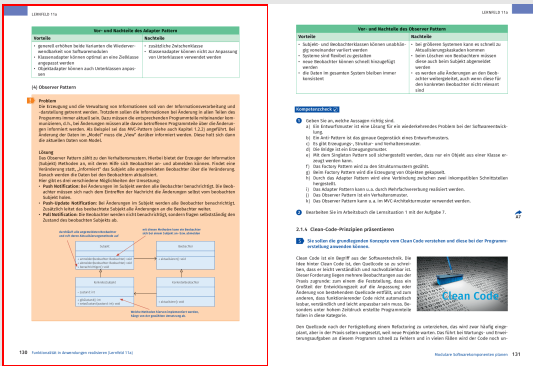
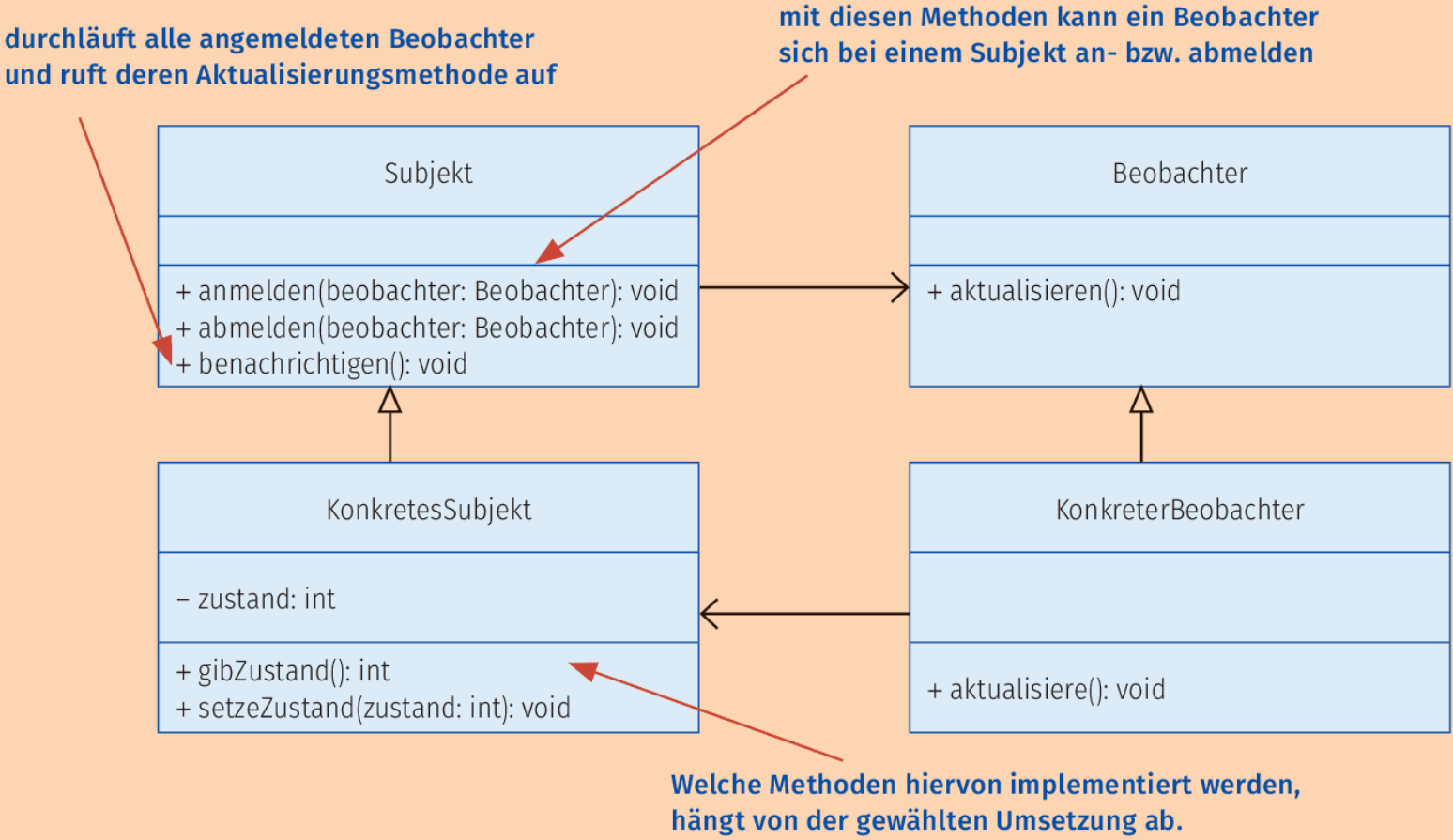
Die Erzeugung und die Verwaltung von Informationen soll von der Informationsverarbeitung und -darstellung getrennt werden. Trotzdem sollen die Informationen bei Änderung in allen Teilen des Programms immer aktuell sein. Dazu müssen die entsprechenden Programmteile miteinander kommunizieren, d. h., bei Änderungen müssen alle davon betroffenen Programmteile über die Änderungen informiert werden. Als Beispiel sei das MVC-Pattern (siehe auch Kapitel 1.2.2) angeführt. Bei Änderung der Daten im „Model“ muss die „View“ darüber informiert werden. Diese holt sich dann die aktuellen Daten vom Model.

Lösung

Das Observer Pattern zählt zu den Verhaltensmustern. Hierbei bietet der Erzeuger der Information (Subjekt) Methoden an, mit deren Hilfe sich Beobachter an- und abmelden können. Findet eine Veränderung statt, „informiert“ das Subjekt alle angemeldeten Beobachter über die Veränderung. Danach werden die Daten bei den Beobachtern aktualisiert.

Hier gibt es drei verschiedene Möglichkeiten der Umsetzung:

- **Push Notification:** Bei Änderungen im Subjekt werden alle Beobachter benachrichtigt. Die Beobachter müssen sich nach dem Eintreffen der Nachricht die Änderungen selbst vom beobachteten Subjekt holen.
- **Push-Update Notification:** Bei Änderungen im Subjekt werden alle Beobachter benachrichtigt. Zusätzlich leitet das beobachtete Subjekt alle Änderungen an die Beobachter weiter.
- **Pull Notification:** Die Beobachter werden nicht benachrichtigt, sondern fragen selbstständig den Zustand des beobachteten Subjekts ab.



Interaktive Lerneinheiten - Digitalisierung

0

Interaktive Lerneinheiten - Fachinformatik

0

Prüfungsvorbereitung online - Demo

0

