

## Hinweise

- Markieren Sie bei jeder Frage die am besten passende(n) Antwort(en)!
- Es gibt zwei Arten von Multiple-Choice-Fragen:
  - Fragen, die keine Punktzahl in Klammern angeben. Diese haben exakt eine richtige Antwort.
  - Fragen, die eine Punktzahl angeben. Hier müssen alle richtigen Antworten angekreuzt werden. Die Anzahl der korrekten Antworten kann sich von der Anzahl der Punkte unterscheiden.
- Falsch beantwortete Multiple-Choice-Fragen wirken sich negativ auf die Gesamtpunktzahl aus.

## Builder Pattern, Geschachtelte Klassen (I)

### Lektion 21

#### Bei einem Fluent Interface

- werden mit settern nacheinander Werte auf einem Objekt gesetzt.
- wird von jeder Methode das gleiche Objekt zurückgegeben, um weitere Aufrufe durchzuführen.
- wird von jeder Methode ein Objekt zurückgegeben, auf dem weitere Aufrufe erfolgen können.

#### Method Chaining ist

- die Verwendung von with-Methoden.
- das Aneinanderreihen von Methodenaufrufen.

#### Ein Fluent Interface endet

- durch die Methode, die nicht dieselbe Objektreferenz wie die anderen Methode zurückgibt.
- in jedem Fall durch eine Methode, die void zurückgibt.

#### Wofür wird das Builder Pattern verwendet? Markieren Sie alle richtigen Antwortmöglichkeiten! (2)

- Um den Konstruktor vor anderen Klassen zu verstecken.
- Um Fehler bei langen Konstruktoraufufen zu vermeiden.
- Um Klassen flexibel um Funktionalitäten zu erweitern.
- Um flexibel Objekte mit vielen Attributen zu erzeugen, ohne für jede mögliche Attributkombination einen eigenen Konstruktor schreiben zu müssen.

#### Eine Klasse kann vor Fremdzugriffen geschützt werden, indem sie... (2)

- mit package-Sichtbarkeit deklariert wird.
- als private geschachtelte Klasse in einer anderen Klasse auftaucht.
- als geschachtelte Klasse in einer anderen Klasse auftaucht.

**Eine innere Klasse kann durch new <NameDerÄußenenKlasse>.<NameDerInnerenKlasse> instanziert werden, wenn es einen sichtbaren Konstruktor gibt.**

- Falsch
- Richtig

**Eine innere Klasse hat Zugriff auf die privaten Attribute der äußeren Klasse.**

- Richtig
- Falsch

**Objekte innerer Klassen halten immer eine Referenz auf das Objekt der äußeren Klasse.**

- Falsch
- Richtig

**Java erlaubt durch Autoboxing einen int in eine List<Integer> einzufügen. Die Klasse Integer dient dabei als ein Adapter für int.**

- Falsch
- Richtig

**Durch den protected-Modifier im Klassenkopf kann man anderen Klassen verbieten von der eigenen Klasse zu erben.**

- Richtig
- Falsch

**Eine Adapter-Klasse lässt eine Instanz einer Klasse wie eine Instanz einer anderen Klasse aussehen.**

- Richtig
- Falsch

**Eine statisch geschachtelte Klasse kann nur abhängig von der äußeren Klasse genutzt werden.**

- Falsch
- Richtig

**Eine statisch geschachtelte Klasse kann ohne ein Objekt der äußeren Klasse instantiiert werden.**

- Richtig
- Falsch