

Test 2 in Programmierparadigmen am 26. 1. 2026

Beispiele für **markierte** (= ausgewählte) Auswahlfelder:

Beispiele für **nicht markierte** (= nicht ausgewählte) Auswahlfelder:

Beurteilung: 1 Punkt pro Zeile mit gänzlich richtig markierten Auswahlfeldern

Aufgabe 1 (10 Punkte)

A und B seien definiert durch `interface A<T> {}` und `interface B<R, S> extends A<S> {}`.
Bitte markieren Sie jedes Auswahlfeld, bei dem der links stehende Typ ein Untertyp des darüber stehenden Typs ist. Es können keines, eines oder mehrere Felder pro Zeile auszuwählen sein.

| | A<String> | A<? super String> | B<? extends String, ?> |
|-------------------------------------|--------------------------|--------------------------|--------------------------|
| A<Object> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| A<String> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| B<String, Object> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| B<Object, String> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| B<?, ? extends String> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| B<? extends String, ?> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| B<? super String, ? extends String> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| B<? extends String, ? super String> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| B<A<String>, Object> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| B<A<Object>, String> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Aufgabe 2 (10 Punkte)

Bitte markieren Sie in jeder Zeile das eine Auswahlfeld, bei dem die links stehende Aussage am ehesten eine Eigenschaft der darüber stehenden Parametrisierungsform in Java oder AspectJ ist.

| | Generizität | Annotationen | Aspekte |
|---|--------------------------|--------------------------|--------------------------|
| Join-Points sind dynamisch identifizierbare Programmstellen | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| kann die Werkzeuge zur Programmerzeugung beeinflussen | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| „*“ steht für beliebige Zeichen außer „.“ (Punkt) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| für binäre Methoden wird Rekursion benötigt | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| zur Laufzeit sind Daten über Reflexion zugreifbar | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| wird für Kernfunktionalität meist gemieden | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Lücken werden durch spezielle Parameter dargestellt | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| around() ersetzt den spezifizierten Programmtext | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| mit @Retention wird die Lebensdauer festgelegt | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Typschranken legen vorausgesetzte Funktionalität fest | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Aufgabe 3 (10 Punkte)

Bitte markieren Sie jedes Auswahlfeld, bei dem der links stehende Typausdruck (mit Typnamen aus den Paketen `java.util.function` und `java.lang`) ein Typ des darüber stehenden Lambdas ist. Es können keines, eines oder mehrere Felder pro Zeile auszuwählen sein.

| | $x \rightarrow y \rightarrow x+y$ | $(x, y) \rightarrow x+y$ | $x \rightarrow x+2$ | $x \rightarrow \{\}$ | <code>Long::sum</code> |
|---|-----------------------------------|--------------------------|-----------------------|-----------------------|------------------------|
| <code>BinaryOperator<String></code> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| <code>LongBinaryOperator</code> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| <code>BiFunction<Long, Long, Long></code> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| <code>Function<Long, Function<Long, Long>></code> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| <code>Function<Long, LongUnaryOperator></code> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| <code>LongUnaryOperator</code> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| <code>Consumer<String></code> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| <code>Function<Long, Function<String, String>></code> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| <code>BiFunction<Long, String, Long></code> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| <code>BiFunction<String, Long, String></code> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Aufgabe 4 (10 Punkte)

Bitte markieren Sie jedes Auswahlfeld, bei dem die links stehende Kommandozeile (in `bash` ausgeführt) die darüber stehende Auswirkung hat. Es können keines, eines oder mehrere Felder pro Zeile auszuwählen sein.

| | | | | | | | | | | | |
|---|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <code>cat < a wc > b</code> | <input type="radio"/> |
| <code>cat a & wc &>> b</code> | <input type="radio"/> |
| <code>cat a &> b && wc b</code> | <input type="radio"/> |
| <code>cat a ; wc a</code> | <input type="radio"/> |
| <code>for i in * ; do cat \$i wc > ../wc/\$i ; done</code> | <input type="radio"/> |
| <code>for i in * ; do (cat \$i > ../bak/\$i &) ; done</code> | <input type="radio"/> |
| <code>if test 'cat a' = "a b c" ; then wc b ; else wc c ; fi</code> | <input type="radio"/> |
| <code>cat a wc b</code> | <input type="radio"/> |
| <code>cat < a & wc &> b &</code> | <input type="radio"/> |
| <code>cat wc</code> | <input type="radio"/> |

mehrere Prozesse laufen gleichzeitig

Prozesse laufen im Hintergrund

Prozesse über Pipeline verbunden

Standardeingabe von `cat` umgeleitet

Standardausgabe von `cat` umgeleitet

Fehlerausgabe von `cat` umgeleitet

Aufgabe 5 (10 Punkte)

Bitte markieren Sie jedes Auswahlfeld, bei dem die links stehende Eigenschaft auf den darüber stehenden Methodenaufruf zutrifft. Es können keines, eines oder mehrere Felder pro Zeile auszuwählen sein.

muss mit `InterruptedException` umgehen
 darf nur in einem Monitor verwendet werden
 soll nicht in einem Monitor verwendet werden
 gehört nicht zu Synchronisationsmechanismen
 suspendiert Ausführung des aktuellen Threads
 suspendiert Ausführung auf bestimmte Zeit
 weckt systemweit alle wartenden Threads auf
 weckt höchstens einen wartenden Threads auf
 weckt mindestens einen wartenden Thread auf
 gibt den Lock vorübergehend frei

| | Thread.sleep(5) | wait() | notify() | notifyAll() |
|--|-----------------------|-----------------------|-----------------------|-----------------------|
| | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Aufgabe 6 (10 Punkte)

Bitte markieren Sie jedes Auswahlfeld, bei dem die links stehende Aussage eine Eigenschaft des darüber stehenden Entwurfsmusters ist. Es können keines, eines oder mehrere Felder pro Zeile auszuwählen sein.

unterstützt Umkehrung der Abhängigkeiten
 führt zu vielen kleinen Objekten
 für oberflächliche Erweiterungen geeignet
 oft große Anzahl an Unterklassen nötig
 ist Entwurfsmuster für Struktur
 ist erzeugendes Entwurfsmuster
 hilft große Zahl an Klassen zu vermeiden
 beruht auf Delegation
 häufig als innere Klasse implementiert
 Objektidentität ist damit unzuverlässig

| | Iterator | Decorator | Proxy | Prototype | Factory-Method |
|--|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| | <input type="radio"/> |
| | <input type="radio"/> |
| | <input type="radio"/> |
| | <input type="radio"/> |
| | <input type="radio"/> |
| | <input type="radio"/> |
| | <input type="radio"/> |
| | <input type="radio"/> |
| | <input type="radio"/> |
| | <input type="radio"/> |

Aufgabe 7 (10 Punkte)

Bitte markieren Sie in jeder Zeile das eine Auswahlfeld, bei dem die links stehende Aussage am ehesten eine Eigenschaft des darüber stehenden Entwurfsmusters ist.

zyklische Strukturen bereiten Probleme
 robuste Varianten werden bevorzugt
 Verantwortlichkeiten wieder entziehbar
 schlecht geeignet für umfangreiche Objekte
 mehrere gleichzeitige Abarbeitungen möglich
 kein Decorator, aber gleiche Struktur möglich
 kann mit kovarianten Problemen umgehen
 es gibt externe und interne Varianten
 flache von tiefen Kopien unterschieden
 führt zu parallelen Klassenhierarchien

| | Iterator | Decorator | Proxy | Prototype | Factory-Method |
|--|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| | <input type="radio"/> |
| | <input type="radio"/> |
| | <input type="radio"/> |
| | <input type="radio"/> |
| | <input type="radio"/> |
| | <input type="radio"/> |
| | <input type="radio"/> |
| | <input type="radio"/> |
| | <input type="radio"/> |
| | <input type="radio"/> |

Aufgabe 8 (10 Punkte)

Bitte markieren Sie in jeder Zeile das eine Auswahlfeld, bei dem die links stehende Aussage am ehesten eine Eigenschaft des darüber stehenden Entwurfsmusters ist.

| | Iterator | Decorator | Proxy | Prototype | Factory-Method |
|--|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <i>Component</i> ist ein Bestandteil | <input type="radio"/> |
| <i>Product</i> ist ein Bestandteil | <input type="radio"/> |
| <i>Subject</i> ist ein Bestandteil | <input type="radio"/> |
| <i>Creator</i> ist ein Bestandteil | <input type="radio"/> |
| <i>Aggregate</i> ist ein Bestandteil | <input type="radio"/> |
| <i>Smart-Reference</i> ist eine Variante davon | <input type="radio"/> |
| wird auch <i>Wrapper</i> genannt | <input type="radio"/> |
| wird auch <i>Surrogate</i> genannt | <input type="radio"/> |
| wird auch <i>Virtual-Constructor</i> genannt | <input type="radio"/> |
| wird auch <i>Cursor</i> genannt | <input type="radio"/> |

Aufgabe 9 (10 Punkte)

Bitte markieren Sie jedes Auswahlfeld, bei dem die links stehende Aussage eine Eigenschaft des darüber stehenden Entwurfsmusters ist. Es können keines, eines oder mehrere Felder pro Zeile auszuwählen sein.

| | Visitor | Singleton | Template-Method | Factory-Method |
|---|-----------------------|-----------------------|-----------------------|-----------------------|
| verwaltet verwandte Operationen zentral | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| für Lazy-Initialization geeignet | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| gibt Grundgerüst von Algorithmus vor | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Anzahl erzeugter Objekte kontrollierbar | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| kann zu vielen Klassen führen | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| kann zu vielen Methoden führen | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| verwendet häufig Hooks | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| kann zu vielen Objekten führen | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| mehrere Arten primitiver Operationen | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| beruht auf dem Hollywood-Prinzip | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Aufgabe 10 (10 Punkte)

Bitte markieren Sie in jeder Zeile das eine Auswahlfeld, bei dem die links stehende Aussage am ehesten eine Eigenschaft des darüber stehenden Entwurfsmusters ist.

| | Visitor | Singleton | Template-Method | Factory-Method |
|---|-----------------------|-----------------------|-----------------------|-----------------------|
| <i>accept</i> hat einen Parameter | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| <i>instance</i> ist meist static | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| <i>Element</i> ist ein Bestandteil | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| hilft Duplikate im Code zu vermeiden | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| kann als Anti-Pattern gesehen werden | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Lambdas können Schreibaufwand reduzieren | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| eine Methode ist häufig final | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| globale Variable ist Alternative dazu | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| unterstützt direkte Code-Wiederverwendung | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Umgang mit Unterklassen problematisch | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |