

Name:

Matrikelnummer:

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="radio"/>	<input type="radio"/>	<input type="radio"/>
A<String>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
B<String, Object>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
B<Object, String>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
B<?, ? extends String>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
B<? extends String, ?>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
B<? super String, ? extends String>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
B<? extends String, ? super String>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
B<A<String>, Object>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
B<A<Object>, String>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

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

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.

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

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.

	mehrere Prozesse laufen gleichzeitig	Prozesse laufen im Hintergrund	Prozesse über Pipeline verbunden	Standardeingabe von <code>cat</code> umgeleitet	Standardausgabe von <code>cat</code> umgeleitet	Fehlerausgabe von <code>cat</code> umgeleitet
<code>cat < a wc > b</code>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<code>cat a & wc &>> b</code>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<code>cat a &> b && wc b</code>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<code>cat a ; wc a</code>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<code>for i in * ; do cat \$i wc > ../wc/\$i ; done</code>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<code>for i in * ; do (cat \$i > ../bak/\$i &) ; done</code>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<code>if test 'cat a' = "a b c" ; then wc b ; else wc c ; fi</code>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<code>cat a wc b</code>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<code>cat < a & wc &> b &</code>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<code>cat wc</code>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

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.

	Thread.sleep(5)	wait()	notify()	notifyAll()
muss mit InterruptedException umgehen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
darf nur in einem Monitor verwendet werden	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
soll nicht in einem Monitor verwendet werden	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
gehört nicht zu Synchronisationsmechanismen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
suspendiert Ausführung des aktuellen Threads	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
suspendiert Ausführung auf bestimmte Zeit	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
weckt systemweit alle wartenden Threads auf	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
weckt höchstens einen wartenden Threads auf	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
weckt mindestens einen wartenden Thread auf	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
gibt den Lock vorübergehend frei	<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.

	Iterator	Decorator	Proxy	Prototype	Factory-Method
unterstützt Umkehrung der Abhängigkeiten	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
führt zu vielen kleinen Objekten	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
für oberflächliche Erweiterungen geeignet	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
oft große Anzahl an Unterklassen nötig	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
ist Entwurfsmuster für Struktur	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
ist erzeugendes Entwurfsmuster	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
hilft große Zahl an Klassen zu vermeiden	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
beruht auf Delegation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
häufig als innere Klasse implementiert	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Objektidentität ist damit unzuverlässig	<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.

	Iterator	Decorator	Proxy	Prototype	Factory-Method
zyklische Strukturen bereiten Probleme	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
robuste Varianten werden bevorzugt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Verantwortlichkeiten wieder entziehbar	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
schlecht geeignet für umfangreiche Objekte	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
mehrere gleichzeitige Abarbeitungen möglich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
kein Decorator, aber gleiche Struktur möglich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
kann mit kovarianten Problemen umgehen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
es gibt externe und interne Varianten	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
flache von tiefen Kopien unterschieden	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
führt zu parallelen Klassenhierarchien	<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"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<i>Product</i> ist ein Bestandteil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<i>Subject</i> ist ein Bestandteil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<i>Creator</i> ist ein Bestandteil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<i>Aggregate</i> ist ein Bestandteil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<i>Smart-Reference</i> ist eine Variante davon	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
wird auch <i>Wrapper</i> genannt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
wird auch <i>Surrogate</i> genannt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
wird auch <i>Virtual-Constructor</i> genannt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
wird auch <i>Cursor</i> genannt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<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"/>