

Name:

Matrikelnummer:

Test 1 in Programmierparadigmen am 25. 11. 2024 (B)

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)

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

	α -Konversion	β -Konversion	η -Konversion	
ist von der Form: $[e/v](\lambda v.f) \equiv \lambda v.f$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	✓
ist (mit $v \notin fv(e)$) von der Form: $\lambda u.e \equiv \lambda v.[v/u]e$	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	✓
ist von der Form: $(\lambda v.f) e \equiv [e/v]f$	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	✓
ist von der Form: $[e/v](f g) \equiv ([e/v]f) ([e/v]g)$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	✓
ist (mit $v \notin fv(e)$) von der Form: $\lambda v.(e v) \equiv e$	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	✓
dient zur Umbenennung eines Parameters	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	✓
kann eine λ -Abstraktion auch ohne Argument vereinfachen	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	✓
wendet eine λ -Abstraktion auf ein Argument an	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	✓
ist von links nach rechts und rechts nach links anwendbar	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	✓
davon gibt es eine gerichtete Variante als Reduktionsregel	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	✓

Aufgabe 2 (10 Punkte)

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

	Klasse	Objekt	Komponente	Modul	
definiert sich über Identität, Zustand und Verhalten	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	✓
wird durch „Deployment“ eingebunden	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	✓
ist per Definition eine Übersetzungseinheit	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	✓
ist statisch	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	✓
hat einen eindeutigen Namen	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	✓
lässt offen, von wo Inhalte importiert werden	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	✓
Inhalte davon sind in Java durch <code>static</code> gekennzeichnet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	✓
kann zur Laufzeit auch durch Kopieren erzeugt werden	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	✓
erlaubt zyklische Abhängigkeiten	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	✓
dient als Vorlage für die Objekterzeugung	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	✓

Aufgabe 3 (10 Punkte)

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

	nominal	strukturell	λ -Abstraktion
durch Kommentare beschriebene Methode ist von dieser Art	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
eine abstrakte Methode ohne Kommentar ist von dieser Art	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
eine Variable, die eine Funktion enthält, kann von dieser Art sein	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
diese Art ist in der objektorientierten Programmierung essenziell	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Änderung der Signatur einer Methode ändert die Abstraktion	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Änderung des Namens einer Methode ändert die Abstraktion	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
inhaltliche Änderung des Programmcodes ändert die Abstraktion	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
inhaltliche Änderung eines Kommentars ändert die Abstraktion	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kommentare haben dafür keine Bedeutung, sie werden ignoriert	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Compiler garantieren die Einhaltung der Einschränkungen	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Aufgabe 4 (10 Punkte)

Bitte markieren Sie in jeder Zeile das eine Auswahlfeld, bei dem die links stehende Aussage für das darüber stehende Paradigma am besten zutrifft.

	funktional	objektorientiert	prozedural
keine Unterscheidung zwischen Original und Kopie (Daten)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
dynamisches Binden verhindert nachvollziehbaren Kontrollfluss	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Kommunikation über Variablen wird örtlich eingegrenzt	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Kommunikation über gemeinsame Variablen wird verhindert	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
gemeinsame Variablen sind nur sparsam einzusetzen	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Kontrollfluss muss nachvollziehbar sein	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
referentielle Transparenz statt nachvollziehbarem Kontrollfluss	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
nominale Abstraktion statt nachvollziehbarem Kontrollfluss	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Aliase sind möglichst zu vermeiden	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
offensiver Umgang mit Aliasen (sind sichtbar und häufig)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Aufgabe 5 (10 Punkte)

Bitte markieren Sie in jeder Zeile das eine Auswahlfeld, bei dem die links stehende Aussage für das darüber stehende Paradigma am besten zutrifft.

	funktional	objektorientiert	prozedural	nebenläufig	parallel
gut für Programmierung im Groben	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
gut für komplexe Algorithmen	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
gut für große, langlebige Projekte	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ziel ist eine möglichst kurze Laufzeit	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
unterstützt algebraische Datentypen	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
am häufigsten eingesetztes Paradigma	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
es wird häufig auf Ereignisse gewartet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Programmablauf ist gut kontrollierbar	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interprozesskommunikation nötig	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Synchronisation (z.B. Monitor) nötig	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Aufgabe 6 (10 Punkte)

Bitte markieren Sie in jeder Zeile ein Auswahlfeld, bei dem die links stehende Aussage am ehesten die darüber stehende Beziehung zwischen zwei Typen bzw. Klassen U und T adressiert (für passend gewählte U und T).

	Vererbung	Delegation	Untertyp
in U werden Aufrufe an eine Variable vom Typ T weitergeleitet	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
manchmal ist es egal, ob U von T oder T von U abgeleitet wird	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
wenn U von T und T von U ableitbar, dann sind U und T äquivalent	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Code kann ohne Typableitung von T nach U übernommen werden	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
U soll durch Ableitung möglichst viel Code aus T übernehmen	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Objekt von U ersetzt Objekt von T	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
<u>Implementierung</u> von U ist einfacher wenn T von U abgeleitet	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Objekt von U statt Objekt von T verwendbar	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
es geht um Wiederverwendung von Code, der auf T bzw. U zugreift	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
es geht um Wiederverwendung von Code aus T in U	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

Aufgabe 7 (10 Punkte)

Bitte markieren Sie in jeder Zeile das eine Auswahlfeld, bei dem der links stehende Typ bei Untertypbeziehungen (wie angegeben bei allgemeinen strukturellen Untertypbeziehungen oder in Java) die darüber stehende Varianz hat, oder „n/a“ wenn es keinen solchen Typ gibt.

	kovariant	kontravariant	invariant	n/a
Typ von Konstante (strukturell)	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Typ von Variable (strukturell)	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Typ von Eingangsparmeter (strukturell)	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Typ von Durchgangsparmeter (strukturell)	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Typ von Ausgangsparmeter (strukturell)	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ergebnistyp von Methode (strukturell)	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Typ von Eingangsparmeter (in Java)	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Typ von Durchgangsparmeter (in Java)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Typ von Ausgangsparmeter (in Java)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Ergebnistyp von Methode (in Java)	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Aufgabe 8 (10 Punkte)

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

	Vorbed.	Nachbed.	Invariante	Client-kontroll. History-Constr.	Server-kontroll. History-Constr.
im Untertyp schwächer oder gleich als im Obertyp	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
im Untertyp stärker oder gleich als im Obertyp	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
bezieht sich immer auf Objektzustände von this	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
bezieht sich auf einzelne Methoden	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
zuständig ist ausnahmslos der Client	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
zuständig ist ausnahmslos der Server	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Server und Client (in Sonderfällen) sind zuständig	<input type="radio"/>				
schränkt die Aufrufreihenfolge von Methoden ein	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
gilt zu Beginn einer Methodenausführung	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
gilt am Ende einer Methodenausführung	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

10

Aufgabe 9 (10 Punkte)

Folgende Klasse ist gegeben:

```
public class CountingSwapper {
    private int cnt = 0;
    public void swap(int[] a, int i) { int h = a[i]; a[i] = a[i+1]; a[i+1] = h; cnt++; }
    public int count() { return cnt; }
}
```

Bitte markieren Sie in jeder Zeile ein Auswahlfeld, bei dem die links stehende Aussage am ehesten eine auf diese Klasse bezogene Zusicherung der über dem Auswahlfeld stehenden Art sein kann.

	Vorbed.	Nachbed.	Invariante	Client-kontroll. History-Constr.	Server-kontroll. History-Constr.
swap tauscht a[i] und a[i+1]	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
swap(a,i) nur aufrufbar wenn a[i] > a[i+1]	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ausnahme in swap(a,i) wenn i > a.length-2	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
i >= 0 und i < a.length-1	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
cnt kann nur größer werden	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
cnt >= 0	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
count retourniert Zahl der Aufrufe von swap	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
count gibt Wert größer oder gleich 0 zurück	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
d==c+1 nach c=count(); swap(a,i); d=count();	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
a != null	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

✓
✓
✓
✓
✓
✓
✓
✓
✓
✓

Aufgabe 10 (10 Punkte)

Folgende Klasse ist gegeben:

```
public class SimpleBufferMemory {
    private int data;
    public void set(int d) { data = d; }
    public int get() { return data; }
}
```

9

Bitte markieren Sie in jeder Zeile ein Auswahlfeld, bei dem die links stehende Aussage am ehesten eine auf diese Klasse bezogene Zusicherung der über dem Auswahlfeld stehenden Art sein kann.

	Vorbed.	Nachbed.	Invariante	Client-kontroll. History-Constr.	Server-kontroll. History-Constr.
x.set(x.get()) lässt Zustand von x unverändert	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
data ist eine ungerade Zahl	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
d ist eine ungerade Zahl	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ergebnis von get ist eine ungerade Zahl	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
set und get nur abwechselnd aufrufbar	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
in neuem Objekt ist zuerst set aufzurufen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
ohne set liefert get wiederholt gleiches Ergebnis	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
d==c gilt nach x.set(c); d=x.get();	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
get nur aufrufbar wenn davor set aufgerufen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
x.set(v) nur aufrufbar wenn v > x.get()	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

✓
✓
✓
✓
✓
✓
✓
x
✓
✓