

Programmiermethodik, Termin 1 SS21 (online) – Teil 1

Prüfung war geteilt in 2 Teile. Teil 1 mit vielen kurzen Fragen und Teil 2 mit wenigen längeren Fragen.

Zeit: 60min

Bei wahr/falsch Fragen: +1 Punkt pro Frage wenn richtig, wenn falsch -0,5 Punkte, wenn unbeantwortet 0 Punkte. Die Frage in sich kann wenigstens 0 Punkte geben (also keine gesamten Minuspunkte)

Bei multiple Choice Fragen: entweder alles richtig angekreuzt oder nicht (aber keine Minuspunkte)

Bei Lückentext: keine Minuspunkte und je nach dem wie viele Lücken man richtig hat bekommt man + Punkte

Frage 1 (wahr/falsch)

3 Punkte

Geben Sie jeweils an, ob die angegebene Aussage wahr oder falsch ist! Die Aussagen beziehen sich auf JUnit5!

- Eine JUnit-Testmethode muss mit einer entsprechenden Annotation (z.B. @Test, @Parameterized) gekennzeichnet sein.
- Eine JUnit-Verwaltungsmethode muss den Methodennamen BeforeEach, AfterEach, BeforeAll oder AfterAll haben.
- Das JUnit-Testframework ist Teil des JDKs

Frage 2 (Gültig/Ungültig)

3 Punkte

Kreuzen Sie für jeden Lambda-Ausdruck an, ob er gültig ist oder nicht!

- $() \rightarrow 15$
- $x, y \rightarrow \{x + y\}$
- $x \rightarrow x + x$

Frage 3 (wahr/falsch)

4 Punkte

Kreuzen Sie jeweils an, ob die angegebene Aussage wahr oder falsch ist!

- Eine abstrakte Klasse muss zumindest die von implementierten Interfaces bereitgestellten abstrakten Methoden implementieren.
- Eine abstrakte Klasse muss abstrakte Methoden haben.
- Eine abstrakte Klasse kann einen Konstruktor haben.
- Eine abstrakte Klasse kann von mehreren anderen abstrakten Klassen erben.

Frage 4 (wahr/falsch)

3 Punkte

Kreuzen Sie jeweils an, ob die angegebene Aussage wahr oder falsch ist!

- Durch Reference Counting können Zyklen im Objektgraph erkannt werden.
- Bei Stop-and-Copy wird der Heap in zwei gleich große Teile unterteilt.
- Jeglicher Speicher muss in Java manuell mit dem Aufruf `System.gc()` freigegeben werden.

Frage 5 (wahr/falsch)

3 Punkte

Kreuzen Sie jeweils an, ob die angegebene Aussage über Java Assertions (z.B. `assert x > 0;`) wahr oder falsch ist!

- Da Assertions aktiviert und deaktiviert werden können, sollten die Ausdrücke keine Seiteneffekte (z.B. Objektzustand verändern) aufweisen.
- Assertions werden zur Kompilierzeit ausgewertet.
- Assertions sind standardmäßig deaktiviert

Frage 6 (Klassenbezogen/Objektbezogen)

3 Punkte

Kreuzen Sie jeweils an, ob sich das Element auf ein Objekt oder eine Klasse bezieht.

- Instanzvariablen
- Statische Variablen
- super-Referenz
- Die main-Methode
- Formaler Typparameter
- Klassenvariablen

Frage 7 (wahr/falsch)

4 Punkte

Kreuzen Sie jeweils an, ob die angegebene Aussage wahr oder falsch ist!

- Ein Interface kann einen Konstruktor implementieren.
- Ein Interface kann andere Interfaces erweitern.
- Ein Interface kann eine Klasse implementieren.
- Ein Interface kann Methoden implementieren.

Frage 8 (wahr/falsch)

3 Punkte

Kreuzen Sie jeweils an, ob die angegebene Aussage wahr oder falsch ist!

- Die von Object geerbte Implementierung von equals vergleicht den Hash-Code von Objekten.
- Wenn zwei Objekte denselben Hash-Code aufweisen, müssen sie auch gemäß der Methode equals gleich sein.
- Eine korrekte equals-Implementierung muss neben weiteren Bedingungen reflexiv, symmetrisch und transitiv sein.

Frage 9 (Lückentext)

3 Punkte

Gegeben seien die folgenden Klassen.

```
package at.ac.uibk.pm.exam;  
public class A { }
```

```
package at.ac.uibk.pm.exam;  
public class B extends A { }
```

```
package at.ac.uibk.pm.exam;  
public class C extends B { }
```

```
package at.ac.uibk.pm.exam;  
  
public class Main {  
    public static void main(String[] args) {  
        A var1 = new B();  
        B var2 = new C();  
        var var3 = new B();  
    }  
}
```

Füllen Sie den Lückentext aus!

Die Variable var 1 hat den statischen Typ _____ und den dynamischen Typ _____.

Die Variable var 2 hat den statischen Typ _____ und den dynamischen Typ _____.

Die Variable var 3 hat den statischen Typ _____ und den dynamischen Typ _____.

Frage 10 (wahr/falsch)

4 Punkte

Kreuzen Sie jeweils an, ob die angegebene Aussage wahr oder falsch ist!

- Eine konkrete Klasse muss zumindest die von implementierten Interfaces bereitgestellten abstrakten Methoden implementieren.
- Eine konkrete Klasse muss explizit einen Konstruktor implementieren.
- Eine konkrete Klasse kann mehrere Interfaces implementieren.
- Eine konkrete Klasse kann abstrakte Methoden haben.

Frage 11 (wahr/falsch)

3 Punkte

Kreuzen Sie jeweils an, ob die angegebene Aussage wahr oder falsch ist!

- Die Reihenfolge von catch-Blöcken in einer try-Anweisung ist immer irrelevant.
- In Java folgt die Ausnahmebehandlung dem Termination Model.
- Der Kontrollfluss kehrt beim Resumption Model nach der Abarbeitung des catch-Blocks an die Stelle zurück, an der die Ausnahme aufgetreten ist.

Frage 12 (wahr/falsch)

3 Punkte

Kreuzen Sie für jedes Statement an, ob es erfolgreich kompiliert werden kann.

- `Consumer<String> consumer = System.out::println;`
- `Function<int, int> function = (x, y) -> x + y;`
- `Predicate<List<Integer>> predicate = Collection::isEmpty;`

Frage 13 (wahr/falsch)

3 Punkte

Kreuzen Sie jeweils an, ob die angegebene Aussage wahr oder falsch ist!

- Wenn MySubClass eine Unterklasse von MyClass ist, können alle Instanzen von MyClass die Instanzen von MySubClass ersetzen.
- Eine Klasse kann eine Methode, die sie von einer Superklasse erbt, neu implementieren. Dies wird als Überschreiben bezeichnet.
- Objektvariablen haben einen statischen und dynamischen Typ

Frage 14 (wahr/falsch)

3 Punkte

Kreuzen Sie jeweils an, ob die angegebene Aussage wahr oder falsch ist!

- Eine Objektvariable vom Typ Double, welche nicht final ist, wird automatisch mit null initialisiert.
- Eine Klassenvariable vom Typ float, welche nicht final ist, wird automatisch mit 0.0f initialisiert.
- Eine lokale Variable vom typ int wird automatisch mit 0 initialisiert.

Frage 15 (wahr/falsch)

3 Punkte

Kreuzen Sie jeweils an, ob die angegebene Aussage wahr oder falsch ist!

- Die Klasse NullPointerException ist eine Systemausnahme (Unchecked Exception).
- Die Exception IllegalArgumentException kann durch das Statement throw IllegalArgumentException(); geworfen werden.
- Durch die try-with-resources-Anweisung können Ressourcen geschlossen werden, welche das Interface AutoCloseable implementieren.

Frage 16 (wahr/falsch)

3 Punkte

Kreuzen Sie jeweils an, ob die angegebene Aussage wahr oder falsch ist!

- Gegeben sei Integer `myInteger = Integer.valueOf(51);`. Autounboxing ist dafür verantwortlich, dass der Referenzvergleich `51 == myInteger` den Wert `true` ergibt.
- Die Zuweisung `Double d = 100;` ist gültig.
- Die automatische Umwandlung von primitiven Werten in Wrapper-Objekte wird als Autoboxing bezeichnet.

Frage 17 (multiple choice)

1 Punkt

Kreuzen Sie alle Keywords an, welche in Java als Zugriffsmodifikatoren verwendet werden können!

- `static`
- `final`
- `private`
- `package`
- `protected`
- `default`
- `public`
- `class`

Frage 18 (wahr/falsch)

3 Punkte

Gegeben sei die Variable **myList**, welche den Typ **List<Integer>** hat. Kreuzen Sie jeweils an, ob die angegebene Aussage wahr oder falsch ist!

- Der Stream `myList.stream().filter(n -> n >= 0).reduce(Integer::sum).orElse(0)` berechnet die Summe aller nicht negativen Zahlen in der Liste **myList**. Die Summe einer leeren Liste bzw. einer Liste mit nur negativen Zahlen wird als 0 definiert.
- Der Stream `myList.stream().distinct().count()` ermittelt die Anzahl der Elemente in **myList**.
- Der Stream `myList.stream().map(x -> 2 * x).collect(Collectors.toList())` verdoppelt den Wert von allen Elementen und sammelt das Ergebnis in einer Liste.

Frage 19 (wahr/falsch)

3 Punkte

Kreuzen Sie jeweils an, ob die angegebene Aussage wahr oder falsch ist!

- Für eine korrekte Implementierung der Methode `compareTo` aus dem Interface `Comparable` muss neben weiteren Bedingungen gelten, dass `compareTo` transitiv ist.
- Zwei Objekte sind, laut der Methode `compare` aus dem Interface `Comparator` gleich, wenn der Rückgabewert kleiner als 0 ist.
- Das Interface `Comparator` kann nicht als Lambda-Ausdruck implementiert werden. Es ist kein Functional Interface.

Frage 20 (wahr/falsch)

3 Punkte

Kreuzen Sie jeweils an, ob die angegebene Aussage wahr oder falsch ist!

- Testen ist der Vorgang ein Programm oder einen Teil davon mit der Absicht auszuführen, möglicherweise enthaltene Fehler zu finden.
- Durch Testen ist für jedes nicht triviale Programm ein endgültiger Korrektheitsbeweis möglich.
- Bei Unit-Tests wird das Zusammenwirken von Systemkomponenten getestet.

Frage 21 (wahr/falsch)

3 Punkte

Gegeben sei die Klasse MyClass.

```
public class MyClass {  
    private double x;  
  
    public double bar(MyClass a) {  
        return this.x + a.x;  
    }  
}
```

- Java verwendet objektbasierte Sichtbarkeit. Daher sind Zugriffe auf `x` in der Methode `bar()` der Klasse `MyClass` gestattet.
- Die Klasse `MyClass` hat eine Superklasse.
- Die Klasse `MyClass` stellt keinen Konstruktor bereit. Daher können keine Instanzen aus dieser Klasse erstellt werden.

Frage 22 (wahr/falsch)

3 Punkte

Kreuzen Sie jeweils an, ob die angegebene Aussage wahr oder falsch ist!

- Eine statische innere Klasse hat Zugriff auf die Klassenvariablen der äußeren Klasse.
- Eine Elementklasse ist nur in der Methode gültig, in der sie deklariert wurde.
- Anonyme Klassen sind Elementklassen ohne Namen.

Frage 23 (wahr/falsch)

3 Punkte

Kreuzen Sie jeweils an, ob die angegebene Aussage über Java Arrays wahr oder falsch ist!

- Die öffentliche Objektvariable `length`, welche Arrays zur Verfügung stellen, gibt die Länge des Arrays an.
- In Arrays können primitive Datentypen oder Referenztypen verwaltet werden.
- Überschreitet der Indexwert beim Zugriff auf ein Array die Arraygrenzen, führt dies zu einem Kompilierfehler.

Frage 14 (multiple choice)

1 Punkt

Kreuzen Sie alle Keywords an mit denen in Java die Serialisierung von bestimmten Feldern verhindert werden kann!

- `final`
- `protected`
- `default`
- `volatile`
- `transient`
- `private`

Frage 25 (wahr/falsch)

3 Punkte

Kreuzen Sie jeweils an, ob die angegebene Aussage wahr oder falsch ist!

- Sets sind Collections, welche keine duplizierten Elemente enthalten.
- HashMaps verwalten Schlüssel mit dazugehörigen Werten. Jeder Wert innerhalb einer HashMap muss eindeutig sein.
- Der Indexzugriff über die Methode `get` einer LinkedList ist für jeden Index gleich schnell.

Frage 26 (wahr/falsch)

3 Punkte

Kreuzen Sie jeweils an, ob die angegebene Aussage wahr oder falsch ist!

- Bei der statischen Bindung wird der Methodenrumpf bereits zur Compilezeit dem Methodenaufruf zugeordnet.
- Datenelemente werden statisch gebunden.
- Aufrufe von objektbezogenen Methoden werden immer dynamisch gebunden.

Frage 27 (wahr/falsch)

Kreuzen Sie jeweils an, ob die angegebene Aussage wahr oder falsch ist!

- Typebounds werden zur Laufzeit verarbeitet.
- Polymorphe Methoden können in nicht-generischen Klassen definiert und aufgerufen werden.
- Bei kovarianten Wildcardtypen (Wildcardtypen mit Upper-Typebound) sind schreibende Zugriffe erlaubt.