

JAVA 06: Gemischte einfache Fragen zu Java

panitz

Zusammenfassung

In diesem Kurs geht es um Terminologie und Grundzüge der Programmiersprache Java. Dieses betrifft sowohl Aspekte der Programmierung als auch des Toolings.

Welche der folgenden Funktionen realisiert die logische Implikation:
($a \rightarrow b$)?

(Mehrere Antwortmöglichkeiten).

- ```
static boolean impl(boolean a,boolean b){
 return !a || b;
}
```
- ```
static boolean impl(boolean a,boolean b){  
    return !(a && !b);  
}
```
- ```
static boolean impl(boolean a,boolean b){
 return (a -> b);
}
```
- ```
static boolean impl(boolean a,boolean b){  
    return (a || !b);  
}
```
- ```
static boolean impl(boolean a,boolean b){
 if (a) return b;
 return true;
}
```
- ```
static boolean impl(boolean a,boolean b){  
    return a && b || !a;  
}
```

Korrekte Antworten

- ```
static boolean impl(boolean a,boolean b){
 return !a || b;
}
```
- ```
static boolean impl(boolean a,boolean b){  
    return !(a && !b);  
}
```
- ```
static boolean impl(boolean a,boolean b){
 if (a) return b;
 return true;
}
```
- ```
static boolean impl(boolean a,boolean b){  
    return a && b || !a;  
}
```

Erläuterung

Diese vier Lösungen realisieren die logische Implikation. Für die logischen Ausdrücke kann man dieses mit Wahrheitstafeln schnell zeigen.

Für die Lösung mit der if-Bedingung muss man den Kontrollfluss für alle Fälle nachvollziehen.

-> ist kein Operator in Java, sondern syntaktischer Teil der Lambda-Ausdrücke.

Frage: Syntax in Java

Mit welchem Symbol wird auf Felder und auf Methoden eines Objektes zugegriffen?

- .

Erläuterung

Felder und Methoden, die nicht statisch sind, gehören immer zu einem konkreten Objekt. Um auf ein Feld oder eine Methode zuzugreifen, schreibt man erst einen Ausdruck, der das Objekt bezeichnet. Dann folgt das Punktsymbol und schließlich der Name des Feldes oder der Methode.

Frage: Terminologie

Wie nennt man (im Singular) einen Ausdruck, der direkt eine Zahl, einen Wahrheitswert, ein Zeichen oder ein Objekt der Klasse `String` bezeichnet?

- Literal

Erläuterung

Literale sind dazu da, um Werte direkt im Quelltext aufzuschreiben. Es gibt

- Literale für Zahlen: `1`, `42`, `42.42`, `42L`, `42E17`
- die beiden Literale `true` und `false` für Wahrheitswerte
- Literale in einfachen Anführungszeichen für Zeichen `'ß'`
- Literale in doppelten Anführungszeichen für Stringobjekte: `"hello world"`.

Frage: Programmfluss

Gegeben sei folgende fehlerhafte Methode:

```
static boolean f(long l){  
    if (l>0) return true;  
    if (l<0) return false;  
}
```

Mit welchem Schlüsselwort beginnt die Anweisung, die man mindestens noch benötigt, damit die Methode kompiliert?

- `return`

Erläuterung

Eine Methode, die nicht als `void` deklariert ist, muss immer ein Ergebnis zurück geben. Dieses geschieht mit der `return`-Anweisung.

Der Programmfluss muss sicherstellen, dass auf jedem Fall irgendwann eine `return`-Anweisung ausgeführt wird.

In diesem Beispiel wird kein Wert zurückgegeben, wenn beide `if`-Bedingungen falsch sind, wenn `1` also `0` ist.

Der Compiler überprüft, ob garantiert ist, dass eine `return`-Anweisung erreicht werden kann.

Frage: Rekursion nachvollziehen

Gegeben sei folgende Methode:

```
static long f(long l){  
    return l==0 ? 0 : (l%10 + f(l/10));  
}
```

Was ergibt der Ausdruck $f(9159)$?

- 24

Erläuterung

Die Funktion berechnet rekursiv die Quersumme. Hierzu wird mit dem Modulooperator letzte Ziffer berechnet. Mit der ganzzahligen Division die Zahl ohne die letzte Ziffer errechnet. Dafür erfolgt der rekursive Aufruf.

Frage: Primitive Datentypen

Wieviel Bits belegt eine Zahl des Datentyps `int` in Java im Speicher?

- 32

Erläuterung

Ein `long` belegt 8 Bytes also 64 Bits.

Ein `int` belegt 4 Bytes also 32 Bits.

Ein `short` belegt 2 Bytes also 16 Bits.

Ein `byte` entsprechend 1 Byte also 8 Bits.

Ein `char` belegt 2 Bytes also 16 Bits.

Frage: Primitive Datentypen

Welches sind primitive Typen der Sprache Java?

(Mehrere Antwortmöglichkeiten).

- long
- byte
- bool
- char
- String
- main
- public
- void

Korrekte Antworten

- long
- byte
- char

Erläuterung

Java hat genau 8 primitive Typen. Diese werden klein geschrieben. Alles andere sind Objekttypen. Insbesondere `String` ist eine Klasse und kein primitiver Typ.

Der primitive Typ für Wahrheitswerte heißt `boolean`.

`void` ist kein Typ, sondern ein Schlüsselwort, das anzeigt, dass eine Methode keinen Rückgabewert hat.

Frage: Konstruktoren in Java

Gegeben sei die Klasse:

```
class IntBox{ int i;}
```

Welche der folgenden Konstruktoren sind korrekt und sinnvoll?

(Mehrere Antwortmöglichkeiten).

- `IntBox(int i){ this.i = i;}`
- `IntBox(int i){ i = i;}`
- `IntBox(int i){ this.i = this.i;}`
- `IntBox(int i){ i = this.i;}`
- `IntBox(int ip){ i = ip;}`
- `IntBox(){ i = 1;}`
- `IntBox(){ i = "1";}`

Korrekte Antworten

- `IntBox(int i){ this.i = i;}`
- `IntBox(int ip){ i = ip;}`
- `IntBox(){ i = 1;}`

Erläuterung

In den Konstruktoren werden die Felder des neu erzeugten Objektes mit Werten belegt. Zumeist erhält ein Konstruktor diese Werte als Parameter. Hat ein Parameter denselben Namen wie ein Feld, dann kann das auf Feld von dem Objekt mit dem Schlüsselwort `this` zugegriffen werden.

Statt mit einem im Konstruktor als Parameter übergebenen Wert kann ein Feld auch sinnvoll mit einem Standardwert initialisiert werden.

Frage: Schleifen

Welche Schleife ist inhaltlich gleich mit

```
for (int i=234; i>17;i = i-3){  
    System.out.println(i);  
}
```

(Eine Antwortmöglichkeit.)

- ```
int i = 234;
while (true){
 if (i<=17) break;
 System.out.println(i);
 i = i-3;
}
```

- ```
int i = 234;  
do{  
    i = i -3;  
    System.out.println(i);  
}while(i>17)
```

```
▪ int i = 234;
  while (true){
    if (i<=17) break;
    System.out.println(i);
    i = i-3;
  }
```

Erläuterung

Die for-Schleife ist eine vorgeprüfte Schleife.

Die break-Anweisung verlässt die Schleife komplett, deshalb ist sie in diesem Beispiel das Negat der Schleifenbedingung.

Frage: Terminologie der Objektorientierung

Welche Komponenten beinhaltet eine Klasse. (Antwortwörter im Plural)

- Felder
- Methoden
- Konstruktoren

Erläuterung

In den Feldern stehen die Eigenschaften, die ein Objekt der Klasse beschreiben. Sie werden auch als Attribute oder Exemplarvariablen bezeichnet.

Die Methoden rechnen mit den Daten eines Objekts.

Ein Konstruktor wird aufgerufen, wenn ein neues Objekt mit `new` erzeugt wird.

Frage: Typsysteme

Was für ein Typsystem hat Java primär?

(Eine Antwortmöglichkeit.)

- Statisches Typsystem
- Dynamisches Typsystem
- Ungetypt
- Typinferenzsystem

- Statisches Typsystem

Erläuterung

Statisch heißt, dass die Typen beim Kompilieren und nicht erst zur Laufzeit geprüft werden. Lokal macht der Javacompiler auch Typinferenz, z.B. wenn eine lokale Variable mit dem Schlüsselwort `var` und nicht mit einer Typangabe deklariert wird. Der Programmierer kann mit einer Typzusicherung auch eine dynamische Typüberprüfung von Objekten erreichen.

Frage: Ausführung von Javaprogrammen

Wie sieht der korrekte Aufruf aus der Kommandozeile der virtuellen Maschine zur Ausführung einer Klasse HelloWorld aus?

(Eine Antwortmöglichkeit.)

- java HelloWorld
- java HelloWorld.class
- javac HelloWorld.java
- java HelloWorld.java
- java helloworld

- java HelloWorld

Erläuterung

Der Interpreter von Java-Byte-Code Klassen heißt `java`. Er bekommt als Argument beim Aufruf den Namen der Klasse, die die auszuführende Methode `main` enthält. Es wird nur der Klassenname ohne die Dateiendung `.class` oder gar `.java` angegeben. Groß- und Kleinschreibung ist bei dem angegebenen Klassennamen relevant.

Frage: Syntax von Java

Markieren Sie die reservierten Schlüsselwörter der Sprache Java in folgender Aufzählung

(Mehrere Antwortmöglichkeiten).

- class
- main
- String
- if
- constructor
- java
- static
- false

Korrekte Antworten

- class
- if
- static
- false

Erläuterung

Schlüsselwörter sind Wörter die nicht als Bezeichner verwendet werden können. »String« ist ein Bezeichner für eine Standardklasse. »main« ist ein Bezeichner, der für den Namen der Einstiegsmethoden einer Klasse zu nehmen ist.

Frage: Ausdrücke und Anweisungen

Markieren Sie alle Ausdrücke in folgender Aufzählung.

(Mehrere Antwortmöglichkeiten).

- 42
- class X int a;
- if (x==y) 1 else 2
- 17== f() ? 1 : 2
- System.out.println("Hallo")
- while (true) f(14,7,2)
- 1+1*0+x >= 17 || x < f()
- false
- new Person("Helmut",SSchmidt")

Korrekte Antworten

- 42
- `17 == f() ? 1 : 2`
- `1 + 1 * 0 + x >= 17 || x < f()`
- `false`
- `new Person("Helmut", "SSchmidt")`

Erläuterung

Ausdrücke können immer zu einem Wert ausgewertet werden. Als Faustregel kann man sagen: alles, was man als Parameter der Methode `System.out.println()` übergeben kann, ist ein Ausdruck.

Keine Ausdrücke sind:

- Bedingungen mit `if`
- Schleifen und andere Anweisungen für den Kontrollfluss
- Deklarationen
- Aufruf von `void` Methoden.

Frage: Auswertung von Ausdrücken

Was ergibt folgender Ausdruck für einen Wert?

$050 + 2$

- 42

Erläuterung

Integer Literale, die mit einer 0 beginnen, sind im Quelltext im Oktalsystem notiert. 050 bedeutet also $5 \cdot 8 + 0 \cdot 1$.

Welche Ausgabe macht folgendes Programm auf der Kommandozeile?

```
class F{
    public static void main(String[] args){
        int x = 42;
        int result = 0;
        if (x<0);
            if (x<=42) result = 5;
            else result = 42;
        System.out.println(result);
    }
}
```

(Eine Antwortmöglichkeit.)

- 5
- 42
- 0

- 5

Erläuterung

Eine gemeine Falle: das Semikolon nach der If-Bedingung steht für eine leere Anweisung. Die Einrückung in der nächsten Zeile ist irreführend.

Welche Aussagen sind für die `switch`-Anweisung korrekt?

(Mehrere Antwortmöglichkeiten).

- Es kann eine Fallunterscheidung auf `String`-Objekte durchgeführt werden.
- Es muss immer einen `default`-Fall geben.
- Es kann ein Ausdruck mit einem beliebigen anderen Ausdruck verglichen werden.
- Es kann eine Fallunterscheidung auf Zahlenkonstanten durchgeführt werden.
- Es realisiert ein allgemeines Pattern-Matching.
- Es wird die Methode `equals` für die Fallunterscheidung verwendet.
- Es kann eine Fallunterscheidung auf Aufzählungsobjekte durchgeführt werden.

Korrekte Antworten

- Es kann eine Fallunterscheidung auf `String`-Objekte durchgeführt werden.
- Es kann eine Fallunterscheidung auf Zahlenkonstanten durchgeführt werden.
- Es kann eine Fallunterscheidung auf Aufzählungsobjekte durchgeführt werden-

Erläuterung

Die `switch-case`-Anweisung erlaubt einen Sprung zu einem konstanten Wert, der eine Zahl, ein `String` oder ein Aufzählungsobjekt sein kann.

Frage: Methoden

Mit welchem Schlüsselwort werden Methoden markiert, die auch aufgerufen werden können, ohne ein Objekt der Klasse zu referenzieren?

- static

Erläuterung

Statische Methoden entsprechen klassischen Prozeduren oder Funktionen, die nicht an ein Objekt gebunden sind.

Frage: main-Methode

Welche Aussagen sind für die `main`-Methode korrekt?

(Mehrere Antwortmöglichkeiten).

- Eine Klasse, die ausgeführt werden kann, muss eine `main`-Methode haben oder von `javafx.application.Application` ableiten.
- Es darf nur eine `main`-Methode im gesamten Projekt geben.
- Der Parameter der `main`-Methode muss `args` heißen.
- Die `main`-Methode darf nirgends aufgerufen werden. Sie kann nur von der virtuellen Methode zum Start des Programms aufgerufen werden.
- Die `main`-Methode darf nicht überladen werden.
- Eine `main`-Methode die zum Start des Programms verwendet wird, muss statisch sein.

Korrekte Antworten

- Eine Klasse, die ausgeführt werden kann, muss eine `main`-Methode haben oder von `javafx.application.Application` ableiten.
- Eine `main`-Methode die zum Start des Programms verwendet wird, muss statisch sein.

Erläuterung

Es können beliebig viele Methoden `main` heißen und unterschiedlich definiert sein, nur die Klasse mit der das Programm gestartet wird mit eine Methode folgender Signatur haben:
`public static void main(String[] args)`

Frage: four for loops

Welche der folgenden Anweisungen iterieren über die Zahlen 1 bis 10 und geben diese auf der Kommandozeile aus?

(Mehrere Antwortmöglichkeiten).

```
▪ int[] xs = {1,2,3,4,5,6,7,8,9,10};  
  for (int x:xs) System.out.println(x);
```

```
▪ int i1 = 1;  
  for(;i1<11;){ System.out.println(i1);  
    i1 += 1;  
  }
```

```
▪ for(int i2=0;i2<=10;i2++) System.out.println(i2);
```

```
▪ int i3 = 1;  
  for(;true;){ System.out.println(i3);  
    i3++;  
    if (i3>10) break;  
  }
```



```
▪ int[] xs = {1,2,3,4,5,6,7,8,9,10};  
  for (int x:xs) System.out.println(x);
```

```
▪ int i1 = 1;  
  for(;;i1<11;){ System.out.println(i1);  
    i1 += 1;  
  }
```

```
▪ int i3 = 1;  
  for(;;true;){ System.out.println(i3);  
    i3++;  
    if (i3>10) break;  
  }
```

Erläuterung

Es gibt viele Arten Schleifen zu schreiben. Man sollte immer darauf aufpassen, dass man nicht einen Offset by One Fehler macht.

Frage: Schleifen

Welche der folgenden Anweisungen innerhalb einer Schleife sorgen dafür, dass die Schleife verlassen wird?

(Mehrere Antwortmöglichkeiten).

- `break`
- `return`
- `continue`
- `throw`

Korrekte Antworten

- break
- return
- throw

Erläuterung

continue verlässt nicht die Schleife, sondern springt zum nächsten Schleifendurchgang.

Frage: Variable Parameterliste

Welche bezüglich einer variablen Parameterlänge durch die Notation ... sind korrekt?

(Mehrere Antwortmöglichkeiten).

- Sie kann nur für primitive Typen verwendet werden.
- Wird sie verwendet, kann es keinen weiteren Parameter mehr geben.
- Es werden die hierfür übergebenen Parameter als ein Array übergeben.
- Es kann nur der letzte Parameter einer Methode eine variable Parameterlänge haben.
- Es kann nur der erste Parameter einer Methode eine variable Parameterlänge haben.

Korrekte Antworten

- Es werden die hierfür übergebenen Parameter als ein Array übergeben.
- Es kann nur der letzte Parameter einer Methode eine variable Parameterlänge haben.

Erläuterung

Der letzte Parameter einer Parameterliste kann eine variable Anzahl haben.

Frage: Attribute für Methoden

Welches sind **keine** Attribute, die für Methoden gesetzt sein können.

(Mehrere Antwortmöglichkeiten).

- `virtual`
- `static`
- `abstract`
- `implicit`
- `native`

Korrekte Antworten

- `virtual`
- `implicit`

Erläuterung

`virtual` und `implicit` mögen aus anderen Sprachen bekannt sein, existieren aber nicht in Java.

Frage: final

Was kann in Java alles `final` sein?

(Mehrere Antwortmöglichkeiten).

- Schnittstellen
- Klassen
- lokale Variablen
- Methoden
- Felder

Korrekte Antworten

- Klassen
- lokale Variablen
- Methoden
- Felder

Erläuterung

Bei Schnittstellen hat `final` keinen Sinn und wird deshalb vom Compiler mit einer Fehlermeldung zurück gewiesen.

Frage: Disassembler

Das Java Development Kit beinhaltet auch einen Disassembler. Wie heißt dieser (Befehl auf der Kommandozeile)?

Korrekte Antworten

- javap

Erläuterung

Mit `javap` kann man sich Class-Dateien auf ihren Inhalt anzeigen lassen.