

Einstieg in Java und OOP

Christian Silberbauer

Übungsblatt 4

Folgende Theoriefragen zur Objektorientierten Programmierung sind in Bezug auf die Programmiersprache Java zu beantworten. Wenn Sie Fragen nicht unmittelbar beantworten können, versuchen Sie sie bitte anhand der Vorlesungsfolien oder anhand kurzer Testprogramme zu beantworten. Die Aufgaben können während der Übungsstunden in kleinen Gruppen (2 – 3 Personen) gelöst werden oder gerne auch in ggf. größeren Lerngruppen vorbereitet werden.

Aufgabe 1

Richtig oder falsch?

- a) Die Garbage Collection gibt den ungenutzten Speicher im Stack frei.
- b) Implizite Typumwandlungen sind immer typsicher.
- c) Lokale Variablen werden ggf. implizit mit einem Standardwert initialisiert.
- d) Ob eine Methode sichtbar ist, wird statisch entschieden.
- e) Mehrfachvererbung bei Klassen ist erlaubt.
- f) Eine Unterklasse kann auch weniger sichtbare Methoden als ihre Oberklasse haben.
- g) Innere Klassen können lediglich von ihrer umschließenden Klasse instanziiert werden.
- h) Anonyme, innere Klassen haben maximal eine Instanz.
- i) Innere Klassen haben Zugriff auf die privaten Attribute ihrer umschließenden Klasse.
- j) Eine Klasse kann mehrere Interfaces implementieren.
- k) Ein Interface kann keine Variablendefinition beinhalten.
- l) Zwei Methoden mit gleichem Namen und gleichen Parametern können in einer Klasse nicht definiert werden, wenn sich ausschließlich der Rückgabetyt unterscheidet (Stichwort: Überladen von Methoden).
- m) Eine Methode kann überschrieben werden, wobei sich bei der überschreibenden Methode lediglich der Rückgabetyt unterscheidet.

Aufgabe 2

Warum werden Attribute ggf. mit einem Standardwert initialisiert?

Aufgabe 3

Können mehrere Variablen auf dasselbe String-Objekt verweisen? Begründen Sie Ihre Antwort und berücksichtigen Sie dabei die Aussage aus der Vorlesung, dass sich String-Objekte nach außen hin wie Instanzen von Wertetypen verhalten.

Aufgabe 4

Konstruktoren werden der Vererbungshierarchie entlang von oben nach unten aufgerufen. Warum?

Aufgabe 5

Wann kann die Garbage Collection den Speicher eines Objekts freigeben?

Aufgabe 6

Testen Sie, ob Java es erlaubt, eine von der Oberklasse überschriebene Methode aufzurufen. Warum ist das Verhalten von Java sinnvoll?

Aufgabe 7

Warum können abstrakte Klassen mit abstrakten Methoden nicht unmittelbar instanziiert werden?

Aufgabe 8

Können Konstruktoren polymorph sein, also zur Laufzeit unterschiedlich gebunden werden? Begründen Sie Ihre Antwort.

Aufgabe 9

Folgende Klassen seien gegeben:

```
class A {
    public void f() { System.out.println("A.f()"); }
    public void h() { System.out.println("A.h()"); }
}

class B extends A {
    public void f() { System.out.println("B.f()"); }
    public void g() { System.out.println("B.g()"); }
}

class C extends B {
    public void g() { System.out.println("C.g()"); }
}
```

a) Zu welcher Ausgabe auf der Konsole führen folgende Anweisungen?

```
A a = new A();
a.f();
a.g();
B b = new B();
```

```
b.f();
b.g();
C c = new C();
c.f();
c.g();
```

b) Zu welcher Ausgabe auf der Konsole führen folgende Anweisungen?

```
A x = new B();
x.f();
x.g();
B y = (B)x;
y.g();
B z = new C();
z.f();
z.g();
z.h();
```

- c) Welche Anweisung müssten Sie in C.g() hinzufügen, um B.f() aufzurufen?
- d) Welche Anweisung müssten Sie in C.g() hinzufügen, um B.g() aufzurufen?
- e) Welche Anweisung müssten Sie in C.g() hinzufügen, um A.h() aufzurufen?
- f) Welche Anweisung müssten Sie in C.g() hinzufügen, um A.f() aufzurufen?

Aufgabe 10

Folgende Typen seien gegeben:

```
interface I {
    void f();
}

class A {
    public void f() { System.out.println("A.f()"); }
}

class B extends A implements I {
}
```

Zu welcher Ausgabe auf der Konsole führen folgende Anweisungen?

```
I x = new B();
x.f();
```

Aufgabe 11

Erklären Sie, wie es die Objektorientierte Programmierung erlaubt, dass Objekte typischer kommunizieren können, ohne dass sich deren Klassen referenzieren. Und erklären Sie warum das gut ist.

Aufgabe 12

Begründen Sie, warum der Compiler beim Übersetzen den Ausdruck „DISPOSE_ON_CLOSE“ in der Kundenverwaltung v1 der Vorlesungsfolien akzeptiert.

