DYS+ Campus

01

Übungsheft · Informatik · Klasse 11

Klassen implementieren und dokumentieren

Erklärungen, 32 Aufgaben in drei Stufen und alle Lösungen, viele mit Begründung.

Name _____ **Datum** _____

Das lernst du

- Ich kann eine Klassenkarte in Java übertragen.
- Ich kann Konstruktor, Methoden und Referenzen lesen.
- Ich kann Methoden kommentieren und vorgegebene Klassen nutzen.

So arbeitest du mit dem Heft

●○○ **Basis** zum Einstieg

●●○ **Standard** zum Üben

●●● **Plus** zum Weiterdenken

Du musst nicht alles machen. Fang bei Basis an. Die Lösungen stehen hinten im Heft.

Noch unsicher?

Scanne den Code. Dort wird das Thema Schritt für Schritt erklärt.

dysplus.de/lernen/gymnasium/klasse-11/informatik/klassen-implementieren/eine-klasse-im-quelltext

Kurz erklärt

Merksatz: Minus wird private, Plus wird public. new ruft den Konstruktor auf. Der Konstruktor heißt wie die Klasse und hat keinen Rückgabetyp.

Diagramm	Java
– farbe: String	`private String farbe;`
+ istFrei(): boolean	`public boolean istFrei()`
Vererbung	`class Buch extends Medium`

Verweise nachverfolgen

1. a = new Fahrrad(1); b = new Fahrrad(2); c = a;
2. Zwei Objekte, drei Variablen. c und a zeigen auf dasselbe Objekt.

Typischer Fehler: Eine Zuweisung wie c = a; kopiert nur den Verweis, kein Objekt. Methoden dürfen nur auf Objekten aufgerufen werden, die mit new erzeugt wurden.

●○○ Basis

Übertragen und verfolgen

1 Übertrage in Java.

a) – farbe: String wird zu _____ String farbe; ☐ **private** ☐ **public**

b) + ausleihen(): void wird zu _____ void ausleihen() ☐ **private** ☐ **public**

c) Der Konstruktor heißt wie die _____. ☐ **Klasse** ☐ **Methode**

d) Ein Objekt erzeugt man mit dem Schlüsselwort _____. ☐ **this** ☐ **new**

2 Die Klasse Zaehler: Konstruktor setzt den Wert 0, erhoehe() addiert 1, zuruecksetzen() setzt auf 0. Verfolge den Wert.

a) nach new Zaehler(): _____

b) nach erhoehe(): _____

c) nach erhoehe(): _____

d) nach zuruecksetzen(): _____

e) nach erhoehe(): _____

3 Fahrrad a = new Fahrrad(1); Fahrrad b = new Fahrrad(2); Fahrrad c = b;

a) Anzahl der Objekte: _____

b) Anzahl der Variablen: _____

c) c verweist auf das Objekt mit der Nummer _____

●●○ Standard

Rechnen, schreiben, begründen

4 **Ein Fahrrad f1 kostet 5 Euro pro Tag, f2 kostet 4 Euro pro Tag. Rechne in ganzen Euro.**

a) f1.berechnePreis(3) = _____

b) f2.berechnePreis(4) = _____

c) f1.berechnePreis(2) + f2.berechnePreis(5) = _____

5 **Schreibe auf.**

a) Schreibe die Klasse Konto in Java mit dem privaten Attribut stand (int) und der Methode getStand().

b) Schreibe einen Dokumentationskommentar für die Methode istFrei(), die true liefert, wenn das Fahrrad frei ist.

c) Der Konstruktor Fahrrad verlangt einen int-Parameter. Was passiert bei new Fahrrad()?

d) Was bewirkt super(titel) im Konstruktor einer Unterklasse?

6 Stimmt der Satz? Schreibe falsche Sätze richtig auf.

a) Ein Konstruktor hat einen Rückgabetyp. ☐ **stimmt** ☐ **stimmt nicht**

b) Mit c = a; wird ein neues Objekt erzeugt. ☐ **stimmt** ☐ **stimmt nicht**

c) Ein Dokumentationskommentar beschreibt Zweck, Parameter und Rückgabe. ☐ **stimmt** ☐ **stimmt nicht**

d) Ein Kommentar verändert das Ergebnis des Programms. ☐ **stimmt** ☐ **stimmt nicht**

e) Vererbung wird in Java mit extends geschrieben. ☐ **stimmt** ☐ **stimmt nicht**

●●● Plus

Beziehungen im Code

7 **Sequenzdiagramm als Tabelle: Schritt 1 kunde ruft verleih.ausleihen(7). Schritt 2 verleih ruft fahrrad7.istFrei(), Rückgabe true. Schritt 3 verleih ruft fahrrad7.ausleihen(). Schritt 4 verleih meldet true an kunde.**

a) Sender von Schritt 2: _____

b) Empfänger von Schritt 3: _____

c) Anzahl der Methodenaufrufe: _____

d) Wenn istFrei den Wert false liefert, entfällt Schritt _____

8 **Medium hat das Attribut titel. Buch erbt von Medium und ergänzt seiten und autor.**

a) Attribute eines Buch-Objekts: _____

b) Argumente für einen Konstruktor Buch(String titel, int seiten), der autor noch nicht setzt: _____

9 **Schreibe auf.**

a) Schreibe die Klasse Ausleihe mit den Attributen rad (Fahrrad) und kunde (String) und einem Konstruktor, der beide setzt.

b) Warum kann eine Unterklasse ein privates Attribut der Oberklasse nicht direkt lesen?

Kannst du das schon?

Kreuze an, wie sicher du dich fühlst. Hier gibt es kein Richtig oder Falsch.

Ich kann …	sicher	fast	noch nicht
Ich kann eine Klassenkarte in Java übertragen.	☐	☐	☐
Ich kann Konstruktor, Methoden und Referenzen lesen.	☐	☐	☐
Ich kann Methoden kommentieren und vorgegebene Klassen nutzen.	☐	☐	☐

Tipp: Bei „noch nicht“ schau dir „Kurz erklärt“ vorne im Heft noch einmal an oder scanne den Code auf dem Deckblatt. Dann probierst du die Aufgaben noch einmal.

Lösungen

Erst selbst lösen, dann vergleichen. Bei vielen Aufgaben steht die Begründung dabei.

●○○ Basis: Übertragen und verfolgen

1 Übertrage in Java.

a) private

→ Minus bedeutet privat.

b) public

→ Plus bedeutet öffentlich.

c) Klasse

→ Gleicher Name wie die Klasse.

d) new

→ new ruft den Konstruktor auf.

2 Die Klasse Zaehler: Konstruktor setzt den Wert 0, erhoehe() addiert 1, zuruecksetzen() setzt auf 0. Verfolge den Wert.

a) nach new Zaehler(): 0

→ Startwert.

b) nach erhoehe(): 1

→ 0 + 1.

c) nach erhoehe(): 2

→ 1 + 1.

d) nach zuruecksetzen(): 0

→ Zurück auf 0.

e) nach erhoehe(): 1

→ 0 + 1.

3 Fahrrad a = new Fahrrad(1); Fahrrad b = new Fahrrad(2); Fahrrad c = b;

a) Anzahl der Objekte: 2

→ Zwei new.

b) Anzahl der Variablen: 3

→ a, b und c.

c) c verweist auf das Objekt mit der Nummer 2

→ c = b.

●●○ Standard: Rechnen, schreiben, begründen

4 Ein Fahrrad f1 kostet 5 Euro pro Tag, f2 kostet 4 Euro pro Tag. Rechne in ganzen Euro.

a) f1.berechnePreis(3) = 15

→ 3 · 5.

b) f2.berechnePreis(4) = 16

→ 4 · 4.

c) f1.berechnePreis(2) + f2.berechnePreis(5) = 30

→ 10 + 20.

5 Schreibe auf.

a) `public class Konto { private int stand; public int getStand() { return stand; } }`

→ Attribut privat, get-Methode öffentlich.

b) **Gibt an, ob das Fahrrad frei ist. @return true, wenn das Fahrrad frei ist, sonst false.**

→ Zweck und Rückgabe.

c) **Die Übersetzung meldet einen Fehler, weil das Argument für den Parameter fehlt. Er wird also vor dem Start gemeldet, wie ein Syntaxfehler, nicht erst beim Ablauf.**

→ Anzahl der Argumente.

d) **Es ruft den Konstruktor der Oberklasse auf und übergibt den Titel.**

→ Oberklasse zuerst.

6 Stimmt der Satz? Schreibe falsche Sätze richtig auf.

a) **Stimmt nicht. Richtig: Ein Konstruktor hat keinen Rückgabetyp.**

b) **Stimmt nicht. Richtig: Es wird nur der Verweis kopiert. Es gibt weiter nur ein Objekt.**

c) **Stimmt.**

d) **Stimmt nicht. Richtig: Kommentare werden vom Computer ignoriert.**

e) **Stimmt.**

●●● Plus: Beziehungen im Code

7 Sequenzdiagramm als Tabelle: Schritt 1 kunde ruft verleih.ausleihen(7). Schritt 2 verleih ruft fahrrad7.istFrei(), Rückgabe true. Schritt 3 verleih ruft fahrrad7.ausleihen(). Schritt 4 verleih meldet true an kunde.

a) **Sender von Schritt 2: verleih**

→ Der Verleih fragt das Fahrrad.

b) **Empfänger von Schritt 3: fahrrad7**

→ Der Verleih bucht das Fahrrad.

c) **Anzahl der Methodenaufrufe: 3**

→ Schritte 1 bis 3.

d) **Wenn istFrei den Wert false liefert, entfällt Schritt 3**

→ Dann wird nicht ausgeliehen.

8 Medium hat das Attribut titel. Buch erbt von Medium und ergänzt seiten und autor.

a) **Attribute eines Buch-Objekts: 3**

→ 1 geerbtes + 2 eigene.

b) **Argumente für einen Konstruktor Buch(String titel, int seiten), der autor noch nicht setzt: 2**

→ Zwei Parameter.

9 Schreibe auf.

a) `public class Ausleihe { private Fahrrad rad; private String kunde; public Ausleihe(Fahrrad rad, String kunde) { this.rad = rad; this.kunde = kunde; } }`

→ Assoziation als Attribut.

b) **Privat bedeutet: nur die eigene Klasse darf zugreifen. Die Unterklasse nutzt eine öffentliche Methode der Oberklasse, zum Beispiel getTitel().**

→ Kapselung.