

Probeklausur – Objektorientierte Programmierung

Bearbeitungszeit: 60 Minuten · Erreichbare Punkte: 54 · Hilfsmittel: keine

Name: Matrikelnr.:

Aufgabe	1	2	3	4	5	Σ
max. Punkte	12	4	12	12	14	54
erreicht						

Aufgabe 1 – Theorie & Fehlersuche (12 Punkte)

a) Multiple Choice (6 P.). Kreuzen Sie jeweils genau eine richtige Antwort an.

1. Bringen Sie die Ganzzahl-Datentypen nach ihrer Größe (Speicherbedarf) in die richtige Reihenfolge – vom kleinsten zum größten:

- ☐ a) byte < short < int < long
- ☐ b) byte < int < short < long
- ☐ c) short < byte < int < long
- ☐ d) int < long < short < byte

2. Was bewirkt das Schlüsselwort `private` bei einem Attribut?

- ☐ a) Zugriff von überall im Programm
- ☐ b) Zugriff nur innerhalb derselben Klasse
- ☐ c) Zugriff nur in Unterklassen
- ☐ d) Es macht das Attribut unveränderlich

3. Wozu dient `this()` als erste Anweisung in einem Konstruktor?

- ☐ a) Es ruft den Konstruktor der Oberklasse auf
- ☐ b) Es ruft einen anderen Konstruktor derselben Klasse auf
- ☐ c) Es erzeugt ein neues Objekt
- ☐ d) Es gibt die aktuelle Instanz zurück

4. Womit ruft eine Unterklasse den Konstruktor ihrer Oberklasse auf?

- ☐ a) `this()`
- ☐ b) `super()`
- ☐ c) `base()`
- ☐ d) `parent()`

5. Was beschreibt Polymorphie am besten?

- ☐ a) Eine Methode kann mehrere Parameter haben
- ☐ b) Eine Variable des Obertyps kann auf Objekte verschiedener Unterklassen verweisen; der Aufruf richtet sich nach dem tatsächlichen Objekt
- ☐ c) Eine Klasse kann mehrere Attribute besitzen
- ☐ d) Ein Attribut kann Werte verschiedener Typen speichern

6. Eine Klasse, die mindestens eine abstrakte Methode enthält, ...

- ☐ a) kann direkt mit `new` instanziiert werden
- ☐ b) muss selbst als `abstract` deklariert sein
- ☐ c) darf keine Attribute besitzen

- d) erbt automatisch von allen Interfaces

b) Fehlersuche (6 P.). Der folgende Code enthält **genau 3 Fehler**. Benennen Sie sie und geben Sie die Korrektur an.

```
class Auto {
    private String marke;
    public Auto(String marke) {
        marke = marke;
    }
    public string getMarke() {
        return marke;
    }
}
class Test {
    public static void main(String[] s) {
        Auto a = Auto("VW");
        System.out.println(a.getMarke());
    }
}
```

Fehler 1 / 2 / 3 mit Korrektur:

Aufgabe 2 – Ausgabe vorhersagen (4 Punkte)

Was gibt der folgende Code-Ausschnitt auf der Konsole aus?

```
int[] a = {2, 4, 6, 8};
int s = 0;
for (int i = 0; i < a.length; i++) {
    if (a[i] % 4 == 0) s += a[i];
    else s -= a[i];
}
System.out.println(s);
```

Ausgabe:

Aufgabe 3 – Code-Lücken ausfüllen (12 Punkte)

Ergänzen Sie die 6 Lücken ____ (n) ____, sodass eine lauffähige Klasse entsteht (je 2 P.).

```

class Rechteck {
    private double laenge;
    private double breite;
    private static int anzahl = 0;        // zaehlt erzeugte Rechtecke

    public Rechteck(double laenge, double breite) {
        ____ (1) ____ .laenge = laenge;
        this.breite = ____ (2) ____;
        anzahl ____ (3) ____;
    }
    public double flaeche() {
        return ____ (4) ____;
    }
    public static int getAnzahl() {
        return ____ (5) ____;
    }
    public static void main(String[] s) {
        Rechteck r = ____ (6) ____ Rechteck(3.0, 4.0);
        System.out.println("Flaeche: " + r.flaeche());
        System.out.println("Anzahl: " + Rechteck.getAnzahl());
    }
}

```

Ergänzungen:

- (1)
- (2)
- (3)
- (4)
- (5)
- (6)

Aufgabe 4 – Klasse mit Methoden schreiben (12 Punkte)

Schreiben Sie eine Klasse `TextTools` mit den beiden folgenden **Instanzmethoden (ohne static)**. Ergänzen Sie außerdem eine Klassenmethode `main`, die ein Objekt der Klasse erzeugt und beide Methoden testet.

a) (6 P.) Eine Methode `zaehleGrossbuchstaben(String text)`, die zurückgibt, wie viele Großbuchstaben (A–Z) in `text` vorkommen. Beispiel: "Hallo Welt" → 2.

b) (6 P.) Eine Methode, die den größten Wert eines **Integer-Arrays** (Array von ganzen Zahlen) zurückgibt (das Array ist nicht leer). Wählen Sie Rückgabebetyp, Methodennamen und Parameter selbst passend.

Platz für Ihre Lösung:

Aufgabe 5 – Klasse mit Vererbung (14 Punkte)

Gegeben ist die Basisklasse Fahrzeug:

```
class Fahrzeug {  
    protected String name;  
    public Fahrzeug(String name) { this.name = name; }  
    public void beschreibung() {  
        System.out.println("Fahrzeug: " + name);  
    }  
}
```

a) (9 P.) Schreiben Sie eine Unterklasse Auto, die von Fahrzeug erbt, ein zusätzliches `int`-Attribut `anzahlTueren` besitzt, einen passenden Konstruktor hat (Name über `super`, Türen setzen) und `beschreibung()` so überschreibt, dass zusätzlich die Türanzahl ausgegeben wird.

b) (5 P.) Schreiben Sie eine `main`-Methode, die ein `Fahrzeug[]`-Array mit einem Fahrzeug und einem Auto füllt und in einer Schleife für jedes Element `beschreibung()` aufruft (Polymorphie).

Platz für Ihre Lösung: