Karlsruher Institut für Technologie

-
-
https://sdqweb.ipd.kit.edu/wiki/Programmieren
programmieren-vorlesung@cs.kit.edu

Programmieren – Wintersemester 2022/23

Präsenzübung *Test*

Name, Vorname: Max

Matrikelnummer: Musterlösung

Tutoriumsnummer: 14

Aufgabe	1	2	3	4	5	Σ
Erreichte Punkte	A	l	l	e		
Mögliche Punktzahl	2 P	2 P	2 P	2 P	2 P	10 P

Bearbeitungshinweise

Für eine korrekte Lösung müssen nicht alle Lücken ausgefüllt werden. Die Größe der Lücken steht weiterhin nicht unbedingt in Relation zu den erreichbaren Punkten oderzur Länge des Textes, der für eine korrekte Lösung eingefüllt werden muss. Der von Ihnen eingefüllte Text muss kompilierbar sein. Verwenden Sie nur Elemente des Pakets java.lang der Java SE 17 Edition. Achten Sie im Folgenden auch auf Groß- und Kleinschreibung. Setzen Sie nur die in den folgenden Aufgabenstellungen angegebenen Informationen um. Sie müssen bei der Beantwortung der Präsenzübungsaufgaben keine Vorgaben zu Checkstyle-Regeln einhalten wie sonst bei der Bearbeitung der Übungsblätter.

Aufgabe A: Java-Basics (2 Punkte)

Ergänzen Sie die folgende Klasse um eine öffentliche und abstrakte Methode `average`, die ein Feld (*array*) von Ganzzahlen entgegennimmt und eine Fließkommazahl zurückgibt.

```
public abstract class AbstractWorker {
    public abstract float average(int[] array);

}
```

Aufgabe B: Aufzählungsdatentypen (2 Punkte)

Schreiben Sie einen Aufzählungsdatentyp (*enum*) `Direction`, der genau die Werte `NORTH`, `WEST`, `SOUTH` und `EAST` enthält.

```
[public] enum Direction {
    NORTH,
    WEST,
    SOUTH,
    EAST;
}
```

Aufgabe C: Kontrollfluss (2 Punkte)

Ergänzen Sie die Methode `getMedian` so, dass der Median des übergebenen, bereits sortierten Arrays berechnet wird.

- Hat das Array eine ungerade Zahl an Elemente, so ist der Median das mittlere Element des Arrays.
- Hat das Array eine gerade Zahl an Elementen, so ist der Median der Mittelwert der beiden mittleren Elemente des Arrays.

```
public double getMedian(double[] values) {
    if ( values.length % 2 == 0 ) {
        // Even length
        double first = values[values.length / 2];
        double second = values[values.length / 2 - 1];
        return (first + second) / 2;
    } else {
        // uneven length
        return values[values.length / 2];
    }
}
```

Aufgabe D: Konstruktoren und Methoden (2 Punkte)

Ergänzen Sie den Konstruktor der Klasse `Article` so, dass die Attribute sinnvoll mit dem Parametern gefüllt werden. `type` soll allerdings nur gesetzt werden, wenn der entsprechende Parameter nicht `null` ist.

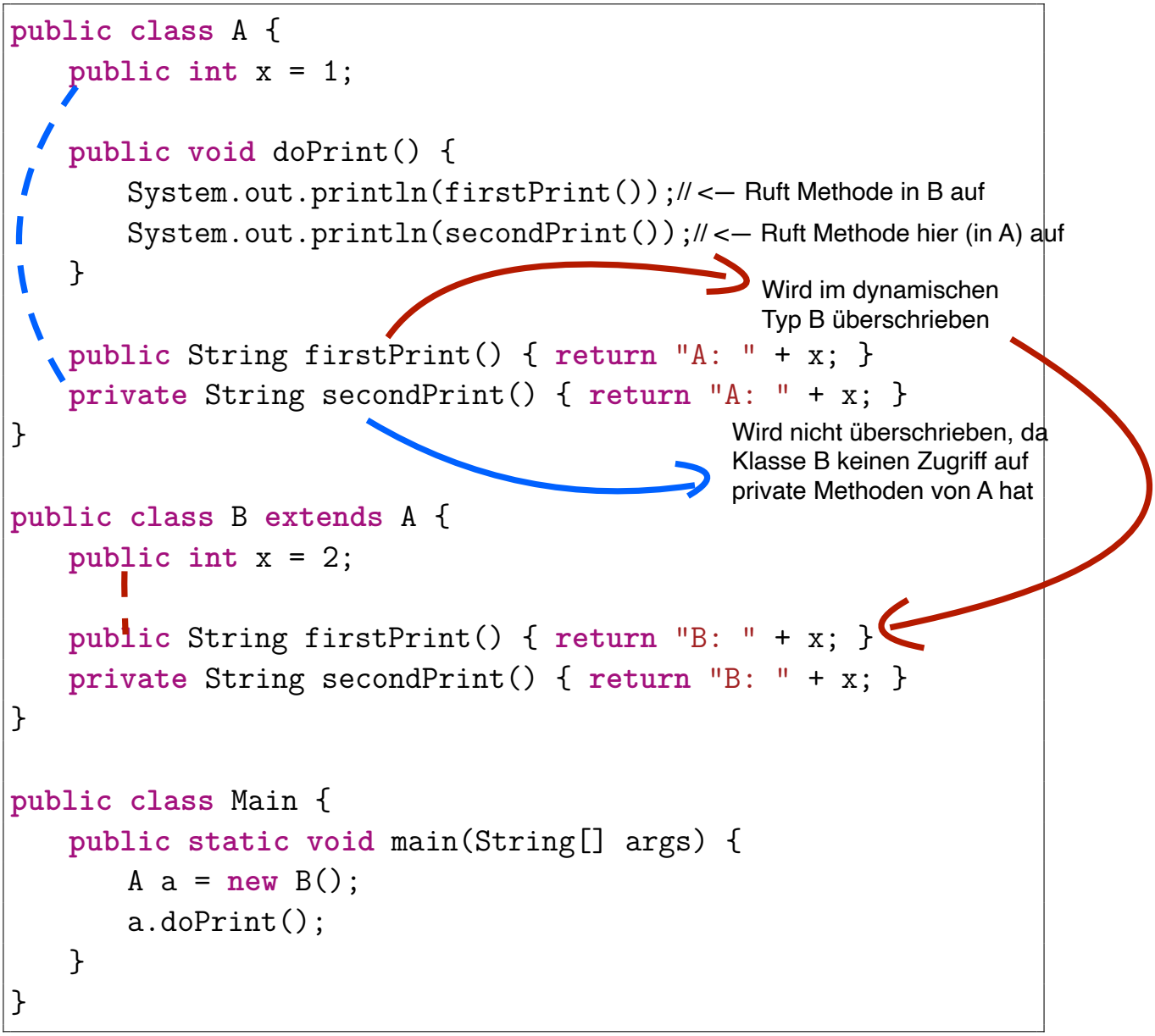
Ergänzen Sie die Methode `toString` der Klasse `Article` so, dass ein String der Form `ID: <id>, Typ: <type>` zurückgeben wird. Beispiel: `ID: 1623673, Typ: Fernbedienung`.

```
public class Article {
    private final int id;
    private final String type;

    public Article(int id, String type) {
        this.id = id;
        if (type != null) {
            this.type = type;
        }
    }

    public String toString() {
        return “ID: %d, Typ: %s”.formatted(id, type);

        // oder:

        return “ID: “ + id + “, Typ: “ + type;
    }
}
```

Aufgabe E: Vererbung (2 Punkte)

Geben Sie die Ausgabe des folgenden Quelltextes *zeilenweise* an.

```
public class A {
    public int x = 1;

    public void doPrint() {
        System.out.println(firstPrint());// <— Ruft Methode in B auf
        System.out.println(secondPrint());// <— Ruft Methode hier (in A) auf
    }

    public String firstPrint() { return "A: " + x; }
    private String secondPrint() { return "A: " + x; }
}

public class B extends A {
    public int x = 2;

    public String firstPrint() { return "B: " + x; }
    private String secondPrint() { return "B: " + x; }
}

public class Main {
    public static void main(String[] args) {
        A a = new B();
        a.doPrint();
    }
}
```



Ausgabe:

```
B: 2
A: 1
```