

Alle Aufgaben beziehen sich auf Java.

## 1. Multiple-Choice-Aufgaben zu Interfaces

25 / 25 Punkte

Die Aufgaben in diesem Abschnitt beziehen sich auf folgende Interfaces und Klassen:

```
interface Printable { void print(); }

interface Sortable { Sortable sort(); }

interface Computable { Computable compute(); }

interface Document { void read(String s); void write(String s); }

interface Text extends Document, Printable { Text format(); }

class Spreadsheet implements Document, Computable, Sortable {
    int row, col;
    public Spreadsheet(int row, int col) {
        this.row = row; this.col = col; ... }
    public void read(String s) { ... }
    public void write(String s) { ... }
    public Spreadsheet compute() {
        Spreadsheet sp = new Spreadsheet(row, col); ... return sp; }
    public Spreadsheet sort() {
        Spreadsheet sp = new Spreadsheet(row, col); ... return sp; }
    public void clear() { ... }
}

class TextDocument implements Text {
    public TextDocument() { ... }
    public void read(String s) { ... }
    public void write(String s) { ... }
    public TextDocument format() {
        TextDocument t = new TextDocument(); ... return t; }
    public void print() { ... }
    public void addLine(String l) { ... }
}
```

In jeder Aufgabe wird ein Objekt erzeugt, danach stehen mehrere mögliche Anweisungen. Welche der Anweisungen werden vom Java-Compiler ohne Fehlermeldung akzeptiert und liefern auch keine Fehler zur Laufzeit? Bitte wählen Sie alle gültigen Antwortmöglichkeiten aus.

### Aufgabe 1.1.

5 / 5 Punkte

```
Spreadsheet sp = new Spreadsheet(47,14);
```

- ☐ `Text t = ((Printable)sp).format();`
- ☒ `Sortable s = sp.compute(); s.toString();`
- ☒ `sp.write("outfile");`
- ☐ `sp = ((Text)sp).format();`

### Aufgabe 1.2.

5 / 5 Punkte

```
TextDocument td = new TextDocument();
```

- ☒ `td.read("infile");`
- ☐ `((Text)td).addLine("Hello World");`
- ☒ `Document d = ((Text)td).format(); d.toString();`
- ☐ `Sortable s = ((Sortable)td).sort();`

### Aufgabe 1.3.

5 / 5 Punkte

```
Document d = new TextDocument();
```

- ☐ `s = ((Sortable)d).sort();`
- ☒ `d = ((Text)d).format();`
- ☐ `d.print();`
- ☒ `((TextDocument)d).addLine("Hello World");`

### Aufgabe 1.4.

5 / 5 Punkte

```
Computable c = new Spreadsheet(47,15);
```

- ☐ `((Text)c).read("infile");`
- ☐ `((Document)c).clear();`
- ☒ `Document d = ((Spreadsheet)c).sort();`
- ☐ `((Printable)c).print();`

### Aufgabe 1.5.

5 / 5 Punkte

```
Printable p = new TextDocument();
```

- ☒ `Text t = ((Text)p).format();`
- ☒ `Document d = ((TextDocument)p).format(); d.toString();`
- ☐ `((Document)p).addLine("Hello World");`
- ☐ `((Document)p).print();`

## 2. Multiple-Choice-Aufgaben zu `equals` und `hashCode`

15 / 15 Punkte

In den folgenden Klassen sind die Implementierungen von `equals` und `hashCode` unvollständig. Ersetzen Sie die Buchstaben **A** und **B** durch einen oder mehrere der vorgeschlagenen Programmteile. Die Methoden müssen sich hinsichtlich der allgemeinen Bedingungen für `equals` und `hashCode` korrekt (und auch korrekt zueinander) verhalten. Bitte wählen Sie alle gültigen Antwortmöglichkeiten aus.

## Aufgabe 2.1.

5 / 5 Punkte

```
class Box {
    final private int weight;
    final private int volume;
    final private Color color;

    // ...

    public boolean equals(Object obj) {
        if (obj == null) return false;
        if (obj.getClass() != getClass()) return false;
        Box b = (Box)obj;
        A
    }

    public int hashCode() {
        B
    }
}
```

☐ A:

```
return b.weight == weight;
```

B:

```
if (color == null) {
    return 0;
} else {
    return weight;
}
```

☐ A:

```
return b.weight == weight;
```

B:

```
if (color == null) {
    return 0;
}
int hash = color.hashCode();
hash = hash * 31 + weight;
hash = hash * 31 + volume;
return hash;
```

✓ A:

```
return b.weight == weight;
```

B:

```
return weight;
```

✓ A:

```
return b.weight == weight  
    && b.volume == volume;
```

B:

```
return weight < volume ? 0 : weight;
```

## Aufgabe 2.2.

5 / 5 Punkte

```
class Point {  
    final private Integer x;  
    final private Integer y;  
  
    // ...  
  
    public boolean equals(Object obj) {  
        if (obj == null) return false;  
        if (obj.getClass() != getClass()) return false;  
        Point p = (Point)obj;  
        A  
    }  
  
    public int hashCode() {  
        B  
    }  
}
```

□ A:

```
return p.x != null && p.x.equals(x);
```

B:

```
if (x == null) {  
    return 0;  
}  
return x.hashCode() + y.hashCode();
```

✓ A:

```
if (p.x == null || p.y == null) return false;
return (p.x * p.y) == (x * y);
```

B:

```
if (x == null || y == null) {
    return 0;
}
return x * y;
```

✓ A:

```
return p.x != null && p.x.equals(x)
    && p.y != null && p.y.equals(y);
```

B:

```
if (x == null || y == null) {
    return 0;
}
return x.hashCode();
```

✓ A:

```
return p.x != null && p.x.equals(x)
    && p.y != null && p.y.equals(y);
```

B:

```
if (x == null) {
    return 0;
}
return x.hashCode();
```

### Aufgabe 2.3.

5 / 5 Punkte

```
class Watch {
    final private String serialNumber;
    final private boolean digital;

    public int currentTime() {
        return (int) (Math.random() * 86400);
    }

    // ...

    public boolean equals(Object obj) {
        if (obj == null) return false;
        if (obj.getClass() != getClass()) return false;
        Watch w = (Watch) obj;
        A
    }

    public int hashCode() {
        B
    }
}
```

☐ A:

```
return w.currentTime() == currentTime();
```

B:

```
return (int) ((0x97e2a1430e3ab551L * currentTime()) >> 32)
```

☐ A:

```
return w.currentTime() == currentTime();
```

B:

```
return currentTime();
```

☒ A:

```
return w.serialNumber != null
    && w.serialNumber.equals(serialNumber);
```

B:

```
if (serialNumber == null) {
    return 0;
}
return serialNumber.hashCode();
```

☐ A:

```
return w.serialNumber != null
    && w.serialNumber.equals(serialNumber)
    && w.digital == digital
    && w.currentTime() == currentTime();
```

B:

```
if (serialNumber == null) {
    return 0;
}
int hash = serialNumber.hashCode();
hash = hash * 31 + (digital ? 1 : 0);
hash = hash * 31 + currentTime();
return hash;
```