

Alle Aufgaben beziehen sich auf Java.

## 1. Multiple-Choice-Aufgaben zu Interfaces

21.25 / 25 Punkte

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

```
interface Movable {
    void move();
}

interface PointSet {
    boolean hasElement(Point p);
}

interface ShapeOnPlane extends Movable, PointSet {
    double area();
}

class Disk implements ShapeOnPlane {
    private Point c = new Point(0,0);
    private double r;
    public Disk(double r) { this.r = r; }
    public void move() { c.move(); }
    public double area() { return r*r*Math.PI; }
    public boolean hasElement(Point p) { return c.distance(p) <= r; }
}

class Point implements Movable {
    private double x, y;
    public Point(double initX, double initY) { x = initX; y = initY; }
    public void move() { x += 1; y += 1; }
    public double distance(Point p) { return Math.hypot(x-p.x, y-p.y); }
}
```

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

### Aufgabe 1.1.

3.75 / 5 Punkte

```
Point p = new Point(1,2);
```

`((Movable)p).distance(p);`

`((ShapeOnPlane)p).move();`

`p.move();`

`((Object)p).equals(p);`

### Aufgabe 1.2.

5 / 5 Punkte

```
ShapeOnPlane s = new Disk(1);
```

`s.hasElement(new Point(0,0));`

`((PointSet)s).area();`

`((Disk)s).move();`

`s.area();`

### Aufgabe 1.3.

5 / 5 Punkte

```
Movable m = new Disk(1);
```

`m.equals(new Disk(1));`

`m.hasElement(new Point(0,0));`

`Object o = m;`

`((PointSet)m).hasElement(new Point(1,2));`

### Aufgabe 1.4.

3.75 / 5 Punkte

```
Object o = new Disk(1);
```

`o = (PointSet) o;`

`o.equals(new Point(1,1));`

`((ShapeOnPlane)o).toString();`

`o.move();`

### Aufgabe 1.5.

3.75 / 5 Punkte

```
Movable m = new Point(1,1);
```

`PointSet ps = m;`

`m = (PointSet) m;`

`((Point)m).distance((Point)m);`

`m = new Disk(1);`

## 2. Single-Choice Fragen zu Arrays, Schleifen und Rekursion

6 / 15 Punkte

Folgende Implementierungen der Methode `lad` sind syntaktisch korrekt. Die Methode ermittelt das Maximum der absoluten Differenzen zwischen benachbarten Einträgen des Teilarrays von `a` ab (inklusive) Index `0` bis (inklusive) Index `to`. Vorbedingungen: `a != null` und `a.length > 1` und `0 < to` und `to < a.length`. Beispiel: für `a = { 1, 3, 20, 1 }` und `to = 3` ist das Ergebnis `19` und für `to = 2` ist das Ergebnis `17`. Geben Sie an, welche Aussage auf die jeweilige Implementierung zutrifft.

## Aufgabe 2.1.

3 / 3 Punkte

```
public int lad(int[] a, int to) {
    int max = 0;
    for (; to > 0; to--) {
        int dist = Math.abs(a[to-1] - a[to]);
        max = dist > max ? dist : max;
    }
    return max;
}
```

Welche Aussage trifft hier zu?

- Laufzeitfehler (führt mit bestimmten gültigen Eingaben zu einem Laufzeitfehler, bzw. zu ungültigem rekursiven Aufruf)
- falscher Wert (bei allen gültigen Eingaben kein Laufzeitfehler, aber liefert einen falschen Wert bei bestimmten gültigen Eingaben)
- korrekt (liefert für alle gültigen Argumente das korrekte Ergebnis)

## Aufgabe 2.2.

0 / 3 Punkte

```
public int lad(int[] a, int to) {
    int dist = a[to-1] > a[to] ? a[to-1] - a[to] : a[to] - a[to-1];
    int max = ( to == 0 ? 0 : lad(a,to-1) );
    return dist > max ? dist : max;
}
```

Welche Aussage trifft hier zu?

- Laufzeitfehler
- falscher Wert
- korrekt

### Aufgabe 2.3.

0 / 3 Punkte

```
public int lad(int[] a, int to) {
    int dist = Math.abs(a[to-1] - a[to]);
    if (to == 1)
        return dist;
    else
        return lad(a,to-1) > dist ? lad(a,to-1) : dist;
}
```

Welche Aussage trifft hier zu?

- Laufzeitfehler
- falscher Wert
- korrekt

### Aufgabe 2.4.

0 / 3 Punkte

```
public int lad(int[] a, int to) {
    if (to == 1) {
        return Math.abs(a[0] - a[1]);
    }
    int dist = Math.abs(a[to-1] - a[to]);
    int max = lad(a,to-1);
    return dist > max ? dist : max;
}
```

Welche Aussage trifft hier zu?

- Laufzeitfehler
- falscher Wert
- korrekt

## Aufgabe 2.5.

3 / 3 Punkte

```
public int lad(int[] a, int to) {
    int sum = 0, max = 0;
    while (1 < to) {
        int dist = a[to] - a[--to];
        dist = (dist > 0 ? dist : -dist);
        if (dist > max) {
            max = dist;
        }
    }
    return max;
}
```

Welche Aussage trifft hier zu?

- Laufzeitfehler
- falscher Wert
- korrekt