Nachtragstest in Programmkonstruktion

20.5 / 40 Punkte

Alle Aufgaben beziehen sich auf Java.

1. Multiple-Choice-Aufgaben zu Interfaces

17.5 / 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);
```

- [x] `p.move();`
- [x] `p.equals((Object)p);`
- [x] `((Movable)p).distance(p);`
- [ ] `((ShapeOnPlane) p).area();`

Aufgabe 1.2.

3.75 / 5 Punkte

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

- [x] `s.area();`
- [x] `((Disk)s).hasElement(new Point(0,0));`
- [x] `((PointSet)s).area();`
- [x] `s.move();`

Aufgabe 1.3.

3.75 / 5 Punkte

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

- [ ] `m.hasElement(new Point(0,0));`
- [x] `Object o = m;`
- [x] `m.equals(new Disk(1));`
- [ ] `((PointSet)m).hasElement(new Point(1,2));`

Aufgabe 1.4.

2.5 / 5 Punkte

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

- [x] `((ShapeOnPlane)o).toString();`
- [ ] `o = (PointSet) o;`
- [x] `o.equals(new PointSet());`
- [x] `o.move();`

Aufgabe 1.5.

3.75 / 5 Punkte

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

- [x] `m = (Point) m;`
- [x] `PointSet ps = m;`
- [x] `((Point)m).distance((Point)m);`
- [ ] `m = new Disk(1);`

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

3 / 15 Punkte

Folgende Implementierungen der Methode `isAscending` sind syntaktisch korrekt. Die Methode soll genau dann `true` liefern, wenn alle Elemente ab inklusive der Position `from` (mit aufsteigendem Index) aufsteigend sortiert sind. Geben Sie an, welche Aussage auf die jeweilige Implementierung zutrifft. Annahme: `a != null` und `a.length > 0` und `from >= 0` und `from < a.length`

Aufgabe 2.1.

0 / 3 Punkte

```
public static boolean isAscending(int[] a, int from) {
    while(from < a.length-1) {
        if(a[from] > a[from+1]) {
            return false;
        }
        from++;
    }

    return true;
}
```

Welche Aussage trifft hier zu?

- ○ Laufzeitfehler (erzeugt einen Laufzeitfehler bei bestimmten gültigen Eingaben)
- ● 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 static boolean isAscending(int[] a, int from) {
    if(from == a.length-1) {
        return true;
    }
    return a[from] <= a[from+1] && isAscending(a, from + 1);
}
```

Welche Aussage trifft hier zu?

- ○ Laufzeitfehler
- ● falscher Wert
- ○ korrekt

Aufgabe 2.3.

0 / 3 Punkte

```
public static boolean isAscending(int[] a, int from) {
    boolean current = a[from] <= a[from+1];
    if(from < a.length-1) {
        return isAscending(a, from + 1);
    }
    return current;
}
```

Welche Aussage trifft hier zu?

- [ ] Laufzeitfehler
- [ ] falscher Wert
- [x] korrekt

Aufgabe 2.4.

3 / 3 Punkte

```
public static boolean isAscending(int[] a, int from) {
    if(from == a.length-1) {
        return true;
    }
    if(a[from] <= a[from+1]) {
        isAscending(a, from + 1);
        return true;
    }
    return false;
}
```

Welche Aussage trifft hier zu?

- [ ] Laufzeitfehler
- [x] falscher Wert
- [ ] korrekt

Aufgabe 2.5.

0 / 3 Punkte

```
public static boolean isAscending(int[] a, int from) {
    boolean current = true;
    if(from < a.length-1) {
        current = a[from] <= a[from+1];
        isAscending(a, from + 1);
    }
    return current;
}
```

Welche Aussage trifft hier zu?

- (•) Laufzeitfehler
- ( ) falscher Wert
- ( ) korrekt