Kontrollfragen zu einfacher objektorientierter Programmierung

Dozent: Prof. Dr. Michael Eichberg
Kontakt: michael.eichberg@dhbw.de, Raum 149B
Version: 1.0

Kontrollfragen

1. Ist diese Aussage korrekt?

   In einer Java Datei können mehrere öffentlichen Klassen definiert werden.

2. Ist die Aussage korrekt?

   Der `new` Operator initialisiert die Attribute eines Objekts mit den Standardwerten.

3. Welche Aussage ist korrekt?

   `var cs = new Circle[10];`

   1. Die Anweisung definiert eine Referenzvariable vom Typ `Circle`.
   2. Die Anweisung erzeugt ein Array mit 10 Elementen vom Typ `Circle`.
   3. Der Typ von `cs` ist `Circle`.

4. Was passiert?

   Circle.java

   ```
   public class Circle {
       int x, y, r;

       private Circle(int x, int y, int r) { this.x = x; this.y = y; this.r = r; }

       static Circle create(int x, int y, int r) { return new Circle(x, y, r); }
   }
   ```

   Main.java

   ```
   String toString(Circle c) {
       return "Circle(" + c.x + "," + c.y + "," + c.r + ")";
   }

   void main(String[] args) { println(toString(Circle.create(1, 2, 3))); }
   ```

5. Was wird ausgegeben?

   ```
   var c1 = new Triangle();
   var c2 = new Triangle();
   var c3 = c2;
   println(c1 == c2);
   println(c2 == c3);
   ```

6. Was passiert/wie ist die Ausgabe?

   Circle.java

   ```
   public class Circle {
       public final static Circle UNIT = new Circle(0, 0, 1);
   ```

```
    int x, y, r;
    private Circle(int x, int y, int r) { this.x = x; this.y = y; this.r = r; }

    static Circle create(int x, int y, int r) {
        if (x == 0 && y == 0 && r == 1) return UNIT;
        else  return new Circle(x, y, r);
}  }
```

Main.java

```
void main(String[] args) {
    println(Circle.create(0, 0, 1) == Circle.create(0, 0, 1));
}
```

7. Stimmt die folgende Aussage?

   Ein Objekt wird dann aus dem Speicher entfernt, wenn kein Zeiger mehr auf das Objekt zeigt?

8. Welche Aussagen sind korrekt?

   1. `this` wird benötigt, um einen anderen Konstruktor aufzurufen.
   2. `this` ist eine Referenz auf das aktuelle Objekt.
   3. `this` ist in statischen Methoden verfügbar.

9. Welche der folgenden Verwendungen von this sind (ggf. in Konstruktoren und bei dem Vorhandensein entsprechender Attribute) korrekt?

   1. `this = new Circle();`
   2. `this.radius = 10;`
   3. `this();`

Ist diese Aussage korrekt?

In einer Java Datei können mehrere öffentlichen Klassen definiert werden.

Ist die Aussage korrekt?

Der `new` Operator initialisiert die Attribute eines Objekts mit den Standardwerten.

Welche Aussage ist korrekt?

```
var cs = new Circle[10];
```

1. Die Anweisung definiert eine Referenzvariable vom Typ `Circle`.
2. Die Anweisung erzeugt ein Array mit 10 Elementen vom Typ `Circle`.
3. Der Typ von `cs` ist `Circle`.

Was passiert?

Circle.java

```
public class Circle {
    int x, y, r;
    private Circle(int x, int y, int r) { this.x = x; this.y = y; this.r = r; }

    static Circle create(int x, int y, int r) { return new Circle(x, y, r); }
}
```

Main.java

```
String toString(Circle c) {
    return "Circle(" + c.x + "," + c.y + "," + c.r + ")";
}

void main(String[] args) { println(toString(Circle.create(1, 2, 3))); }
```

Was wird ausgegeben?

```
var c1 = new Triangle();
var c2 = new Triangle();
var c3 = c2;
println(c1 == c2);
println(c2 == c3);
```

Was passiert/wie ist die Ausgabe?

Circle.java

```java
public class Circle {
    public final static Circle UNIT = new Circle(0, 0, 1);
    int x, y, r;
    private Circle(int x, int y, int r) { this.x = x; this.y = y; this.r = r; }

    static Circle create(int x, int y, int r) {
        if (x == 0 && y == 0 && r == 1) return UNIT;
        else  return new Circle(x, y, r);
}  }
```

Main.java

```java
void main(String[] args) {
    println(Circle.create(0, 0, 1) == Circle.create(0, 0, 1));
}
```

Stimmt die folgende Aussage?

Ein Objekt wird dann aus dem Speicher entfernt, wenn kein Zeiger mehr auf das Objekt zeigt?

Welche Aussagen sind korrekt?

1. `this` wird benötigt, um einen anderen Konstruktor aufzurufen.
2. `this` ist eine Referenz auf das aktuelle Objekt.
3. `this` ist in statischen Methoden verfügbar.

Welche der folgenden Verwendungen von this sind (ggf. in Konstruktoren und bei dem Vorhandensein entsprechender Attribute) korrekt?

1. `this = new Circle();`
2. `this.radius = 10;`
3. `this();`