

Kontrollfragen zu einfacher objektorientierter Programmierung

Dozent: Prof. Dr. Michael Eichberg
Kontakt: michael.eichberg@dhw.de, Raum 149B
Version: 1.1

1. Grundlagen

Kontrollfragen

1.1. Ist diese Aussage korrekt?

In einer Java Datei können mehrere öffentlichen Klassen definiert werden.

1.2. Ist die Aussage korrekt?

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

1.3. Welche Aussage ist korrekt?

1 | 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`.

1.4. Was passiert wenn `main` ausgeführt wird?

Circle.java

```
1 public class Circle {  
2     int x, y, r;  
3     private Circle(int x, int y, int r) { this.x = x; this.y = y; this.r = r; }  
4     static Circle create(int x, int y, int r) { return new Circle(x, y, r); }  
5 }
```

Main.java

```
1 String toString(Circle c) { return "Circle(" + c.x + "," + c.y + "," + c.r + ")"; }  
2  
3 void main(String[] args) { IO.println(toString(Circle.create(1, 2, 3))); }
```

1.5. Was wird ausgegeben?

```
1 | var c1 = new Triangle();
2 | var c2 = new Triangle();
3 | var c3 = c2;
4 | IO.println(c1 == c2);
5 | IO.println(c2 == c3);
```

1.6. Was passiert/wie ist die Ausgabe?

Circle.java

```
1 public class Circle {  
2     public final static Circle UNIT = new Circle(0, 0, 1);  
3     int x, y, r;  
4     private Circle(int x, int y, int r) { this.x = x; this.y = y; this.r = r; }  
5  
6     static Circle create(int x, int y, int r) {  
7         if (x == 0 && y == 0 && r == 1) return UNIT;  
8         else return new Circle(x, y, r);  
9     } }
```

Main.java

```
1 void main(String[] args) {  
2     IO.println(Circle.create(0, 0, 1) == Circle.create(0, 0, 1));  
3 }
```

1.7. Stimmt die folgende Aussage?

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

2. Die Selbstreferenz **this**

Kontrollfragen

2.1. 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.

2.2. 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();`