



From good to great!

Gutes kann noch besser werden,
Sortiments- und Platzierungsoptimierung in einer neuen Dimension.

<https://github.com/HInformatikAG>



Lehreinheit 4

Ziele

Selbst Programmieren ;)

Java Keywords: this, static, void, return

Benennung von Variablen

Überdeckung von Variablen

Java Keywords

this, static, void, return

- this
 - Im Quellcode eine Referenz auf das Objekt/die Instanz der eigenen Klasse
 - Referenz = Variable, Zeiger

Java Keywords

this, static, void, return

- static
 - Variablen, die static sind, sind globale Variablen, die für alle Objekte einer Klasse gelten
 - Ändert man eine static Variable, so ändert man diese gleichzeitig für alle Objekte!
 - Eine static Variable zu ändern ist nicht empfehlenswert
 - Nützlich als Konstante, die sich nie ändert

Java Keywords

this, static, void, return

- void
 - Eine Methode mit dem Keyword void tut zwar etwas, teilt das Ergebnis aber niemandem mit
- return
 - Eine Methode, die nicht void ist, gibt irgendwas zurück, z. B. int oder double (primitive Datentypen)
 - `public int getTetrominoXKoordinate(){...}`
 - Eine Methode kann auch komplexe Datentypen zurück geben, z. B. ein Rechteck (ein Konstruktor ist ein Beispiel dafür)

Java Keywords

this, static, void, return

- return gibt den Wert, den eine Methode berechnet, zurück
- berechneDiagonale in der Klasse Rechteck gibt ein double zurück

```
public double berechneDiagonale() {
```

```
    /* c ist gesuchte Diagonale, a und b sind breite und laenge */
```

```
    /*  $a^2 + b^2 = c^2$  */
```

```
    /* Wurzel aus  $(a^2 + b^2)$  = Wurzel aus  $(c^2)$  */
```

```
    /*  $c = \text{Wurzel aus } (a^2 + b^2)$  */
```

```
    double diagonaleQuadrat = Math.pow(breite, 2) + Math.pow(laenge, 2);
```

```
    return Math.sqrt(diagonaleQuadrat);
```

```
}
```

Benennung von Variablen

- Legale Zeichen: Buchstaben, Zahlen, _, \$
- Konvention (allgemein bekannte Regel)
 - nur Buchstaben nutzen
 - Erster Buchstabe klein
 - CamelCase
 - erster Buchstabe klein, wenn Name aus mehreren Worten besteht, jeweils den ersten Buchstaben ab dem zweiten Wort groß
 - `Tetromino tetromino = new Tetromino();`
 - `Tetromino blauerTetromino = new Tetromino(blau);`
 - `Tetromino blauerTetrominoBlock = new Tetromino(TetrominoTyp.Block, blau);`

Überdeckung von Variablen

- Klasse hat Variable `private int flaeche`;
- Klasse hat Methode `berechneFlaeche(){...}`
- In `berechneFlaeche(){...}` gibt es auch eine Variable `int flaeche`;
- In `berechneFlaeche`: `System.out.println(flaeche)`
 - Was passiert?

```
int flache;
```

```
private double berechneFlaeche(int a, int b){
```

```
    in flaeche = a * b;  
    System.out.println(flaeche);  
    return flaeche;  
}
```

- Erweitere die Klasse Wuerfel um eine Methode, die das Ergebnis eines Würfelwurfs zurückgibt
 - Stichworte Rückgabetyp und return
 - Wie kann man Zufallszahlen erzeugen? Finde es heraus, alles ist erlaubt 😊
- Passe die Klasse FensterMitLabel so an, dass das Label das Ergebnis eines Würfelwurfs anzeigt
 - Es muss kein Button oder Ähnliches definiert werden (aber darf), die Anzeige kann sofort erfolgen
 - Stichworte Setter, Instanz, Methode nutzen

Vielen Dank!

