

Referenz

Tom Egermann

13. Januar 2022

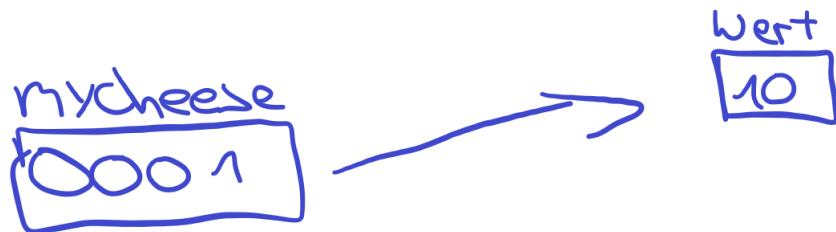
Immer dann, wenn Sie ein Objekt einer Klasse deklarieren, verwenden Sie einen Zeiger!

```
1 public class ZeigeZeiger{
2     Bruch b1, b2;
3
4     public ZeigeZeiger(){
5         b1 = new Bruch(3,4);
6         b2 = new Bruch(5,6);
7
8         System.out.println(b1);
9         System.out.println(b2);
10    }
11 }
```

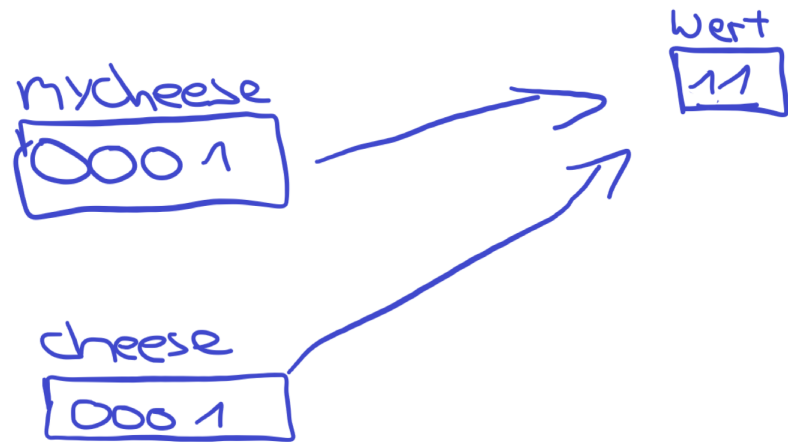
hier werden im Konstruktor die beiden Objekte der Klasse Bruch initialisiert, wenn wir die jetzt ausgeben wollen, dann bekommen wir die Adresse, wo sie sich gerade im Speicher befinden.

1 Beispiel

```
1 public static void main(String[] args) {
2     Cheese myCheese = new Cheese();
3     myCheese.setLevelOfStinkiness(10);
4     increaseStinkiness(myCheese);
5     System.out.println(myCheese.getLevel());
6 }
7
8 static void increaseStinkiness(Cheese cheese){
9     cheese.setLevelOfStinkiness(cheese.getLevel() + 1);
10 }
```



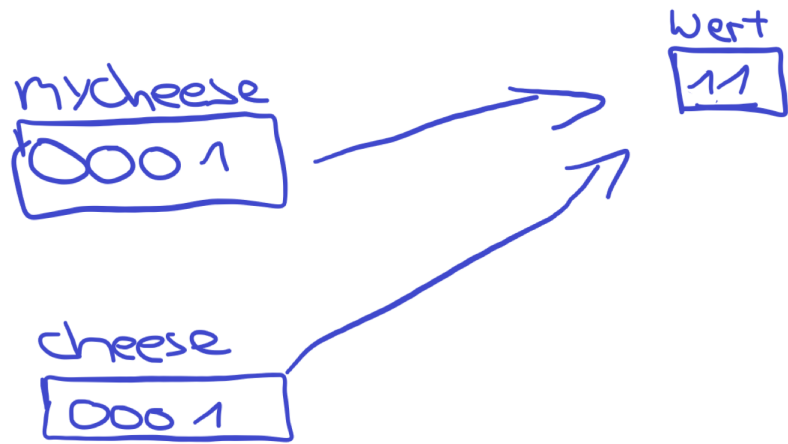
Vor dem aufrufen:



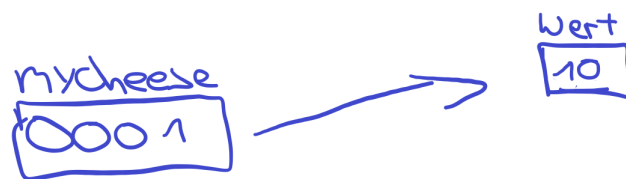
Nach dem aufrufen:

Man kann sehr gut sehen das wenn man die Methode aufruft der Adresse dem neuen Object mitgegeben wird, das Objekt kopiert sie sich also und arbeitet wie das Objekt myCheese

```
1 public static void main(String[] args) {
2     Cheese myCheese = new Cheese();
3     myCheese.setLevelOfStinkiness(10);
4     increaseStinkiness(myCheese);
5     System.out.println(myCheese.getLevel());
6 }
7
8 static void increaseStinkiness(Cheese cheese){
9     cheese = new Cheese();           //Schritt 1
10    cheese.setLevelOfStinkiness(756); //Schritt 2
11 }
```



Nach dem Aufrufen ohne Schritt 1,2:



Mit Schritt 1,2:

hieran kann man also sehen das man mit den Referenz vorsichtig umgehen muss.