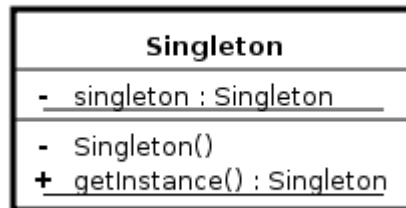


Merkblatt Design Pattern - Singleton

Zweck

Mit dem Einzelstück - Muster sorgen Sie dafür das nur ein einziges Objekt einer Klasse erstellt wird.

UML



Problem

Sie haben eine Klasse, welche eine beliebige Logik beinhaltet und das ganze System managen muss.

Lösung

Sie erstellen eine Singleton Klasse, welche nur mit einer Abfragemethode sich selbst erstellen kann und benutzen immer das gleiche Objekt.

Eigenschaften

- Kategorie: Erzeugungsmuster
- ist eines der Viererbande Muster (ist im Buch GoF enthalten) erzeugt und
- verwaltet ein einziges Objekt der Klasse
- bietet Zugriff auf dieses Objekt durch eine statische Abfragemethode (*getInstance()*) das
- private Attribut ist ein statisches Attribut Global verfügbar

Vorteile:

- Einfache Anwendung
 - Die Einzelstück Klasse ist schnell und unkompliziert geschrieben
- Die Klasse hat die eigene Verantwortung wie sie erstellt wird, da sie einen privaten Konstruktor hat.

Beispiel Java

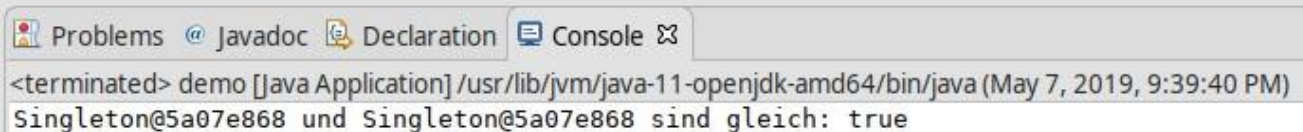
Singleton Klasse

```
public class Singleton {  
  
    private static Singleton INSTANCE;  
  
    private Singleton() {}  
  
    public static Singleton getInstance() {  
        if(INSTANCE == null)  
            INSTANCE = new Singleton();  
        return INSTANCE;  
    }  
  
    public void helloWorld() {  
        System.out.println("Hello World");  
    }  
}
```

Main

```
public class demo {  
    public static void main(String[] args) {  
        Singleton object1 = Singleton.getInstance();  
        Singleton object2 = Singleton.getInstance();  
        System.out.println(String.format("%s und %s sind gleich: %s",  
            object1, object2, object1 == object2));  
    }  
}
```

Output



The screenshot shows an IDE interface with tabs for Problems, Javadoc, Declaration, and Console. The Console tab is active, displaying the output of the Java application. The output indicates that the program has terminated and shows the memory addresses of the two Singleton objects, which are identical, confirming that only one instance was created.

```
<terminated> demo [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (May 7, 2019, 9:39:40 PM)  
Singleton@5a07e868 und Singleton@5a07e868 sind gleich: true
```