




Singleton


powered by APCON & SVC
Apcon Workplace Solutions
Member of itelligence

- ➔ Weitere Anforderungen an das Beispiel
- ➔ Singleton

APCON Workplace Solutions

Singleton

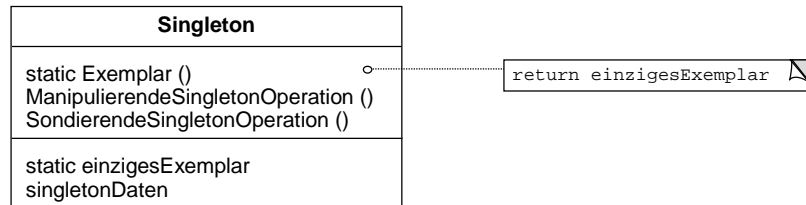

powered by APCON & SVC
Apcon Workplace Solutions
Member of itelligence

Zweck
Sichere ab, daß eine Klasse genau ein Exemplar besitzt, und stelle einen globalen Zugriffspunkt darauf bereit.

Anwendungsbeispiele

- Abstrakte Fabriken
- Window Manager
- Printer Spooler
- ...

Struktur des Singleton-Musters



- Ein Singleton ist für die Erzeugung und Verwaltung seines einzigen Exemplars zuständig.
- Klienten greifen auf das Singleton-Exemplar ausschließlich über die Exemplar() Operation der Singleton-Klasse zu:
`Singleton.Exemplar().ManipulierendeSingletonOperation()`

Beispiel: Der Standardzinssatz als Singleton

```

public class StandardZinssatz
{
    public static StandardZinssatz GibStandardZinssatz() {
        if (_instance == null){
            _instance = new StandardZinssatz();
        }
        return _instance;
    }
    public void SetzeStandardZinssatz(float NeuerWert){
    };
    public float GibWert(){
        return wert;
    }
    ...

    // geschützter Konstruktor
    protected StandardZinssatz(){
        wert = 2.0;
    }

    private float wert;
    private static StandardZinssatz _instance;
};
    
```

Diskussion des Singleton-Musters

- Da die Singleton-Klasse ihr einziges Exemplar kapselt, kann es kontrollieren, wie und wann Klienten darauf zugreifen.
- Etwas allgemeiner kann das Singleton-Muster eingesetzt werden, um die Anzahl der von einer Anwendung benutzten Exemplare zu steuern (statt einem Exemplar k Exemplare).
- Das Singleton-Muster vermeidet die Überfrachtung des Namensraumes mit globalen Variablen.