

::Singleton

21. April 2005

Was ist ein "Singleton"?

- ▶ ein Singleton ist ein Erzeugungsmuster, womit sichergestellt wird, dass nur eine Instanz einer bestimmten Klasse existiert
- ▶ diese Klassen haben einen globalen Zugriffspunkt (Methode), damit beliebig viele Klienten zugreifen können
- ▶ die Verwaltung wird von der Klasse selbst übernommen, damit kann die Klasse die Erzeugung neuer Instanzen abfangen, sowie die Zugriffe gewähren

Wann verwendet man ein "Singleton"?

- ▶ wenn es nur eine Instanz einer Klasse geben und es für Klienten an einem definierten Punkt zugreifbar sein muss
- ▶ wenn die einzige Instanz durch Bildung von Unterklassen erweiterbar sein soll und Klienten in der Lage sein sollen, die erweiterte Instanz ohne Veränderungen ihres Codes verwenden zu können

Wo werden "Singleton" verwendet?

- ▶ Druckerspooler
- ▶ Audioausgabe (DAC)
- ▶ Dateizugriff
- ▶ Datenbankzugriff

Wie sieht die "Singleton"-Struktur aus?

CSingleton
-static CSingleton* m_pSingleton
+static CSingleton* GetSingleton() #CSingleton()

Wie implementiert man ein "Singleton"?

in C++:

```
class CSingleton
{
public:
    ~CSingleton() {}
    static CSingleton *GetSingleton();

protected:
    CSingleton() {}

private:
    static CSingleton *m_pSingleton;
};

CSingleton* CSingleton::m_pSingleton = NULL;

CSingleton *CSingleton::GetSingleton()
{
    if(!m_pSingleton)
    {
        m_pSingleton = new CSingleton();
    }
    return m_pSingleton;
}

int main()
{
    CSingleton *x = CSingleton::GetSingleton();
    x-> ...

    return 0;
}
```

in Java:

```
public class CSingleton {

    private CSingleton() {}

    private static CSingleton m_Singleton =
        new CSingleton();

    public static CSingleton GetSingleton() {
        return m_Singleton;
    }
}

public class CTestSingleton {

    public static void main(String args[]) {
        CSingleton.GetSingleton(). ...;
    }
}
```