

# Refactoring

## 1 Allgemeines

---

### 1.1 Gründe

- Code ist unübersichtlich
- Ähnliche Funktionalitäten an mehreren Orten verschieden eingebaut
- Namensüberschneidungen
- Performanceprobleme

### 1.2 Nutzen

- Verbessert Leserlichkeit vom Code
- Ermöglicht mehr Wiederverwendung des Codes
- Verbesserung Struktur der Applikation
- Vereinfacht einlesen in den Code von neuen Entwicklern

### 1.3 Wann lieber kein Refactoring

- Wenn bei einem unsicheres Refactoring keine Tests zur Verfügung stehen
- Wenn das Refactoring vom Team nicht befürwortet wird

## 2 Varianten

---

### 2.1 Sicheres Refactoring

- Meist von der ID übernommen
- Z.B. Variablen umbenennen oder Schleifen umbauen

### 2.2 Unsicheres Refactoring

- Von Hand
- Z.B. Neuen Service einbauen, Methoden auslagern

## 3 Abläufe

---

### 3.1 Code Struktur verbessern

- Redundanz verhindern
- Single Responsibility Principle
- Komplexität reduzieren
- Code Funktion sollte dieselbe sein nach Refactoring

## 3.2 Namensgebung

- Namen klar und beschreibend
- Globale Variablen vermeiden

## 3.3 Grosse Codeblöcke aufteilen

- Grosse Klassen in kleinere separieren
- Lange Methoden aufspalten

## 3.4 Abstraktion

- Interfaces und Abstraktionen benützen
- Abhängigkeiten reduzieren

## 3.5 Testgetrieben

- Sicherstellen, dass Tests vorhanden sind, welche das Verhalten vom behandelten Code überprüfen
- Tests sollten nicht angepasst werden

## 3.6 Performance Optimierung

- Bottlenecks identifizieren
- Unnötige Ressourcen/Code vermeiden

# 4 Beispiel

---

## 4.1 Vor Refactoring

```
using System;

namespace MyConsoleApp
{
    0 references
    public class Program
    {
        0 references
        public static void Main(string[] args)
        {
            int a = 5, b = 10, c = 15;
            int r = a + b * c;
            if (r > 50)
            {
                Console.WriteLine("big num");
            }
            else
            {
                Console.WriteLine("small num");
            }
        }
    }
}
```

## 4.2 Nach Refactoring

```
using System;

namespace MyConsoleApp
{
    0 references
    public class Program
    {
        0 references
        public static void Main(string[] args)
        {
            int firstNumber = 5, secondNumber = 10, thirdNumber = 15;
            int result = firstNumber + secondNumber * thirdNumber;

            Console.WriteLine(result > 50 ? "Large number" : "Small number");
        }
    }
}
```