

## Entwurfsmuster für Java, C++ oder C#-Entwickler (S721)

Eigentlich ganz einfach: Es gibt typische Aufgaben und dazu passende, typische Lösungen. Dennoch zählt das Thema Design Patterns unter Softwareentwicklern zu den anspruchsvolleren Themen. Das Wissen über verfügbare Patterns allein reicht nicht aus. Es ist auch notwendig, ein Gefühl für Design Pattern zu entwickeln. Es gilt, den Blick dafür zu schärfen und zwar sowohl in der Modellierungsphase als auch in der nachfolgenden Implementierungsphase. Der Kurs wird mit Eclipse durchgeführt, Teilnehmer können aber auch andere Entwicklungsumgebungen verwenden. Der Kurs ist auch für C++- und C#-Entwickler geeignet.

### Offene Termine

Termin	Tage	Freie Plätze	Ort	Preis
25.03.-27.03.2019 ✓ <span style="border: 1px solid black; padding: 0 2px;">3=2</span>	3	>3	Köln	€ 2.170,00 *
12.06.-14.06.2019 ✓ <span style="border: 1px solid black; padding: 0 2px;">3=2</span>	3	>3	Köln	€ 2.170,00 *
26.08.-28.08.2019 ✓ <span style="border: 1px solid black; padding: 0 2px;">3=2</span>	3	>3	Köln	€ 2.170,00 *
28.10.-30.10.2019 ✓ <span style="border: 1px solid black; padding: 0 2px;">3=2</span>	3	>3	Köln	€ 2.170,00 *
02.12.-04.12.2019 ✓ <span style="border: 1px solid black; padding: 0 2px;">3=2</span>	3	>3	Köln	€ 2.170,00 *

#### \* Buchen ohne Risiko

- › Keine Vorkasse
- › Kostenloses Storno bis zum Vortag des Seminars
- › Rechnung nach erfolgreichem Seminar

#### ✓ Garantierter Termin und Veranstaltungsort

€ Preise zzgl. Mehrwertsteuer

3=2 Der dritte Mitarbeiter nimmt kostenlos teil

### Weitere Buchungsmöglichkeiten

<b>Firmenschulung</b>	Schulung für Ihre Mitarbeiter mit individuellen Inhalten zum Wunschtermin im GFU-Schulungszentrum.
<b>Inhouse-Schulung</b>	Schulung für Ihre Mitarbeiter mit individuellen Inhalten zum Wunschtermin in Ihrem Hause.
<b>Individualschulung</b>	Schulung für eine Einzelperson mit individuellen Inhalten zum Wunschtermin, wahlweise in Ihrem Hause oder im GFU-Schulungszentrum.

### Schulungs-Ziel

Es werden viele der Patterns aus dem Buch-Klassiker "Design Patterns" der GoF ("Gang of

### Inhalt

- › **Einführung in Design Pattern**
- › **Pattern in der "Realen Welt"**

Four") erarbeitet. Diese Kenntnisse heben die Qualität der Implementierung, helfen aber auch bei der Erstellung von UML-Klassendiagrammen in der Designphase. Bei einem hohen Vorwissen der Teilnehmer wird die Agenda um weitere Patterns ergänzt.

## Wer sollte teilnehmen

---

Erfahrene Softwareentwickler mit Java, C++ oder C#-Kenntnissen, die sich verbessern möchten.

## Organisation

---

### Teilnehmerzahl

min. 1, max. 8 Personen

### Seminarzeiten

3 Tage, 1. Tag 10:00 - 17:00 Uhr, Folgetage 09:00 - 16:00 Uhr

### Ort der Schulung

GFU-Schulungszentrum Köln oder bei Ihnen als Inhouse-Schulung

## Enthaltene Leistungen

---

### Im Preis enthalten:

- > Voll ausgestatteter Arbeitsplatz pro Teilnehmer
- > Fachbuch zum Seminar
- > Teilnahmezertifikat
- > Kostenloser persönlicher Parkplatz
- > Kostenloser Shuttle-Service
- > Frühstück, Snacks und Getränke ganztägig
- > Mittagessen im eigenen Restaurant, täglich 6 Menüs, auch vegetarisch

## Haben Sie Fragen?

---

Gerne beraten wir Sie persönlich per [Mail](#) oder Telefon.

- > [info@gfu.net](mailto:info@gfu.net)
- > Infoline 0221 82 80 90

- > **Beispiele und Übungen für Erzeugungsmuster**
  - > "Singleton" - von manchen Objekten darf es nur eine Instanz geben
  - > "Factory" - Erzeugung ähnlicher Objekte
  - > "Prototyp" - komplexe Objekte anhand einer Vorlage erzeugen
- > **Beispiele und Übungen für Strukturmuster**
  - > "Facade" - einfache Schnittstellen für komplexe Systeme schaffen
  - > "Observer" - Kommunikationsstruktur für unabhängige Komponenten
  - > "Composite" - die Implementierung für "Teile-Ganze-Beziehungen"
  - > "Decorator" - eine flexible Form, um eine Klasse zu erweitern
  - > "Proxy" - ein vorgelagertes Stellvertreterobjekt
  - > "Adapter" - inkompatible Objekte über Schnittstellen koppeln
- > **Beispiele und Übungen für Verhaltensmuster**
  - > "Command" - Trennung von Befehlen und Ausführung
  - > "Iterator" - Elemente einer Liste durcharbeiten
  - > "Visitor" - Auslagerung von Operationen in externe Klassen mit Interface-Kopplung
- > **Mikropattern (bei Bedarf)**
  - > Wie wird eine equals()-Methode sauber implementiert
  - > Verschiedene Möglichkeiten, um clone() zu implementieren
  - > Die Rolle des Konstruktors