

# Applikationsentwicklung mit Java – Vertiefung («JAVA05»)

Erweitern Sie Ihre Grundkenntnisse der Applikationsentwicklung mit der Programmiersprache Java um fortgeschrittene Konzepte! Sie erhalten in diesem Kurs einen Überblick über deren vielseitigen Einsatzmöglichkeiten.

**Dauer:** 2 Tage

**Preis:** 1'350.– zzgl. 8.1% MWST

**Kursdokumente:** Digicomp Kursunterlagen

## Inhalt

1. Singleton Pattern
  - Beschreibung
  - Modellierung
2. Builder Pattern
  - Beschreibung
  - Modellierung
3. Introspektion
  - Klassenobjekt
  - Klasse `java.lang.Class`
  - Klasse `java.lang.Package`
  - Schnittstelle `java.lang.reflect.Type`
  - Schnittstelle `java.lang.reflect.Member`
  - Klasse `java.lang.reflect.AccessibleObject`
  - Klasse `java.lang.reflect.Field`
  - Klasse `java.lang.reflect.Executable`
  - Klasse `java.lang.reflect.Constructor`
  - Klasse `java.lang.reflect.Method`
  - Klasse `java.lang.reflect.Parameter`
  - Klasse `java.lang.reflect.Modifier`
4. Zahlen und Währungen
  - Klasse `java.lang.Math`
  - Klasse `java.math.BigInteger`
  - Klasse `java.math.BigDecimal`
  - Klasse `java.util.Currency`
5. Daten und Zeiten
  - Klasse `java.util.Date`
  - Klasse `java.util.Calendar`
  - Klasse `java.util.GregorianCalendar`
  - Schnittstelle `java.time.temporal.Temporal`
  - Klasse `java.time.Instant`
  - Klasse `java.time.LocalDate`
  - Klasse `java.time.LocalDateTime`
  - Klasse `java.time.ZonedDateTime`
  - Klasse `java.time.OffsetDateTime`
  - Klasse `java.time.Period`
  - Klasse `java.time.Duration`
  - Schnittstelle `java.time.temporal.TemporalAmount`
  - Klasse `java.time.Period`
  - Klasse `java.time.Duration`
  - Schnittstelle `java.time.temporal.TemporalUnit`

- Enumeration `java.time.temporal.ChronoUnit`
- Klasse `java.time.format.DateTimeFormatter`
- Enumeration `java.time.format.FormatStyle`

#### 6. Listen

- Schnittstelle `java.util.List`
- Klasse `java.util.ArrayList`
- Klasse `java.util.AbstractList`

#### 7. Mengen

- Schnittstelle `java.util.Set`
- Klasse `java.util.HashSet`
- Klasse `java.util.AbstractSet`

#### 8. Abbildungen

- Schnittstelle `java.util.Map`
- Schnittstelle `java.util.Map.Entry`
- Klasse `java.util.HashMap`
- Klasse `java.util.AbstractMap`
- Lambdas und Closures
- Funktionale Schnittstellen

#### 9. Lambda-Ausdrücke

- Annotation `java.lang.FunctionalInterface`
- Filteralgorithmen
- Ausführungsalgorithmen
- Transformationsalgorithmen
- Erzeugungsalgorithmen
- Methodenreferenzen

## Key Learnings

- Kennen des fortgeschrittenen Vorgehens bei der Applikationsentwicklung mit der Programmiersprache Java

## Zielpublikum

Dieser Kurs richtet sich an Programmierer, die bereits das grundlegende Vorgehen bei der Applikationsentwicklung mit der Programmiersprache Java kennen und nun ihre Kenntnisse vertiefen wollen.

## Anforderungen

Besuch des folgenden Kurses oder gleichwertige Kenntnisse:

- [Applikationsentwicklung mit Java – Grundlagen \(«JAVA04»\)](#)

## Weiterführende Kurse

- [Collections Framework von Java \(«JAVA06»\)](#)
- [Textverarbeitung mit Java \(«JAVA07»\)](#)
- [Assessment – Oracle Certified Associate Java Programmer SE 8 \(OCAJP\)](#)
- [Java SE Application Developer Professional \(«JSEADP»\)](#)

**Haben Sie Fragen oder möchten Sie einen Firmenkurs buchen?**

Wir beraten Sie gerne unter 044 447 21 21 oder [info@digicomp.ch](mailto:info@digicomp.ch). Detaillierte Infos zu den Terminen finden Sie unter [www.digicomp.ch/weiterbildung-software-engineering/programmiersprachen/java-javascript/kurs-applikationsentwicklung-mit-java-vertiefung](http://www.digicomp.ch/weiterbildung-software-engineering/programmiersprachen/java-javascript/kurs-applikationsentwicklung-mit-java-vertiefung)

