NEXIO Online Akademie GmbH

Softwaredesign II: Analyse und Design in der Softwareentwicklung - von der Architektur zur Implementierung (Algorithmen)

Sonstiges · Anbieter: NEXIO Online Akademie GmbH

FÖRDERUNG & KOSTEN

Kurspreis: 2.966 €

Förderfähig über: Bildungsgutschein

AZAV-zertifizierter Bildungsträger — Förderung durch die Agentur für Arbeit möglich

Kursdaten auf einen Blick

Durchführung	Online
Lernformat	Zeitgebunden
Dauer	4 Wochen in Vollzeit
Unterrichtssprache	Deutsch
Kursstart	Startet jede Woche
Vorkenntnisse	Grundlegende Kenntnisse
Abschluss	Trägerinternes Zertifikat bzw. Teilnahmebescheinigung

Bildungsträger & Kontakt

NEXIO Online Akademie GmbH
Am Herrschaftsweiher 11, 67071 Ludwigshafen am Rhein
Telefon: 0800 100 87 34
E-Mail: beratung@nexio-akademie.de
Web: https://nexio-akademie.de

Kurs online ansehen

01 Kursbeschreibung

Der Kurs baut auf den grundlegenden Kenntnissen des Softwaredesigns und der Softwarearchitektur auf und konzentriert sich auf die praktische Umsetzung dieser Architektur in konkrete Implementierungen. Im Mittelpunkt stehen zunächst Algorithmen und deren Umsetzung: Wie können abstrakte Architekturmodelle und Entwürfe in effiziente und robuste Algorithmen übersetzt werden? Danach geht es um den Übergang von Design zu Programmierung: Der Kurs vermittelt Ihnen den Prozess und die Prinzipien, um von der Architektur zur realen Softwareimplementierung zu gelangen. Sie lernen, Performance- und Skalierungsthemen frühzeitig zu berücksichtigen, Hotspots zu erkennen (Suche/Sortierung/Aggregation), Datenstrukturen sinnvoll zu wählen und Caching pragmatisch einzusetzen. Sie lernen, Qualität und Zuverlässigkeit durch Fehler- und Ausnahmebehandlung abzusichern. Sie entwickeln defensive Strategien und systematische Testfälle direkt aus Pseudocode, Verträgen und Abläufen. Außerdem beschäftigen Sie sich mit Qualitätssteigerung im Code durch Design-Patterns und algorithmische Konzepte, die zu wartbarem, erweiterbarem und performantem Code führen. Sie erwerben ein Verständnis für Datenstrukturen und algorithmische Techniken als Basis für die Implementierung von Softwaremodulen gemäß der vorigen Design-Schritte. Nach Kursabschluss können Sie fachliche Abläufe 'in Code denken' - klar, effizient, robust - und sie in einer Architektur sauber verankern. Das macht Sie in Entwicklungsteams schnell zu einer gefragten Person für die 'entscheidenden Details', die Qualität und Geschwindigkeit im Projekt bestimmen.

02 Lehrplan & Module

1. Von Use Cases zu Algorithmen

Du lernst, wie Du aus Use Cases und Anforderungen schrittweise konkrete Abläufe und Algorithmen ableitest. Dabei übersetzt Du fachliche Szenarien in Kontrollflüsse, Eingaben, Ausgaben und Randfälle.

2. Qualität & Traceability

Im Mittelpunkt steht die Rückverfolgbarkeit von Anforderungen über Design und Code bis zu den Tests. Du setzt Qualitätsmerkmale wie Wartbarkeit und Testbarkeit als Kriterien ein, um Implementierungsentscheidungen nachvollziehbar zu machen.

3. SOLID/GRASP in der Implementierung

Du wendest die SOLID-Prinzipien und die GRASP-Muster direkt im Code an, etwa bei der Verteilung von Zuständigkeiten auf Klassen. So erkennst Du, wie lose Kopplung und hohe Kohäsion die Änderbarkeit einer Implementierung verbessern.

4. Datenstrukturen

Behandelt werden zentrale Datenstrukturen wie Arrays, Listen, Stacks, Queues, Bäume, Hashtabellen und Graphen. Du lernst, sie anhand von Zugriffsmustern und Anforderungen begründet auszuwählen.

5. Algorithmische Korrektheit

Du prüfst, ob ein Algorithmus das tut, was er soll, und arbeitest dazu mit Vor- und Nachbedingungen sowie Schleifeninvarianten. Ergänzend leitest Du aus diesen Bedingungen gezielte Testfälle ab.

6. Laufzeitaspekte in der Implementierung

In diesem Modul geht es um Zeit- und Speicherkomplexität sowie die O-Notation. Du schätzt das Laufzeitverhalten Deiner Implementierungen ein und erkennst typische Engpässe in Schleifen und Datenzugriffen.

7. Ereignisgetriebene Implementierung & Persistenz

Du setzt ereignisgesteuerte Abläufe mit Events, Listenern und Nachrichten um. Außerdem lernst Du, wie Daten über Repositories, Dateien oder Datenbanken dauerhaft gespeichert und wieder geladen werden.

8. Architekturkommunikation in der Umsetzung

Hier erfährst Du, wie Architekturentscheidungen im Entwicklungsteam verständlich kommuniziert werden, etwa über Diagramme, Schnittstellenbeschreibungen und Entscheidungsprotokolle. So bleibt die Umsetzung konsistent mit der geplanten Architektur.

9. Realisierungskonsens

Du lernst, wie sich Entwicklerinnen, Entwickler und Fachseite auf eine gemeinsame Umsetzungslösung einigen. Code-Reviews, Design-Diskussionen und das Abwägen von Alternativen stehen dabei im Vordergrund.

10. Qualitätssicherung

Abschließend beschäftigst Du Dich mit Unit-Tests, Integrationstests, statischer Codeanalyse und Reviews. Damit sicherst Du die Qualität der Implementierung systematisch ab.

03 Enthaltene Zertifikate

Trägerzertifikat: Trägerinternes Zertifikat bzw. Teilnahmebescheinigung

04 Voraussetzungen

Vorkenntnisse

Niveau: Grundlegende Kenntnisse

Vorausgesetzt werden Grundkenntnisse in Programmierung, OO-Konzepte sowie ein Verständnis gängiger Entwicklungsprozesse. Grundkenntnisse in einer oder mehreren Programmiersprachen (z.B. Python, Java, C#, JavaScript) sind erforderlich, wenn Implementierungen konkret programmiert werden sollen.

05 Zielgruppen & Berufsperspektiven

- Arbeitssuchende

Mögliche Berufsbilder: Softwareentwickler, Anwendungsentwickler, Fachinformatiker

NÄCHSTER SCHRITT

Kursdetails online ansehen und Anbieter kontaktieren

NEXIO Online Akademie GmbH · Am Herrschaftsweiher 11, 67071 Ludwigshafen am Rhein
0800 100 87 34 · beratung@nexio-akademie.de · https://nexio-akademie.de

Online ansehen