

Weiterbildender Masterstudiengang

Softwareentwicklung





Weiterbildender Masterstudiengang Softwareentwicklung

- » Modalität: online
- » Dauer: 12 Monate
- » Qualifizierung: TECH Global University
- » Akkreditierung: 60 ECTS
- » Zeitplan: in Ihrem eigenen Tempo
- » Prüfungen: online

Internetzugang: www.techtitute.com/de/informatik/masterstudiengang/masterstudiengang-softwareentwicklung



Index

01

Präsentation des Programms

Seite 4

02

Warum an der TECH studieren?

Seite 8

03

Lehrplan

Seite 12

04

Lehrziele

Seite 24

05

Karrieremöglichkeiten

Seite 30

06

Inbegriffene Softwarelizenzen

Seite 34

07

Studienmethodik

Seite 38

08

Lehrkörper

Seite 48

09

Qualifizierung

Seite 52

01

Präsentation des Programms

Die Softwareentwicklung erlebt ein beispielloses Wachstum, mit einer konstanten Nachfrage nach Experten, die in der Lage sind, effiziente, skalierbare und sichere Lösungen zu entwickeln. Laut den Vereinten Nationen haben mehr als 60% der Unternehmen Schwierigkeiten, Stellen im Technologiebereich zu besetzen, insbesondere in der Softwareentwicklung. Angesichts dieser Situation ist es unerlässlich, über aktuelle Kenntnisse zu verfügen, die auf die tatsächlichen Anforderungen der IT-Umgebung ausgerichtet sind. In diesem Sinne präsentiert TECH einen innovativen und zu 100% online absolvierten Studiengang im Bereich Softwareentwicklung, der für Fachleute konzipiert ist, die sich zu einem hochgradig wettbewerbsfähigen technischen Profil weiterentwickeln möchten, das an die aktuellen Herausforderungen der Branche angepasst ist.



“

Nehmen Sie an einem akademisch anspruchsvollen Universitätsprogramm teil, in dem Sie lernen, neue Trends in der Softwareentwicklung mit KI-Technologien umzusetzen”

Die Softwareentwicklung ist ein strategischer Schwerpunkt für die digitale Transformation in allen Produktionsbereichen, vom Gesundheitswesen über das Bildungswesen bis hin zum Handel. In diesem Zusammenhang macht die Schaffung effizienter, skalierbarer und sicherer digitaler Lösungen den Unterschied in der Wettbewerbsfähigkeit von Unternehmen und Institutionen aus. In diesem Zusammenhang ist es weltweit zu einer Priorität geworden, über Spezialisten zu verfügen, die in der Lage sind, komplexe technologische Systeme zu entwerfen, aufzubauen und zu warten, was die Nachfrage nach hochqualifizierten Fachkräften in diesem Bereich ankurbelt.

Aus diesem Grund führt TECH einen innovativen Weiterbildenden Masterstudiengang in Softwareentwicklung ein. Dieses Programm vermittelt Schlüsselkompetenzen in den Bereichen Programmierung, Systemarchitektur, agile Methoden und neue Technologien, wobei stets ein praktischer und aktueller Ansatz verfolgt wird. Durch einen rigorosen, auf die Lösung realer Probleme ausgerichteten Lehrplan wird eine ganzheitliche Sichtweise der Softwareentwicklung gefördert. Diese Fortbildung verbessert nicht nur die beruflichen Chancen, sondern erleichtert auch den Zugang zu technischen Führungspositionen, Beratung und der Entwicklung innovativer Lösungen mit Auswirkungen auf verschiedene Branchen.

Hervorzuheben ist, dass das Online-Format des Programms es ermöglicht, ohne geografische Barrieren oder zeitliche Einschränkungen voranzukommen. Darüber hinaus bietet die virtuelle Plattform kontinuierlichen Zugang zu hochwertigen Inhalten, interaktiven Ressourcen und angewandten Übungen und fördert so ein flexibles, autonomes und an jedes Tempo angepasstes Lernen. Diese Modalität erleichtert die Vereinbarkeit mit anderen Aktivitäten und fördert gleichzeitig eine Bildungserfahrung, die auf die aktuellen Dynamiken des technologischen und beruflichen Umfelds abgestimmt ist. Darüber hinaus wird das Programm von einem renommierten internationalen Gastdirektor begleitet, der 10 anspruchsvolle Masterclasses anbieten wird.

Dieser **Weiterbildender Masterstudiengang in Softwareentwicklung** enthält das vollständigste und aktuellste Programm auf dem Markt. Seine herausragendsten Merkmale sind:

- Die Entwicklung von Fallstudien, die von Experten für Softwareentwicklung vorgestellt werden
- Der anschauliche, schematische und äußerst praxisnahe Inhalt vermittelt alle für die berufliche Praxis unverzichtbaren wissenschaftlichen und praktischen Informationen
- Praktische Übungen, bei denen der Selbstbewertungsprozess zur Verbesserung des Lernens genutzt werden kann
- Sein besonderer Schwerpunkt liegt auf innovativen Methoden
- Theoretische Lektionen, Fragen an den Experten, Diskussionsforen zu kontroversen Themen und individuelle Reflexionsarbeit
- Die Verfügbarkeit des Zugangs zu Inhalten von jedem festen oder tragbaren Gerät mit Internetanschluss

“

Ein renommierter internationaler Gastdirektor wird zehn exklusive Masterclasses zu den neuesten Trends in der Softwareentwicklung anbieten“

“

Sie werden verteilte und vernetzte Anwendungen entwickeln, die den Herausforderungen der digitalen Transformation in Bezug auf Konnektivität, Sicherheit und Leistung gerecht werden“

Zu den Dozenten gehören Fachleute aus dem Bereich der Softwareentwicklung, die ihre Erfahrungen in dieses Programm einbringen, sowie anerkannte Spezialisten von führenden Gesellschaften und renommierten Universitäten.

Die multimedialen Inhalte, die mit der neuesten Bildungstechnologie entwickelt wurden, ermöglichen der Fachkraft ein situiertes und kontextbezogenes Lernen, d. h. eine simulierte Umgebung, die eine immersive Fortbildung bietet, die auf die Ausführung von realen Situationen ausgerichtet ist.

Das Konzept dieses Programms konzentriert sich auf problemorientiertes Lernen, bei dem der Student versuchen muss, die verschiedenen Situationen aus der beruflichen Praxis zu lösen, die während des gesamten Studiengangs gestellt werden. Dabei wird die Fachkraft durch ein innovatives interaktives Videosystem unterstützt, das von anerkannten Experten entwickelt wurde.

Sie werden relationale Datenbanken in komplexe Architekturen integrieren und so die Verwaltung großer Datenmengen optimieren.

Mithilfe des von TECH verwendeten Relearning-Systems werden Sie den Zeitaufwand für das Lernen und Auswendiglernen reduzieren.



02

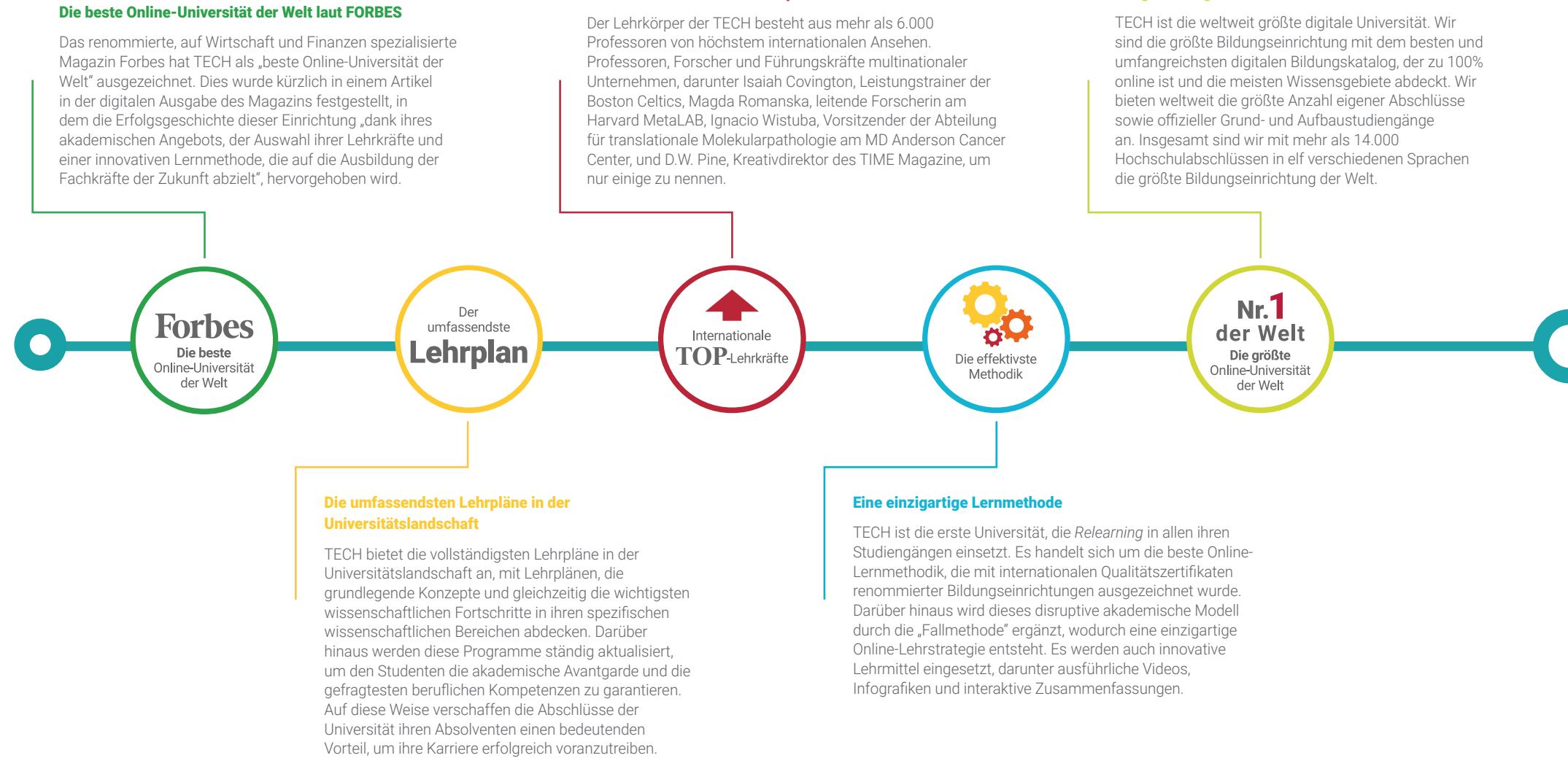
Warum an der TECH studieren?

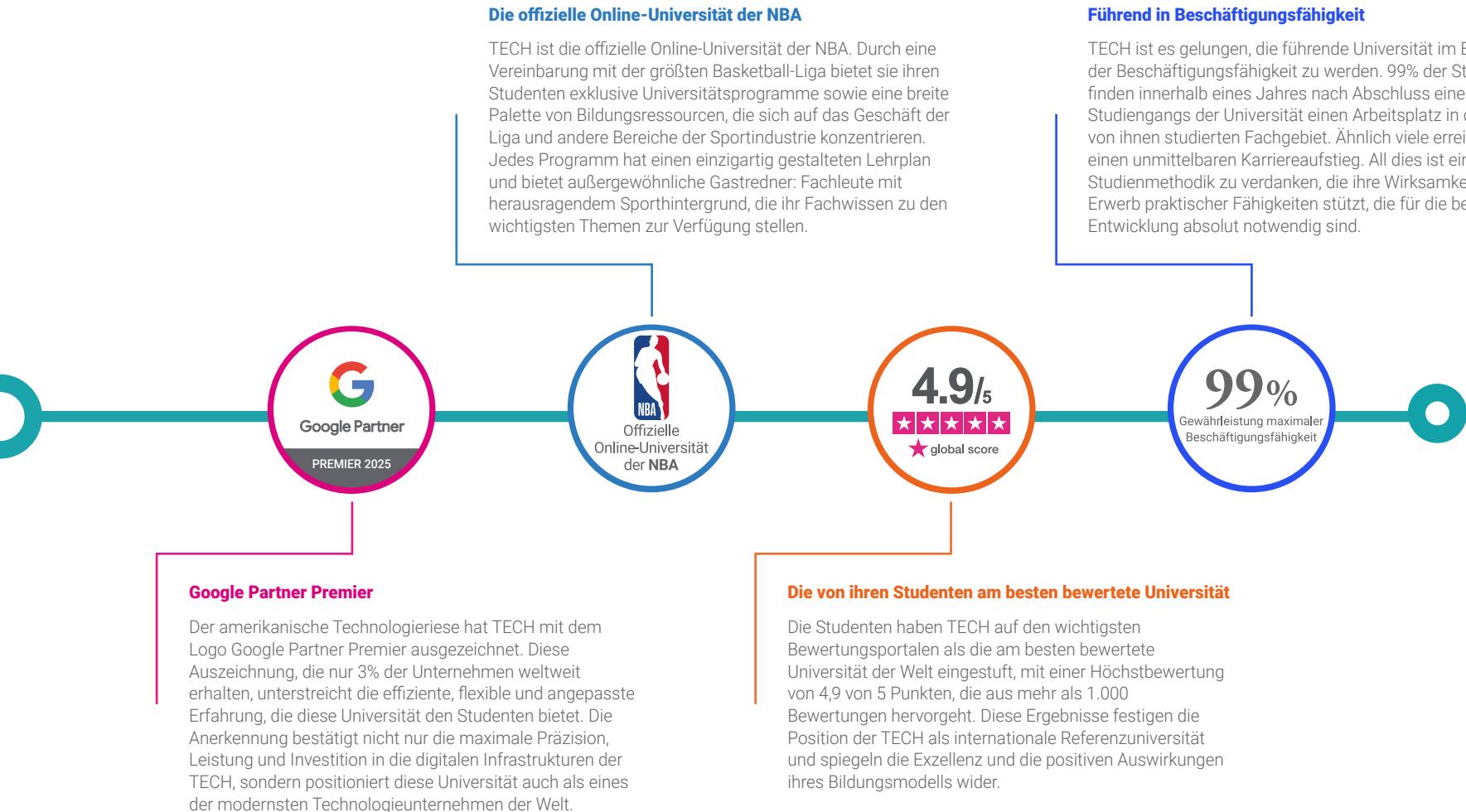
TECH ist die größte digitale Universität der Welt. Mit einem beeindruckenden Katalog von über 14.000 Hochschulprogrammen, die in 11 Sprachen angeboten werden, ist sie mit einer Vermittlungsquote von 99% führend im Bereich der Beschäftigungsfähigkeit. Darüber hinaus verfügt sie über einen beeindruckenden Lehrkörper mit mehr als 6.000 Professoren von höchstem internationalem Prestige.



“

Studieren Sie an der größten digitalen Universität der Welt und sichern Sie sich Ihren beruflichen Erfolg. Die Zukunft beginnt bei TECH”

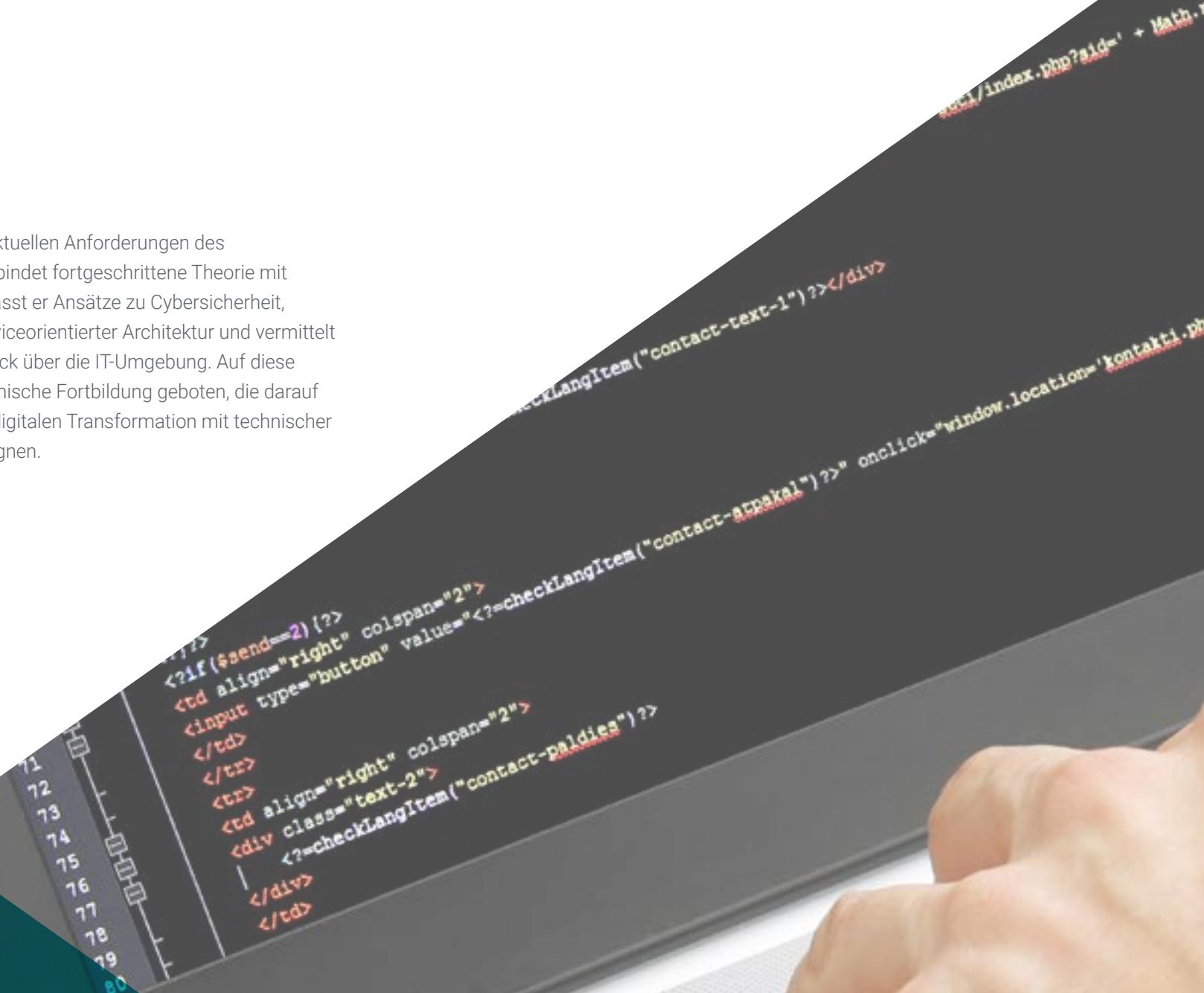




03

Lehrplan

Dieser Lehrplan umfasst Inhalte, die auf die aktuellen Anforderungen des Technologiesektors abgestimmt sind und verbindet fortgeschrittene Theorie mit praktischer Anwendung. Darüber hinaus umfasst er Ansätze zu Cybersicherheit, plattformübergreifender Entwicklung und serviceorientierter Architektur und vermittelt so einen umfassenden und modernen Überblick über die IT-Umgebung. Auf diese Weise wird eine fundierte und aktuelle akademische Fortbildung geboten, die darauf ausgerichtet ist, den Herausforderungen der digitalen Transformation mit technischer Kompetenz und innovativem Denken zu begegnen.



66

Sie werden die wesentlichen Grundlagen der Softwareentwicklung beherrschen, um solide, effiziente und an die Anforderungen des beruflichen Umfelds angepasste technologische Lösungen zu entwickeln"

Modul 1. Grundlagen der Programmierung

- 1.1. Einführung in die Programmierung
 - 1.1.1. Grundlegende Struktur eines Computers
 - 1.1.2. Software
 - 1.1.3. Programmiersprachen
 - 1.1.4. Lebenszyklus einer Softwareanwendung
- 1.2. Entwurf von Algorithmen
 - 1.2.1. Lösung von Problemen
 - 1.2.2. Deskriptive Techniken
 - 1.2.3. Elemente und Struktur eines Algorithmus
- 1.3. Elemente eines Programms
 - 1.3.1. Ursprung und Merkmale der Sprache C++
 - 1.3.2. Die Entwicklungsumgebung
 - 1.3.3. Konzept des Programms
 - 1.3.4. Arten von grundlegenden Daten
 - 1.3.5. Operatoren
 - 1.3.6. Ausdrücke
 - 1.3.7. Anweisungen
 - 1.3.8. Dateneingabe und -ausgabe
- 1.4. Kontrollstrukturen
 - 1.4.1. Anweisungen
 - 1.4.2. Verzweigungen
 - 1.4.3. Schleifen
- 1.5. Abstraktion und Modularität: Funktionen
 - 1.5.1. Modularer Aufbau
 - 1.5.2. Konzept der Funktion und des Nutzens
 - 1.5.3. Definition einer Funktion
 - 1.5.4. Ausführungsablauf beim Aufruf einer Funktion
 - 1.5.5. Prototyp einer Funktion
 - 1.5.6. Rückgabe der Ergebnisse
 - 1.5.7. Aufrufen einer Funktion: Parameter
 - 1.5.8. Übergabe von Parametern per Referenz und per Wert
 - 1.5.9. Gültigkeitsbereich





- 1.6. Statische Datenstrukturen
 - 1.6.1. Arrays
 - 1.6.2. Matrizen. Polyeder
 - 1.6.3. Suche und Sortierung
 - 1.6.4. Zeichenketten. E/A-Funktionen für Zeichenketten
 - 1.6.5. Strukturen. Verbindungen
 - 1.6.6. Neue Datentypen
- 1.7. Dynamische Datenstrukturen: Zeiger
 - 1.7.1. Konzept. Definition von Zeiger
 - 1.7.2. Operatoren und Operationen mit Zeigern
 - 1.7.3. Arrays von Zeigern
 - 1.7.4. Zeiger und Arrays
 - 1.7.5. Zeiger auf Zeichenketten
 - 1.7.6. Zeiger auf Strukturen
 - 1.7.7. Mehrfache Indirektion
 - 1.7.8. Zeiger auf Funktionen
 - 1.7.9. Übergabe von Funktionen, Strukturen und Arrays als Funktionsparameter
- 1.8. Dateien
 - 1.8.1. Grundlegende Konzepte
 - 1.8.2. Dateioperationen
 - 1.8.3. Datentypen
 - 1.8.4. Organisation von Dateien
 - 1.8.5. Einführung in C++-Dateien
 - 1.8.6. Handhabung von Dateien
- 1.9. Rekursion
 - 1.9.1. Definition von Rekursion
 - 1.9.2. Arten der Rekursion
 - 1.9.3. Vor- und Nachteile
 - 1.9.4. Überlegungen
 - 1.9.5. Umwandlung von Rekursion in Iteration
 - 1.9.6. Der Rekursionsstapel

- 1.10. Test und Dokumentation
 - 1.10.1. Programmtests
 - 1.10.2. White-Box-Tests
 - 1.10.3. Black-Box-Tests
 - 1.10.4. Werkzeuge zur Testdurchführung
 - 1.10.5. Programmdokumentation

Modul 2. Datenstruktur

- 2.1. Einführung in die Programmierung in C++
 - 2.1.1. Klassen, Konstruktoren, Methoden und Attribute
 - 2.1.2. Variablen
 - 2.1.3. Bedingte Ausdrücke und Schleifen
 - 2.1.4. Objekte
- 2.2. Abstrakte Datentypen (ADT)
 - 2.2.1. Datentypen
 - 2.2.2. Grundlegende Strukturen und ADTs
 - 2.2.3. Vektoren und Arrays
- 2.3. Lineare Datenstrukturen
 - 2.3.1. Definition der ADT-Liste
 - 2.3.2. Verknüpfte und doppelt verknüpfte Listen
 - 2.3.3. Geordnete Listen
 - 2.3.4. Listen in C++
 - 2.3.5. ADT-Stapel
 - 2.3.6. ADT-Warteschlange
 - 2.3.7. Stapel und Warteschlange in C++
- 2.4. Hierarchische Datenstrukturen
 - 2.4.1. ADT-Baum
 - 2.4.2. Traversierungen
 - 2.4.3. n-äre Bäume
 - 2.4.4. Binärbäume
 - 2.4.5. Binäre Suchbäume
- 2.5. Hierarchische Datenstrukturen: komplexe Bäume
 - 2.5.1. Perfekt balancierte oder minimal hohe Bäume
 - 2.5.2. Mehrwegbäume

- 2.5.3. Literaturverzeichnis
- 2.6. Heaps und Prioritätswarteschlange
 - 2.6.1. ADT-Heaps
 - 2.6.2. ADT-Prioritätswarteschlange
- 2.7. Hash-Tabellen
 - 2.7.1. ADT-Hash-Tabellen
 - 2.7.2. Hash-Funktionen
 - 2.7.3. Hash-Funktion in Hash-Tabellen
 - 2.7.4. Redispersion
 - 2.7.5. Offene Hash-Tabellen
- 2.8. Graphen
 - 2.8.1. ADT-Graph
 - 2.8.2. Arten von Graphen
 - 2.8.3. Grafische Darstellung und Grundoperationen
 - 2.8.4. Entwurf von Graphen
- 2.9. Algorithmen und weiterführende Konzepte zu Graphen
 - 2.9.1. Graph-Probleme
 - 2.9.2. Wege-Algorithmen
 - 2.9.3. Such- oder Traversierungsalgorithmen
 - 2.9.4. Andere Algorithmen
- 2.10. Andere Datenstrukturen
 - 2.10.1. Mengen
 - 2.10.2. Parallele Arrays
 - 2.10.3. Symboltabellen
 - 2.10.4. Tries

Modul 3. Algorithmen und Komplexität

- 3.1. Einführung in Entwurfsstrategien für Algorithmen
 - 3.1.1. Rekursion
 - 3.1.2. Teile-und-herrsche (*Divide and Conquer*)
 - 3.1.3. Andere Strategien
- 3.2. Effizienz und Analyse von Algorithmen
 - 3.2.1. Effizienzmaße
 - 3.2.2. Messung der Eingabegröße

- 3.2.3. Messung der Laufzeit
- 3.2.4. Schlimmster, bester und durchschnittlicher Fall
- 3.2.5. Asymptotische Notation
- 3.2.6. Kriterien für die mathematische Analyse von nichtrekursiven Algorithmen
- 3.2.7. Mathematische Analyse von rekursiven Algorithmen
- 3.2.8. Empirische Analyse von Algorithmen
- 3.3. Sorteralgorithmen
 - 3.3.1. Konzept der Sortierung
 - 3.3.2. Sortieren durch Aufsteigen
 - 3.3.3. Sortieren durch Auswählen
 - 3.3.4. Sortieren durch Einfügen
 - 3.3.5. Sortieren durch zusammenführen (*Merge Sort*)
 - 3.3.6. Schnelle Sortierung (*Quick Sort*)
- 3.4. Algorithmen mit Bäumen
 - 3.4.1. Konzept des Baums
 - 3.4.2. Binäräume
 - 3.4.3. Baumtraversierungen
 - 3.4.4. Darstellung von Ausdrücken
 - 3.4.5. Binäre Suchbäume
 - 3.4.6. Balancierte Binäräume
- 3.5. Algorithmen mit *Heaps*
 - 3.5.1. *Heaps*
 - 3.5.2. Der *HeapSort*-Algorithmus
 - 3.5.3. Prioritätswarteschlangen
- 3.6. Algorithmen mit Graphen
 - 3.6.1. Darstellung
 - 3.6.2. Breitensuche
 - 3.6.3. Tiefensuche
 - 3.6.4. Topologische Sortierung
- 3.7. Greedy-Algorithmen
 - 3.7.1. Die *Greedy*-Strategie
 - 3.7.2. Elemente der *Greedy*-Strategie
 - 3.7.3. Münzwechselproblem
 - 3.7.4. Problem des Handlungsreisenden
- 3.7.5. Rucksackproblem
- 3.8. Suche nach kürzesten Wegen
 - 3.8.1. Das Problem des kürzesten Weges
 - 3.8.2. Negative Kanten und Zyklen
 - 3.8.3. Dijkstra-Algorithmus
- 3.9. Greedy-Algorithmen auf Graphen
 - 3.9.1. Minimaler Spannbaum
 - 3.9.2. Prim-Algorithmus
 - 3.9.3. Kruskal-Algorithmus
 - 3.9.4. Komplexitätsanalyse
- 3.10. *Backtracking*
 - 3.10.1. Das *Backtracking*
 - 3.10.2. Alternative Techniken

Modul 4. Datenbanken

- 4.1. Anwendungen und Zwecke von Datenbanksystemen
 - 4.1.1. Anwendungen der verschiedenen Datenbanksysteme
 - 4.1.2. Zweck der verschiedenen Datenbanksysteme
 - 4.1.3. Datenübersicht
- 4.2. Datenbank und Architektur
 - 4.2.1. Relationale Datenbanken
 - 4.2.2. Das Design von Datenbanken
 - 4.2.3. Objektbasierte und semistrukturierte Datenbanken
 - 4.2.4. Datenspeicherung und Abfragen
 - 4.2.5. Transaktionsmanagement
 - 4.2.6. *Data Mining* und Datenanalyse
 - 4.2.7. Datenbankarchitektur
- 4.3. Das relationale Modell: Struktur, Operationen und erweiterte relationale Algebra
 - 4.3.1. Die Struktur von relationalen Datenbanken
 - 4.3.2. Grundlegende Operationen in der relationalen Algebra
 - 4.3.3. Andere Operationen der relationalen Algebra
 - 4.3.4. Erweiterte Operationen der relationalen Algebra
 - 4.3.5. Nullwerte
 - 4.3.6. Modifizierung der Datenbank

- 4.4. SQL (I)
 - 4.4.1. Was ist SQL?
 - 4.4.2. Definition der Daten
 - 4.4.3. Grundlegende Struktur der SQL-Abfragen
 - 4.4.4. Mengenoperationen
 - 4.4.5. Aggregatfunktionen
 - 4.4.6. Nullwerte
- 4.5. SQL (II)
 - 4.5.1. Verschachtelte Unterabfragen
 - 4.5.2. Komplexe Abfragen
 - 4.5.3. Ansichten
 - 4.5.4. Cursor
 - 4.5.5. Komplexe Abfragen
 - 4.5.6. Trigger
- 4.6. Datenbankdesign und das ER-Modell
 - 4.6.1. Überblick über den Entwurfsprozess
 - 4.6.2. Das *Entity-Relationship*-Modell
 - 4.6.3. Einschränkungen
- 4.7. *Entity-Relationship*-Diagramme
 - 4.7.1. *Entity-Relationship*-Diagramme
 - 4.7.2. Aspekte der Gestaltung von *Entity*-Beziehungen
 - 4.7.3. Gruppen von schwachen Entitäten
- 4.8. Das erweiterte Entity-Relationship-Modell
 - 4.8.1. Merkmale des erweiterten ER-Modells
 - 4.8.2. Entwurf einer Datenbank
 - 4.8.3. Reduktion auf relationale Schemata
- 4.9. Entwurf relationaler Datenbanken
 - 4.9.1. Merkmale eines guten relationalen Designs
 - 4.9.2. Atomare Domänen und die erste Normalform (1NF)
 - 4.9.3. Zerlegung durch funktionale Abhängigkeiten
 - 4.9.4. Theorie der funktionalen Abhängigkeiten
 - 4.9.5. Zerlegungsalgorithmen
 - 4.9.6. Zerlegung durch mehrwertige Abhängigkeiten
- 4.9.7. Weitere Normalformen
- 4.9.8. Entwurfsprozess für Datenbanken
- 4.10. NoSQL-Datenbanken
 - 4.10.1. Was sind NoSQL-Datenbanken?
 - 4.10.2. Analyse der verschiedenen NoSQL-Optionen und ihrer Eigenschaften
 - 4.10.3. MongoDB

Modul 5. Fortgeschrittene Datenbanken

- 5.1. Einführung in die verschiedenen Datenbanksysteme
 - 5.1.1. Historischer Überblick
 - 5.1.2. Hierarchische Datenbanken
 - 5.1.3. Netzwerkdatenbanken
 - 5.1.4. Relationale Datenbanken
 - 5.1.5. Nichtrelationale Datenbanken
- 5.2. XML und Datenbanken für das Web
 - 5.2.1. Validierung von XML-Dokumenten
 - 5.2.2. Transformationen von XML-Dokumenten
 - 5.2.3. Speicherung von XML-Daten
 - 5.2.4. Relationale XML-Datenbanken
 - 5.2.5. SQL/XML
 - 5.2.6. Native XML-Datenbanken
- 5.3. Parallele Datenbanken
 - 5.3.1. Parallele Systeme
 - 5.3.2. Parallele Datenbankarchitekturen
 - 5.3.3. Parallelität bei Abfragen
 - 5.3.4. Parallelität zwischen den Abfragen
 - 5.3.5. Entwurf paralleler Systeme
 - 5.3.6. Parallele Verarbeitung in SQL
- 5.4. Verteilte Datenbanken
 - 5.4.1. Verteilte Systeme
 - 5.4.2. Verteilte Speicherung
 - 5.4.3. Verfügbarkeit
 - 5.4.4. Verteilte Abfrageverarbeitung
 - 5.4.5. Anbieter verteilter Datenbanken

- 5.5. Indexierung und Assozierung
 - 5.5.1. Geordnete Indizes
 - 5.5.2. Dichte und verstreute Indizes
 - 5.5.3. Mehrstufige Indizes
 - 5.5.4. Aktualisierung des Indexes
 - 5.5.5. Statische Assoziation
 - 5.5.6. Wie verwendet man Indizes in Datenbanken?
- 5.6. Einführung in die Transaktionsverarbeitung
 - 5.6.1. Zustände einer Transaktion
 - 5.6.2. Implementierung von Atomizität und Dauerhaftigkeit
 - 5.6.3. Sequenzialität
 - 5.6.4. Wiederherstellbarkeit
 - 5.6.5. Implementierung der Isolation
- 5.7. Wiederherstellungssysteme
 - 5.7.1. Klassifizierung von Fehlern
 - 5.7.2. Speicherstrukturen
 - 5.7.3. Wiederherstellung und Atomizität
 - 5.7.4. Wiederherstellung auf Basis historischer Aufzeichnungen
 - 5.7.5. Parallel Transaktionen und Wiederherstellung
 - 5.7.6. Hohe Verfügbarkeit in Datenbanken
- 5.8. Ausführung und Bearbeitung von Abfragen
 - 5.8.1. Kosten einer Abfrage
 - 5.8.2. Auswahloperation
 - 5.8.3. Sortierung
 - 5.8.4. Einführung in die Abfrageoptimierung
 - 5.8.5. Leistungsüberwachung
- 5.9. Nichtrelationale Datenbanken
 - 5.9.1. Dokumentorientierte Datenbanken
 - 5.9.2. Graphorientierte Datenbanken
 - 5.9.3. Key-Value-Datenbanken
- 5.10. *Data Warehouse, OLAP und Data Mining*
 - 5.10.1. Komponenten von *Data Warehouses*
 - 5.10.2. Architektur eines *Data Warehouse*
 - 5.10.3. OLAP

- 5.10.4. *Data-Mining-Funktionen*
- 5.10.5. Andere Arten von *Mining*

Modul 6. Fortgeschrittenen Algorithmusentwurf

- 6.1. Analyse von rekursiven und *Divide-and-Conquer*-Algorithmen
 - 6.1.1. Aufstellen und Lösen von homogenen und nicht-homogenen Rekursionsgleichungen
 - 6.1.2. Allgemeine Beschreibung der *Divide-and-Conquer*-Strategie
- 6.2. Amortisierte Analyse
 - 6.2.1. Aggregierte Analyse
 - 6.2.2. Die Buchhaltungsmethode
 - 6.2.3. Die Potentialmethode
- 6.3. Dynamische Programmierung und Algorithmen für NP-Probleme
 - 6.3.1. Merkmale der dynamischen Programmierung
 - 6.3.2. Umkehr: *Backtracking*
 - 6.3.3. Verzweigung und Beschniedung
- 6.4. Kombinatorische Optimierung
 - 6.4.1. Problemdarstellung
 - 6.4.2. 1D-Optimierung
- 6.5. Randomisierte Algorithmen
 - 6.5.1. Beispiele für randomisierte Algorithmen
 - 6.5.2. Das Buffonsche Theorem
 - 6.5.3. Monte-Carlo-Algorithmus
 - 6.5.4. Las-Vegas-Algorithmus
- 6.6. Lokale Suche und Kandidatensuche
 - 6.6.1. *Gradient Ascent*
 - 6.6.2. *Hill Climbing*
 - 6.6.3. *Simulated Annealing*
 - 6.6.4. *Tabu Search*
 - 6.6.5. Kandidatensuche
- 6.7. Formale Verifizierung von Programmen
 - 6.7.1. Spezifikation von funktionalen Abstraktionen
 - 6.7.2. Die Sprache der Prädikatenlogik erster Stufe
 - 6.7.3. Hoare's formales System

- 6.8. Verifizierung von iterativen Programmen
 - 6.8.1. Regeln des formalen Hoare-Systems
 - 6.8.2. Konzept der invarianten Iterationen
- 6.9. Numerische Methoden
 - 6.9.1. Die Methode der Halbierung
 - 6.9.2. Die Newton-Raphson-Methode
 - 6.9.3. Die Sekantenmethode
- 6.10. Parallele Algorithmen
 - 6.10.1. Parallele binäre Operationen
 - 6.10.2. Parallele Operationen mit Diagrammen
 - 6.10.3. Parallelität in Teilen und Erobern
 - 6.10.4. Parallelität in der dynamischen Programmierung
- 7.4. Der menschliche Faktor: sensorische und physische Einschränkungen
 - 7.4.1. Funktionale Vielfalt, Behinderung und Beeinträchtigung
 - 7.4.2. Visuelle Vielfalt
 - 7.4.3. Akustische Vielfalt
 - 7.4.4. Kognitive Vielfalt
 - 7.4.5. Motorische Vielfalt
 - 7.4.6. Der Fall der digitalen Einwanderer
- 7.5. Der Designprozess (I): Anforderungsanalyse für die Gestaltung der Benutzeroberfläche
 - 7.5.1. Benutzerzentriertes Design
 - 7.5.2. Was ist eine Anforderungsanalyse?
 - 7.5.3. Sammeln von Informationen
 - 7.5.4. Analyse und Interpretation der Informationen
 - 7.5.5. Analyse der Benutzerfreundlichkeit und Zugänglichkeit
- 7.6. Der Designprozess (II): Prototyping und Aufgabenanalyse
 - 7.6.1. Konzeptioneller Entwurf
 - 7.6.2. Prototypenerstellung
 - 7.6.3. Hierarchische Aufgabenanalyse
- 7.7. Der Designprozess (III): Bewertung
 - 7.7.1. Bewertung im Designprozess: Ziele und Methoden
 - 7.7.2. Bewertungsmethoden ohne Benutzer
 - 7.7.3. Bewertungsmethoden mit Benutzern
 - 7.7.4. Bewertungsstandards und -normen
- 7.8. Barrierefreiheit: Definition und Leitlinien
 - 7.8.1. Barrierefreiheit und universelles Design
 - 7.8.2. WAL-Initiative und WCAG-Richtlinien
 - 7.8.3. WCAG-Richtlinien 2.0 und 2.1
- 7.9. Barrierefreiheit: Bewertung und funktionale Vielfalt
 - 7.9.1. Tools zur Bewertung der Barrierefreiheit im Web
 - 7.9.2. Barrierefreiheit und funktionale Vielfalt
- 7.10. Der Computer und die Interaktion: Peripheriegeräte und Zubehör
 - 7.10.1. Herkömmliche und Peripheriegeräte
 - 7.10.2. Alternative und Peripheriegeräte
 - 7.10.3. Handys und Tablets
 - 7.10.4. Funktionale Vielfalt, Interaktion und Peripheriegeräte

Modul 7. Mensch-Computer-Interaktion

- 7.1. Einführung in die Mensch-Computer Interaktion
 - 7.1.1. Was ist die Mensch-Computer Interaktion?
 - 7.1.2. Beziehung der Mensch-Computer-Interaktion zu anderen Disziplinen
 - 7.1.3. Die Benutzeroberfläche
 - 7.1.4. Benutzerfreundlichkeit und Zugänglichkeit
 - 7.1.5. Benutzererfahrung und benutzerzentriertes Design
- 7.2. Der Computer und die Interaktion: Benutzeroberfläche und Interaktionsparadigmen
 - 7.2.1. Interaktion
 - 7.2.2. Interaktionsparadigmen und Interaktionsstile
 - 7.2.3. Entwicklung von Benutzeroberflächen
 - 7.2.4. Klassische Benutzeroberflächen: WIMP/GUI, Befehle, Stimme, virtuelle Realität
 - 7.2.5. Innovative Benutzeroberflächen: mobil, tragbar, kollaborativ, BCI
- 7.3. Der menschliche Faktor: psychologische und kognitive Aspekte
 - 7.3.1. Die Bedeutung des menschlichen Faktors in der Interaktion
 - 7.3.2. Menschliche Informationsverarbeitung
 - 7.3.3. Die Eingabe und Ausgabe von Informationen: visuell, auditiv und taktil
 - 7.3.4. Wahrnehmung und Aufmerksamkeit
 - 7.3.5. Wissen und mentale Modelle: Darstellung, Organisation und Erwerb
- 7.4. Der menschliche Faktor: sensorische und physische Einschränkungen
 - 7.4.1. Funktionale Vielfalt, Behinderung und Beeinträchtigung
 - 7.4.2. Visuelle Vielfalt
 - 7.4.3. Akustische Vielfalt
 - 7.4.4. Kognitive Vielfalt
 - 7.4.5. Motorische Vielfalt
 - 7.4.6. Der Fall der digitalen Einwanderer
- 7.5. Der Designprozess (I): Anforderungsanalyse für die Gestaltung der Benutzeroberfläche
 - 7.5.1. Benutzerzentriertes Design
 - 7.5.2. Was ist eine Anforderungsanalyse?
 - 7.5.3. Sammeln von Informationen
 - 7.5.4. Analyse und Interpretation der Informationen
 - 7.5.5. Analyse der Benutzerfreundlichkeit und Zugänglichkeit
- 7.6. Der Designprozess (II): Prototyping und Aufgabenanalyse
 - 7.6.1. Konzeptioneller Entwurf
 - 7.6.2. Prototypenerstellung
 - 7.6.3. Hierarchische Aufgabenanalyse
- 7.7. Der Designprozess (III): Bewertung
 - 7.7.1. Bewertung im Designprozess: Ziele und Methoden
 - 7.7.2. Bewertungsmethoden ohne Benutzer
 - 7.7.3. Bewertungsmethoden mit Benutzern
 - 7.7.4. Bewertungsstandards und -normen
- 7.8. Barrierefreiheit: Definition und Leitlinien
 - 7.8.1. Barrierefreiheit und universelles Design
 - 7.8.2. WAL-Initiative und WCAG-Richtlinien
 - 7.8.3. WCAG-Richtlinien 2.0 und 2.1
- 7.9. Barrierefreiheit: Bewertung und funktionale Vielfalt
 - 7.9.1. Tools zur Bewertung der Barrierefreiheit im Web
 - 7.9.2. Barrierefreiheit und funktionale Vielfalt
- 7.10. Der Computer und die Interaktion: Peripheriegeräte und Zubehör
 - 7.10.1. Herkömmliche und Peripheriegeräte
 - 7.10.2. Alternative und Peripheriegeräte
 - 7.10.3. Handys und Tablets
 - 7.10.4. Funktionale Vielfalt, Interaktion und Peripheriegeräte

Modul 8. Fortgeschrittene Programmierung

- 8.1. Einführung in die objektorientierte Programmierung
 - 8.1.1. Einführung in die objektorientierte Programmierung
 - 8.1.2. Klassendesign
 - 8.1.3. Einführung in UML für die Modellierung von Problemen
- 8.2. Beziehungen zwischen Klassen
 - 8.2.1. Abstraktion und Vererbung
 - 8.2.2. Fortgeschrittene Konzepte der Vererbung
 - 8.2.3. Polymorphismen
 - 8.2.4. Zusammensetzung und Aggregation
- 8.3. Einführung in Entwurfsmuster für objektorientierte Probleme
 - 8.3.1. Was sind Entwurfsmuster
 - 8.3.2. *Factory*-Muster
 - 8.3.3. *Singleton*-Muster
 - 8.3.4. *Observer*-Muster
 - 8.3.5. *Composite*-Muster
- 8.4. Ausnahmen
 - 8.4.1. Was sind Ausnahmen?
 - 8.4.2. Auffangen und Behandlung von Ausnahmen
 - 8.4.3. Auslösen von Ausnahmen
 - 8.4.4. Erstellung von Ausnahmen
- 8.5. Benutzeroberflächen
 - 8.5.1. Einführung in Qt
 - 8.5.2. Positionierung
 - 8.5.3. Was sind Ereignisse?
 - 8.5.4. Ereignisse: Definition und Erfassung
 - 8.5.5. Entwicklung von Benutzeroberflächen
- 8.6. Einführung in die nebenläufige Programmierung
 - 8.6.1. Einführung in die nebenläufige Programmierung
 - 8.6.2. Das Konzept von Prozessen und *Threads*
 - 8.6.3. Interaktion zwischen Prozessen oder *Threads*
 - 8.6.4. *Threads* in C++
 - 8.6.5. Vor- und Nachteile der nebenläufigen Programmierung

8.7. *Thread*-Verwaltung und Synchronisation

- 8.7.1. Lebenszyklus eines *Threads*
- 8.7.2. Die Klasse Thread
- 8.7.3. *Thread*-Planung
- 8.7.4. *Thread*-Gruppen
- 8.7.5. Daemon-*Threads*
- 8.7.6. Synchronisation
- 8.7.7. Sperrmechanismen
- 8.7.8. Kommunikationsmechanismen
- 8.7.9. Monitore
- 8.8. Häufige Probleme in der nebenläufigen Programmierung
 - 8.8.1. Das Erzeuger-Verbraucher-Problem
 - 8.8.2. Das Leser-Schreiber-Problem
 - 8.8.3. Das Problem der speisenden Philosophen
- 8.9. Softwaredokumentation und -tests
 - 8.9.1. Warum ist es wichtig, Software zu dokumentieren?
 - 8.9.2. Entwurfsdokumentation
 - 8.9.3. Verwendung von Tools zur Dokumentation
- 8.10. Softwaretests
 - 8.10.1. Einführung in Softwaretests
 - 8.10.2. Arten von Tests
 - 8.10.3. Unit-Test
 - 8.10.4. Integrationstests
 - 8.10.5. Validierungstest
 - 8.10.6. Systemprüfung

Modul 9. Entwicklung von Netzwerkanwendungen

- 9.1. Auszeichnungssprachen – HTML5
 - 9.1.1. Grundlegende Konzepte der HTML
 - 9.1.2. Neue HTML 5-Elemente
 - 9.1.3. Formulare: neue Steuerelemente
- 9.2. Einführung in CSS-Stylesheets
 - 9.2.1. Erste Schritte mit CSS
 - 9.2.2. Einführung in CSS3

- 9.3. Browser-Skriptsprache: JavaScript
 - 9.3.1. Grundkonzepte von JavaScript
 - 9.3.2. DOM
 - 9.3.3. Ereignisse
 - 9.3.4. jQuery
 - 9.3.5. Ajax
- 9.4. Konzept der komponentenorientierten Programmierung
 - 9.4.1. Kontext
 - 9.4.2. Komponenten und Schnittstellen
 - 9.4.3. Zustände einer Komponente
- 9.5. Architektur von Komponenten
 - 9.5.1. Zeitgenössische Architekturen
 - 9.5.2. Integration und Bereitstellung von Komponenten
- 9.6. *Frontend-Framework: Bootstrap*
 - 9.6.1. Rasterbasiertes Design
 - 9.6.2. Formulare
 - 9.6.3. Komponenten
- 9.7. *Model-View-Controller*
 - 9.7.1. Methoden der Webentwicklung
 - 9.7.2. Entwurfsmuster: MVC
- 9.8. Grid-Informationstechnologien
 - 9.8.1. Erhöhung der Rechenressourcen
 - 9.8.2. Konzept der Grid-Technologie
- 9.9. Serviceorientierte Architektur
 - 9.9.1. SOA und Webdienste
 - 9.9.2. Topologie eines Webdienstes
 - 9.9.3. Plattformen für Webdienste
- 9.10. HTTP-Protokoll
 - 9.10.1. Nachrichten
 - 9.10.2. Persistente Sitzungen
 - 9.10.3. Kryptographisches System
 - 9.10.4. Funktionsweise des HTTPS-Protokolls

Modul 10. Softwaretechnik

- 10.1. Einführung in die Softwaretechnik und -modellierung
 - 10.1.1. Die Natur der Software
 - 10.1.2. Die einzigartige Natur der WebApps
 - 10.1.3. Softwaretechnik
 - 10.1.4. Der Softwareentwicklungsprozess
 - 10.1.5. Die Praxis der Softwareentwicklung
 - 10.1.6. Mythen über Software
 - 10.1.7. Wie alles beginnt
 - 10.1.8. Objektorientierte Konzepte
 - 10.1.9. Einführung in UML
- 10.2. Der Softwareentwicklungsprozess
 - 10.2.1. Ein generisches Prozessmodell
 - 10.2.2. Vorgegebene Prozessmodelle
 - 10.2.3. Spezialisierte Prozessmodelle
 - 10.2.4. Der vereinheitlichte Prozess
 - 10.2.5. Persönliche und Teamprozessmodelle
 - 10.2.6. Was ist Agilität?
 - 10.2.7. Was ist ein agiler Prozess?
 - 10.2.8. Scrum
 - 10.2.9. Werkzeuge für agile Prozesse
- 10.3. Grundsätze für die Praxis der Softwareentwicklung
 - 10.3.1. Prinzipien zur Prozessführung
 - 10.3.2. Prinzipien als Leitfaden für die Praxis
 - 10.3.3. Kommunikationsprinzipien
 - 10.3.4. Planungsprinzipien
 - 10.3.5. Modellierungsprinzipien
 - 10.3.6. Konstruktionsprinzipien
 - 10.3.7. Prinzipien der Bereitstellung
- 10.4. Verständnis der Anforderungen
 - 10.4.1. Anforderungsmanagement
 - 10.4.2. Schaffung der Grundlagen
 - 10.4.3. Bedarfsermittlung

- 10.4.4. Entwicklung von Anwendungsfällen
- 10.4.5. Ausarbeitung des Anforderungsmodells
- 10.4.6. Aushandeln von Anforderungen
- 10.4.7. Validierung der Anforderungen
- 10.5. Anforderungsmodellierung: Szenarien, Information und Analyseklassen
 - 10.5.1. Analyse der Anforderungen
 - 10.5.2. Szenarienbasierte Modellierung
 - 10.5.3. UML-Modelle, die den Anwendungsfall liefern
 - 10.5.4. Konzepte der Datenmodellierung
 - 10.5.5. Klassenbasierte Modellierung
 - 10.5.6. Klassendiagramme
- 10.6. Anforderungsmodellierung: Fluss, Verhalten und Muster
 - 10.6.1. Strategieorientierte Anforderungen
 - 10.6.2. Flussorientierte Modellierung
 - 10.6.3. Zustandsdiagramme
 - 10.6.4. Erstellung eines Verhaltensmodells
 - 10.6.5. Sequenzdiagramme
 - 10.6.6. Kommunikationsdiagramme
 - 10.6.7. Muster zur Anforderungsmodellierung
- 10.7. Designkonzepte
 - 10.7.1. Design im Kontext der Softwaretechnik
 - 10.7.2. Der Designprozess
 - 10.7.3. Designkonzepte
 - 10.7.4. Konzepte des objektorientierten Designs
 - 10.7.5. Das Designmodell
- 10.8. Design der Architektur
 - 10.8.1. Softwarearchitektur
 - 10.8.2. Architektonische Gattungen
 - 10.8.3. Architekturstile
 - 10.8.4. Architektonisches Design
 - 10.8.5. Entwicklung von alternativen Designs für die Architektur
 - 10.8.6. Abbildung der Architektur mit Hilfe von Datenflüssen
- 10.9. Komponentendesign und musterbasiertes Design
 - 10.9.1. Was ist eine Komponente?
 - 10.9.2. Klassenbasiertes Komponentendesign
 - 10.9.3. Umsetzung des Komponentendesigns
 - 10.9.4. Traditionelles Komponentendesign
 - 10.9.5. Komponentenbasierte Entwicklung
 - 10.9.6. Entwurfsmuster
 - 10.9.7. Musterbasierte Softwaredesign
 - 10.9.8. Architekturmuster
 - 10.9.9. Entwurfsmuster auf Komponentenebene
 - 10.9.10. Entwurfsmuster für Benutzeroberflächen
- 10.10. Softwarequalität und Projektmanagement
 - 10.10.1. Softwarequalität
 - 10.10.2. Das Dilemma der Softwarequalität
 - 10.10.3. Erreichen von Softwarequalität
 - 10.10.4. Qualitätssicherung der Software
 - 10.10.5. Das administrative Spektrum
 - 10.10.6. Das Personal
 - 10.10.7. Das Produkt
 - 10.10.8. Der Prozess
 - 10.10.9. Das Projekt
 - 10.10.10. Grundsätze und Praktiken

“

Sie werden Kriterien hinsichtlich Qualität, Leistung und Nachhaltigkeit während des gesamten Lebenszyklus der Software anwenden“

04

Lehrziele

Die Lehrziele dieses Programms entsprechen den Anforderungen eines sich ständig weiterentwickelnden digitalen Ökosystems. Es geht nicht nur darum, fortgeschrittene technische Kompetenzen zu entwickeln, sondern auch kritisches Denken, agile Problemlösung und Innovationsfähigkeit zu fördern. Durch einen multidisziplinären und aktuellen Ansatz sollen Fachkräfte ausgebildet werden, die in der Lage sind, Technologieprojekte zu leiten, sich an neue Tools anzupassen und neue Methoden effizient anzuwenden. Darüber hinaus wird ein tiefgreifendes Verständnis des gesamten Entwicklungszyklus gefördert, wobei wichtige Aspekte wie Sicherheit, Skalierbarkeit und Nachhaltigkeit von Software in hochgradig wettbewerbsorientierten und dynamischen Umgebungen integriert werden.

77
78

79 <? if(\$_COOKIE['lang'] =='eng') {
80 echo "Wood-frame houses";
81 } else {
82 echo "Holzrahmenhäuser";
83 } </?>

84

85

86

87

88

89

90

91

92 <? if(\$_GET[type]==1){
93 echo "foto-galerija.php?";
94 } <?>

95 <div id="left_sidebar">
96 <div id="left_ico">

97 <p <?if(\$_COOKIE['la
98 <?>
99 </p> </div> </div>

100 <div id="content">
101 <div id="content_top">

102 <div id="content_main">
103 <div id="content_main_top">

104 <div id="content_main_bottom">
105 </div> </div> </div>

66

Spezialisieren Sie sich auf die Gestaltung
benutzerorientierter digitaler Erlebnisse unter
Verwendung von Techniken der Mensch-
Computer-Interaktion und Barrierefreiheit"



```
| !$_GET['type']) echo "erlaubt";  
type=1{text_margin:>  
</div>  
ang'!>--!300!>66
```



Allgemeine Ziele

- Entwickeln fortgeschritten technischer Kompetenzen in den Bereichen Programmierung, Design, Softwarearchitektur und Vereinigung verbundener Methodiken
- Anwenden agiler Methoden und kollaborativer Entwicklungsansätze in Technologieprojekten unterschiedlicher Komplexität
- Integrieren von Automatisierungs-, Versionskontroll- und Testtools in modernen professionellen Entwicklungsumgebungen
- Analysieren und Entwerfen skalierbarer, sicherer, effizienter und nachhaltiger Lösungen in verschiedenen digitalen Kontexten und Sektoren
- Beherrschen von plattformübergreifenden Entwicklungsumgebungen und neuen Technologien für das Design digitaler Lösungen
- Verstehen des gesamten Software-Lebenszyklus und seiner Auswirkungen auf aktuelle Geschäftsprozesse
- Bewerten der technischen und wirtschaftlichen Machbarkeit von Technologieentwicklungsprojekten in verschiedenen beruflichen Umgebungen
- Fördern der Innovationsfähigkeit beim Design komplexer digitaler Produkte, Dienstleistungen und Systeme





Spezifische Ziele

Modul 1. Grundlagen der Programmierung

- Verstehen der Grundprinzipien der strukturierten und objektorientierten Programmierung
- Identifizieren und Anwenden von Kontrollstrukturen, Variablen und Operatoren in verschiedenen Sprachen
- Entwickeln einfacher Algorithmen unter Verwendung bewährter Codierungspraktiken
- Interpretieren häufiger Fehler und effizientes Debuggen von Programmen

Modul 2. Datenstruktur

- Analysieren der Nützlichkeit von Datenstrukturen bei der Lösung von Computerproblemen
- Implementieren von Listen, Stapeln, Warteschlangen, Bäumen und Graphen in bestimmten Kontexten
- Bewerten der Effizienz verschiedener Strukturen entsprechend der erforderlichen Verwendung
- Anwenden geeigneter Datenstrukturen beim Entwerfen von Algorithmen

Modul 3. Algorithmen und Komplexität

- Entwerfen effizienter Algorithmen unter Berücksichtigung ihrer Rechenkosten
- Vergleichen der Leistung von Algorithmen mithilfe der Big-O-Notation
- Anwenden von Analysetechniken zur Identifizierung von Engpässen
- Optimieren von Algorithmen zur Verbesserung ihrer Leistung bei großen Datenmengen

Modul 4. Datenbanken

- Entwerfen von relationalen Modellen unter Anwendung von Normalisierungsprinzipien
- Implementieren von Datenbanken unter Verwendung herkömmlicher Verwaltungssysteme
- Erstellen von SQL-Abfragen zum Extrahieren, Einfügen und Ändern von Informationen
- Verstehen der Grundlagen von Integrität, Transaktionen und Parallelität

Modul 5. Fortgeschrittene Datenbanken

- Anwenden von Techniken zur Abfrageoptimierung in komplexen Umgebungen
- Verwalten von verteilten und nicht relationalen Datenbanken
- Einsetzen von gespeicherten Prozeduren, *Triggern* und Ansichten in realen Szenarien
- Analysieren von *Big-Data*-orientierten Datenbanklösungen

Modul 6. Fortgeschrittener Algorithmusentwurf

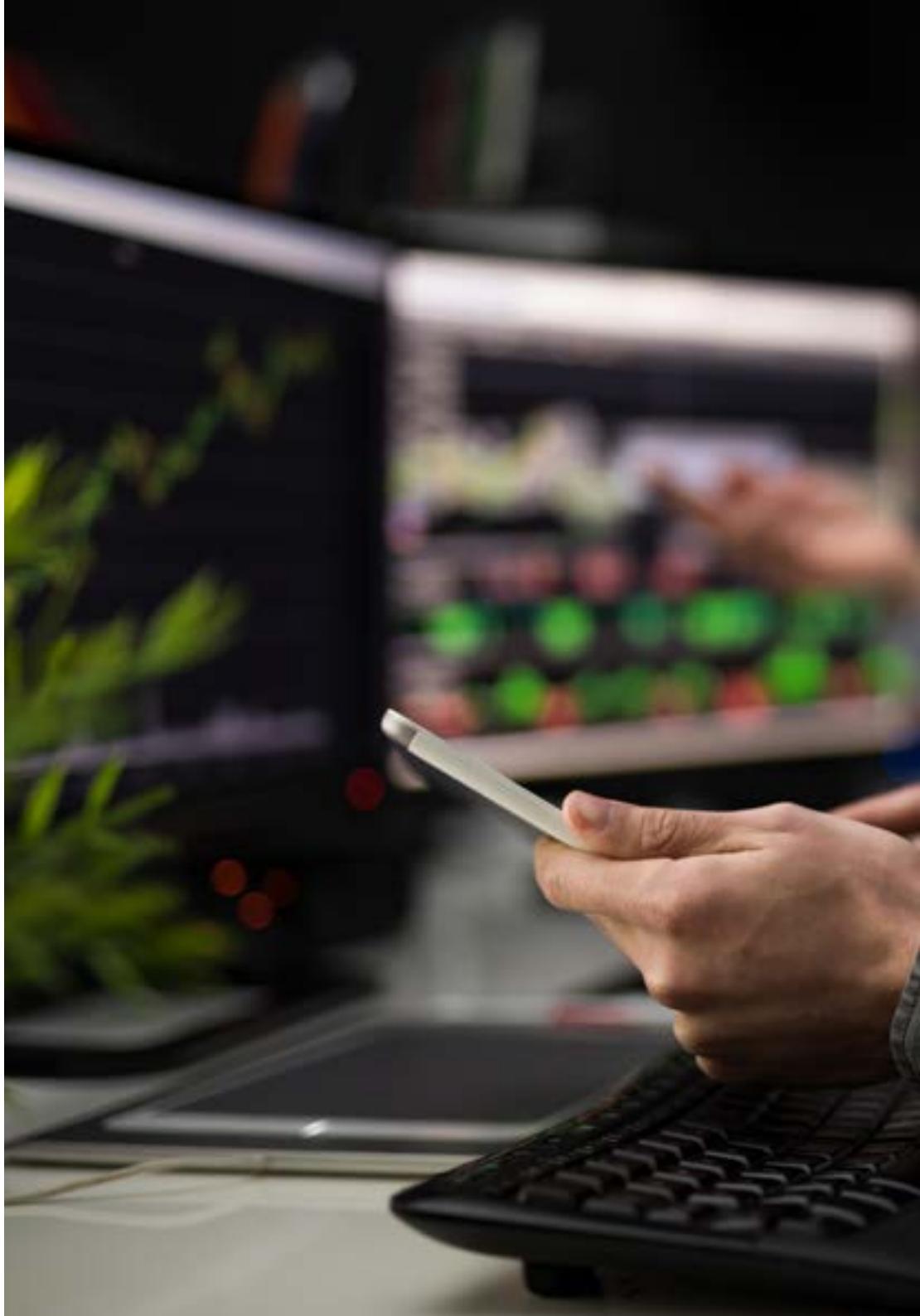
- Entwickeln von Techniken wie „Aufteilen und Erobern“, dynamische Programmierung und *Backtracking*
- Lösen komplexer Probleme mithilfe klassischer und heuristischer Algorithmen
- Identifizieren NP-vollständiger Probleme und Bewerten möglicher Lösungen
- Implementieren effizienter Algorithmen, die an konkrete Anforderungen angepasst sind

Modul 7. Mensch-Computer-Interaktion

- Entwerfen von Schnittstellen, die auf die Benutzererfahrung ausgerichtet sind
- Anwenden von Prinzipien der Benutzerfreundlichkeit und Barrierefreiheit in interaktiven Systemen
- Bewerten von Prototypen durch Benutzertests und Interaktionsmetriken
- Integrieren von Schnittstellendesignmustern in reale Anwendungen

Modul 8. Fortgeschrittene Programmierung

- Implementieren komplexer Anwendungen unter Verwendung moderner Programmietechniken
- Anwenden von Konzepten wie Threads, dynamischem Speicher und kontrollierten Ausnahmen
- Entwickeln modularer, wiederverwendbarer und skalierbarer Software
- Integrieren externer Bibliotheken und Tools in Entwicklungsprojekte



Modul 9. Entwicklung von Netzwerkanwendungen

- ♦ Entwerfen und Implementieren von Client-Server-Anwendungen unter Verwendung von Standardprotokollen
- ♦ Verwalten von Verbindungen, Parallelität und Sicherheit in verteilten Umgebungen
- ♦ Anwenden von Techniken zur Serialisierung und Kommunikation zwischen Remote-Prozessen
- ♦ Einsetzen von Tools zur Entwicklung und zum Testen von Netzwerkdiensten

Modul 10. Softwaretechnik

- ♦ Anwenden von Entwicklungsmethoden wie Scrum, XP und DevOps
- ♦ Verwalten des Software-Lebenszyklus von der Planung bis zur Wartung
- ♦ Erstellen klarer und strukturierter technischer Dokumentationen für Softwareprojekte
- ♦ Koordinieren von Teams und Tools für die kollaborative Entwicklung komplexer Systeme

“

Sie werden sich durch Ihr Engagement für Qualität, Codesicherheit und ethische Softwareentwicklung auszeichnen“

05

Karrieremöglichkeiten

Die Softwareentwicklung hat sich als eine der Säulen der digitalen Wirtschaft etabliert und die Nachfrage nach spezialisierten Profilen in zahlreichen Branchen angekurbelt. Somit eröffnet dieses Programm berufliche Möglichkeiten in Bereichen wie Anwendungsentwicklung, Systemarchitektur, Cybersicherheit oder Datenanalyse. Darüber hinaus ermöglichen die erworbenen Kompetenzen dank der ständigen technologischen Weiterentwicklung und der globalen digitalen Transformation den Zugang zu strategischen Positionen in Technologieunternehmen, Startups und internationalen Institutionen. Es handelt sich also um einen Bereich mit hoher Beschäftigungsfähigkeit, internationaler Ausrichtung und realer Einflussmöglichkeit auf unternehmerische Innovationen.



“

*Fördern Sie Ihr berufliches Profil in
Richtung strategischer Positionen wie
DevOps-Ingenieur, Softwarearchitekt
oder Systemanalytiker”*

Profil des Absolventen

Das Profil des Absolventen dieses Universitätsprogramms von TECH zeichnet sich durch eine solide Kombination aus technischen Fähigkeiten, analytischem Denken und der Fähigkeit aus, sich an veränderte technologische Umgebungen anzupassen. Dank einer ganzheitlichen Sichtweise auf die Softwareentwicklung ist er in der Lage, effiziente, skalierbare und sichere digitale Lösungen zu entwerfen, zu implementieren und zu optimieren. Darüber hinaus beherrscht er aktuelle Tools und Sprachen, arbeitet mit agilen Methoden und versteht die Prinzipien der modernen Softwareentwicklung. Diese Vielseitigkeit ermöglicht es ihm, sich erfolgreich in multidisziplinäre Teams zu integrieren, innovative Projekte zu leiten und effektiv auf die Herausforderungen eines hart umkämpften globalen Marktes zu reagieren.

Möchten Sie als Softwarearchitekt tätig sein?

Dieser Lehrplan vermittelt Ihnen die Schlüssel dazu, dies in nur wenigen Monaten zu erreichen.

- **Logisches und strukturiertes Denken:** Entscheidend für die Entwicklung und Optimierung komplexer Algorithmen
- **Abstraktionsfähigkeit:** Fähigkeit, technologische Lösungen für reale Probleme zu modellieren
- **Angewandte Innovation:** Kompetenz in der Entwicklung digitaler Produkte mit Mehrwert
- **Digitale Verantwortung und Berufsethik:** Geschick im Umgang mit und in der Entwicklung von Technologien mit gesellschaftlicher Wirkung





Nach Abschluss des Studiengangs werden Sie in der Lage sein, Ihre Kenntnisse und Fähigkeiten in den folgenden Positionen anzuwenden:

- 1. Softwareentwickler:** Entwirft, programmiert und testet Anwendungen, die auf die Bedürfnisse von Unternehmen und Anwendern in verschiedenen technologischen Umgebungen zugeschnitten sind.
- 2. Softwareingenieur:** Plant, entwickelt und wartet komplexe Systeme unter Anwendung von Prinzipien der Qualität, Skalierbarkeit und Sicherheit.
- 3. Softwarearchitekt:** Definiert die technische Struktur von Anwendungen und koordiniert wichtige Entscheidungen, um deren Effizienz und Nachhaltigkeit sicherzustellen.
- 4. Systemanalytiker:** Bewertet Anforderungen, entwirft digitale Lösungen und fungiert als Bindeglied zwischen technischen Teams und Geschäftszielen.
- 5. DevOps-Ingenieur:** Automatisiert Entwicklungsprozesse und kontinuierliche Integration und optimiert so den Lebenszyklus der Software.
- 6. Backend-Entwickler:** Ist auf die interne Logik von Systemen spezialisiert und verwaltet Datenbanken, Server und Dienste.
- 7. Full-Stack-Entwickler:** Beherrscht sowohl die Frontend- als auch die Backend-Entwicklung und ist an allen Phasen eines digitalen Projekts beteiligt.
- 8. QA- und Testing-Spezialist:** Stellt die Qualität der Software durch Funktions-, automatisierte und Leistungstests sicher.

“

Sie werden automatisierte Tests für die Validierung von Software in Integrationsumgebungen erstellen und so eine optimale Bereitstellung gewährleisten“

06

Inbegriffene Softwarelizenzen

TECH ist eine Referenz in der Universitätswelt, da sie die neueste Technologie mit Lehrmethoden kombiniert, um den Lehr- und Lernprozess zu optimieren. Zu diesem Zweck hat sie ein Netzwerk von Partnerschaften aufgebaut, das ihr Zugang zu den fortschrittlichsten Software-Tools der Berufswelt ermöglicht.



“

Bei der Einschreibung erhalten Sie völlig kostenlos die Zugangsdaten für die akademische Nutzung der folgenden professionellen Softwareanwendungen“

TECH hat ein Netzwerk professioneller Partnerschaften aufgebaut, dem die wichtigsten Anbieter von Software für verschiedene Berufsbereiche angehören. Diese Partnerschaften ermöglichen TECH den Zugang zu Hunderten von Computeranwendungen und Softwarelizenzen, um diese ihren Studenten zur Verfügung zu stellen.

Die Softwarelizenzen für akademische Zwecke ermöglichen es den Studenten, die fortschrittlichsten Computeranwendungen in ihrem Berufsfeld zu nutzen, sodass sie diese kennenlernen und beherrschen lernen können, ohne dass ihnen Kosten entstehen. TECH übernimmt die Formalitäten für den Erwerb der Lizenzen, sodass die Studenten diese während der gesamten Dauer ihres Studiums im Rahmen des Programms Weiterbildender Masterstudiengang in Softwareentwicklung unbegrenzt und völlig kostenlos nutzen können.

TECH gewährt Ihnen kostenlosen Zugang zu folgenden Softwareanwendungen:



Google Career Launchpad

Google Career Launchpad ist eine Lösung zur Entwicklung digitaler Kompetenzen in den Bereichen Technologie und Datenanalyse. Mit einem geschätzten Wert von **5.000 US-Dollar** ist sie **kostenlos** im Hochschulprogramm von TECH enthalten und bietet Zugang zu interaktiven Labors und branchenweit anerkannten Zertifizierungen.

Diese Plattform kombiniert technische Ausbildung mit praktischen Fallbeispielen unter Verwendung von Technologien wie BigQuery und Google AI. Sie bietet simulierte Umgebungen zum Experimentieren mit realen Daten sowie ein Netzwerk von Experten für individuelle Beratung.

Wichtigste Funktionen:

- ◆ **Spezialisierte Kurse:** aktuelle Inhalte zu Cloud Computing, maschinellem Lernen und Datenanalyse
- ◆ **Live-Labore:** praktische Übungen mit echten Google Cloud-Tools ohne zusätzliche Konfiguration
- ◆ **Integrierte Zertifizierungen:** Vorbereitung auf international anerkannte Prüfungen
- ◆ **Professionelles Mentoring:** Sitzungen mit Experten von Google und Technologiepartnern
- ◆ **Kollaborative Projekte:** Herausforderungen basierend auf realen Problemen führender Unternehmen

Zusammenfassend lässt sich sagen, dass **Google Career Launchpad** Nutzer mit den neuesten Technologien auf dem Markt verbindet und ihnen den Einstieg in Bereiche wie künstliche Intelligenz und Datenwissenschaft mit branchenweit anerkannten Qualifikationen erleichtert.

```
require 'spec_helper'

# This file is copied to spec/ when you run 'rails generate rspec:install'
require 'rspec/rails'
require 'rspec/expectations'
require 'rspec/matchers'
require 'rspec/autorun'

# Requires supporting files with custom matchers and helpers
# in spec/support/ and its subdirectories
# in _spec.rb will both be run
# run twice. It is recommended that you
# end with _spec.rb. You can
# run your tests with
# $ rake rspec:install
# $ rake rspec
# $ rails generate rspec:install
```

DBeaver Enterprise Edition

DBeaver Enterprise Edition ist die professionelle Version des renommierten Datenbankmanagers DBeaver mit einem Verkaufspreis von ca. **250 Euro** pro Jahr. Während des Universitätsprogramms an der TECH wird es kostenlos angeboten, sodass die Studenten Daten in komplexen Umgebungen professionell und sicher verwalten, entwickeln und analysieren können.

Diese Plattform befähigt die Studenten der TECH, die Verwaltung relationaler und nicht-relationaler Datenbanken zu optimieren, intelligente SQL-Abfragen zu generieren, fortgeschrittene Schemata zu entwerfen und Informationen mit interaktiven Grafiken zu visualisieren. Darüber hinaus integriert sie Funktionen zur Unternehmensanalyse durch die Verbindung mit *Business-Intelligence*-Tools und wandelt Daten in strategisches Wissen für Entscheidungen um.

Wichtigste Funktionen

- **Umfassende Kompatibilität:** unterstützt Oracle, SQL Server, PostgreSQL, MongoDB, Cassandra und mehr
 - **Erweiterter SQL-Editor:** Autovervollständigung, Debugging und intelligente Assistenten
 - **Datenvisualisierung:** interaktive *Dashboards* und integrierte Grafiken
 - **Integration mit Tableau:** direkte Verbindung mit *Business-Intelligence*-Tools
 - **Schema-Design:** ERD-Bearbeitung und Reverse Engineering
 - **Umfassende Verwaltung:** *Backup*, Wiederherstellung, Vergleich und Benutzerverwaltung

Zusammenfassend lässt sich sagen, dass **DBeaver Enterprise Edition** den Studenten der TECH dabei unterstützt, Datenmanagement präzise, effizient und innovativ zu beherrschen.

07

Studienmethodik

TECH ist die erste Universität der Welt, die die Methodik der **case studies** mit **Relearning** kombiniert, einem 100%igen Online-Lernsystem, das auf geführten Wiederholungen basiert.

Diese disruptive pädagogische Strategie wurde entwickelt, um Fachleuten die Möglichkeit zu bieten, ihr Wissen zu aktualisieren und ihre Fähigkeiten auf intensive und gründliche Weise zu entwickeln. Ein Lernmodell, das den Studenten in den Mittelpunkt des akademischen Prozesses stellt und ihm die Hauptrolle zuweist, indem es sich an seine Bedürfnisse anpasst und die herkömmlichen Methoden beiseitelässt.



“

TECH bereitet Sie darauf vor, sich neuen Herausforderungen in einem unsicheren Umfeld zu stellen und in Ihrer Karriere erfolgreich zu sein”

Der Student: die Priorität aller Programme von TECH

Bei der Studienmethodik von TECH steht der Student im Mittelpunkt.

Die pädagogischen Instrumente jedes Programms wurden unter Berücksichtigung der Anforderungen an Zeit, Verfügbarkeit und akademische Genauigkeit ausgewählt, die heutzutage nicht nur von den Studenten, sondern auch von den am stärksten umkämpften Stellen auf dem Markt verlangt werden.

Beim asynchronen Bildungsmodell von TECH entscheidet der Student selbst, wie viel Zeit er mit dem Lernen verbringt und wie er seinen Tagesablauf gestaltet, und das alles bequem von einem elektronischen Gerät seiner Wahl aus. Der Student muss nicht an Präsenzveranstaltungen teilnehmen, die er oft nicht wahrnehmen kann. Die Lernaktivitäten werden nach eigenem Ermessen durchgeführt. Er kann jederzeit entscheiden, wann und von wo aus er lernen möchte.

“

*Bei TECH gibt es KEINE Präsenzveranstaltungen
(an denen man nie teilnehmen kann)“*





Die international umfassendsten Lehrpläne

TECH zeichnet sich dadurch aus, dass sie die umfassendsten Studiengänge im universitären Umfeld anbietet. Dieser Umfang wird durch die Erstellung von Lehrplänen erreicht, die nicht nur die wesentlichen Kenntnisse, sondern auch die neuesten Innovationen in jedem Bereich abdecken.

Durch ihre ständige Aktualisierung ermöglichen diese Programme den Studenten, mit den Veränderungen des Marktes Schritt zu halten und die von den Arbeitgebern am meisten geschätzten Fähigkeiten zu erwerben. Auf diese Weise erhalten die Studenten, die ihr Studium bei TECH absolvieren, eine umfassende Vorbereitung, die ihnen einen bedeutenden Wettbewerbsvorteil verschafft, um in ihrer beruflichen Laufbahn voranzukommen.

Und das von jedem Gerät aus, ob PC, Tablet oder Smartphone.

“

Das Modell der TECH ist asynchron, d. h. Sie können an Ihrem PC, Tablet oder Smartphone studieren, wo immer Sie wollen, wann immer Sie wollen und so lange Sie wollen“

Case studies oder Fallmethode

Die Fallmethode ist das am weitesten verbreitete Lernsystem an den besten Wirtschaftshochschulen der Welt. Sie wurde 1912 entwickelt, damit Studenten der Rechtswissenschaften das Recht nicht nur auf der Grundlage theoretischer Inhalte erlernten, sondern auch mit realen komplexen Situationen konfrontiert wurden. Auf diese Weise konnten sie fundierte Entscheidungen treffen und Werturteile darüber fällen, wie diese zu lösen sind. Sie wurde 1924 als Standardlehrmethode in Harvard etabliert.

Bei diesem Lehrmodell ist es der Student selbst, der durch Strategien wie *Learning by doing* oder *Design Thinking*, die von anderen renommierten Einrichtungen wie Yale oder Stanford angewandt werden, seine berufliche Kompetenz aufbaut.

Diese handlungsorientierte Methode wird während des gesamten Studiengangs angewandt, den der Student bei TECH absolviert. Auf diese Weise wird er mit zahlreichen realen Situationen konfrontiert und muss Wissen integrieren, recherchieren, argumentieren und seine Ideen und Entscheidungen verteidigen. All dies unter der Prämisse, eine Antwort auf die Frage zu finden, wie er sich verhalten würde, wenn er in seiner täglichen Arbeit mit spezifischen, komplexen Ereignissen konfrontiert würde.



Relearning-Methode

Bei TECH werden die case studies mit der besten 100%igen Online-Lernmethode ergänzt: *Relearning*.

Diese Methode bricht mit traditionellen Lehrmethoden, um den Studenten in den Mittelpunkt zu stellen und ihm die besten Inhalte in verschiedenen Formaten zu vermitteln. Auf diese Weise kann er die wichtigsten Konzepte der einzelnen Fächer wiederholen und lernen, sie in einem realen Umfeld anzuwenden.

In diesem Sinne und gemäß zahlreicher wissenschaftlicher Untersuchungen ist die Wiederholung der beste Weg, um zu lernen. Aus diesem Grund bietet TECH zwischen 8 und 16 Wiederholungen jedes zentralen Konzepts innerhalb einer Lektion, die auf unterschiedliche Weise präsentiert werden, um sicherzustellen, dass das Wissen während des Lernprozesses vollständig gefestigt wird.

Das Relearning ermöglicht es Ihnen, mit weniger Aufwand und mehr Leistung zu lernen, sich mehr auf Ihre Spezialisierung einzulassen, einen kritischen Geist zu entwickeln, Argumente zu verteidigen und Meinungen zu kontrastieren: eine direkte Gleichung zum Erfolg.



Ein 100%iger virtueller Online-Campus mit den besten didaktischen Ressourcen

Um ihre Methodik wirksam anzuwenden, konzentriert sich TECH darauf, den Studenten Lehrmaterial in verschiedenen Formaten zur Verfügung zu stellen: Texte, interaktive Videos, Illustrationen und Wissenskarten, um nur einige zu nennen. Sie alle werden von qualifizierten Lehrkräften entwickelt, die ihre Arbeit darauf ausrichten, reale Fälle mit der Lösung komplexer Situationen durch Simulationen, dem Studium von Zusammenhängen, die für jede berufliche Laufbahn gelten, und dem Lernen durch Wiederholung mittels Audios, Präsentationen, Animationen, Bildern usw. zu verbinden.

Die neuesten wissenschaftlichen Erkenntnisse auf dem Gebiet der Neurowissenschaften weisen darauf hin, dass es wichtig ist, den Ort und den Kontext, in dem der Inhalt abgerufen wird, zu berücksichtigen, bevor ein neuer Lernprozess beginnt. Die Möglichkeit, diese Variablen individuell anzupassen, hilft den Menschen, sich zu erinnern und Wissen im Hippocampus zu speichern, um es langfristig zu behalten. Dies ist ein Modell, das als *Neurocognitive context-dependent e-learning* bezeichnet wird und in diesem Hochschulstudium bewusst angewendet wird.

Zum anderen, auch um den Kontakt zwischen Mentor und Student so weit wie möglich zu begünstigen, wird eine breite Palette von Kommunikationsmöglichkeiten angeboten, sowohl in Echtzeit als auch zeitversetzt (internes Messaging, Diskussionsforen, Telefondienst, E-Mail-Kontakt mit dem technischen Sekretariat, Chat und Videokonferenzen).

Darüber hinaus wird dieser sehr vollständige virtuelle Campus den Studenten der TECH die Möglichkeit geben, ihre Studienzeiten entsprechend ihrer persönlichen Verfügbarkeit oder ihren beruflichen Verpflichtungen zu organisieren. Auf diese Weise haben sie eine globale Kontrolle über die akademischen Inhalte und ihre didaktischen Hilfsmittel, in Übereinstimmung mit ihrer beschleunigten beruflichen Weiterbildung.

“

Der Online-Studienmodus dieses Programms wird es Ihnen ermöglichen, Ihre Zeit und Ihr Lerntempo zu organisieren und an Ihren Zeitplan anzupassen“

Die Wirksamkeit der Methode wird durch vier Schlüsselergebnisse belegt:

1. Studenten, die diese Methode anwenden, nehmen nicht nur Konzepte auf, sondern entwickeln auch ihre geistigen Fähigkeiten durch Übungen zur Bewertung realer Situationen und zur Anwendung ihres Wissens.
2. Das Lernen basiert auf praktischen Fähigkeiten, die es den Studenten ermöglichen, sich besser in die reale Welt zu integrieren.
3. Eine einfachere und effizientere Aufnahme von Ideen und Konzepten wird durch die Verwendung von Situationen erreicht, die aus der Realität entstanden sind.
4. Das Gefühl der Effizienz der investierten Anstrengung wird zu einem sehr wichtigen Anreiz für die Studenten, was sich in einem größeren Interesse am Lernen und einer Steigerung der Zeit, die für die Arbeit am Kurs aufgewendet wird, niederschlägt.



Die von ihren Studenten am besten bewertete Hochschulmethodik

Die Ergebnisse dieses innovativen akademischen Modells lassen sich an der Gesamtzufriedenheit der Absolventen der TECH ablesen.

Die Studenten bewerten die pädagogische Qualität, die Qualität der Materialien, die Struktur und die Ziele der Kurse als ausgezeichnet. Es überrascht nicht, dass die Einrichtung im Global Score Index mit 4,9 von 5 Punkten die von ihren Studenten am besten bewertete Universität ist.

Sie können von jedem Gerät mit Internetanschluss (Computer, Tablet, Smartphone) auf die Studieninhalte zugreifen, da TECH in Sachen Technologie und Pädagogik führend ist.

Sie werden die Vorteile des Zugangs zu simulierten Lernumgebungen und des Lernens durch Beobachtung, d. h. Learning from an expert, nutzen können.

In diesem Programm stehen Ihnen die besten Lehrmaterialien zur Verfügung, die sorgfältig vorbereitet wurden:



Studienmaterial

Alle didaktischen Inhalte werden von den Fachkräfte, die den Kurs unterrichten werden, speziell für den Kurs erstellt, so dass die didaktische Entwicklung wirklich spezifisch und konkret ist.

Diese Inhalte werden dann auf ein audiovisuelles Format übertragen, das unsere Online-Arbeitsweise mit den neuesten Techniken ermöglicht, die es uns erlauben, Ihnen eine hohe Qualität in jedem der Stücke zu bieten, die wir Ihnen zur Verfügung stellen werden.



Übungen für Fertigkeiten und Kompetenzen

Sie werden Aktivitäten durchführen, um spezifische Kompetenzen und Fertigkeiten in jedem Fachbereich zu entwickeln. Übungen und Aktivitäten zum Erwerb und zur Entwicklung der Fähigkeiten und Fertigkeiten, die ein Spezialist im Rahmen der Globalisierung, in der wir leben, entwickeln muss.



Interaktive Zusammenfassungen

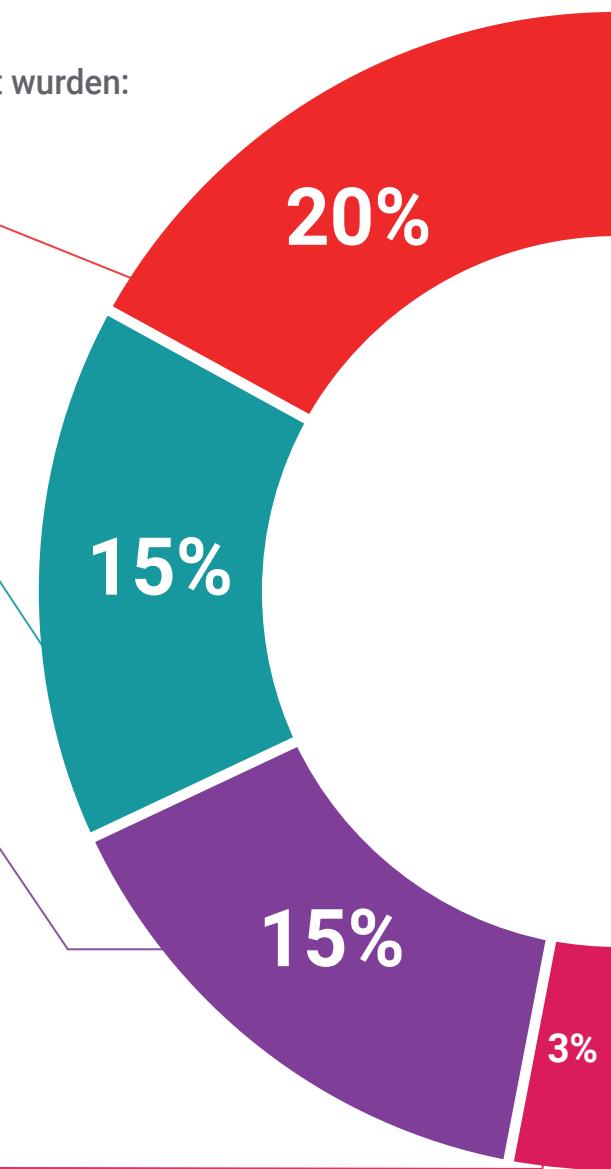
Wir präsentieren die Inhalte auf attraktive und dynamische Weise in multimedialen Pillen, Audios, Videos, Bilder, Diagramme und konzeptionelle Karten enthalten, um das Wissen zu festigen.

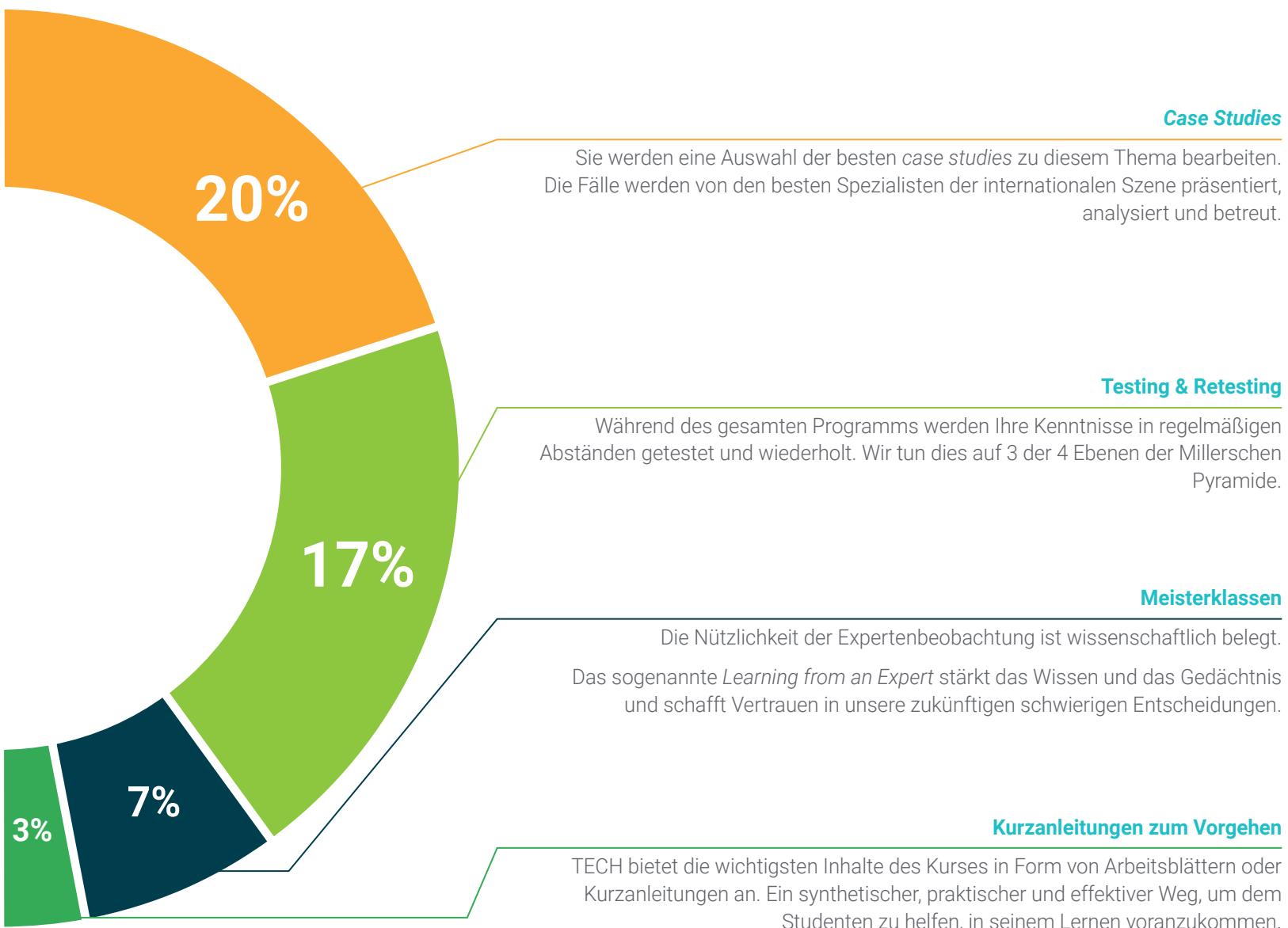
Dieses einzigartige System für die Präsentation multimedialer Inhalte wurde von Microsoft als „Europäische Erfolgsgeschichte“ ausgezeichnet.



Weitere Lektüren

Aktuelle Artikel, Konsensdokumente, internationale Leitfäden... In unserer virtuellen Bibliothek haben Sie Zugang zu allem, was Sie für Ihre Ausbildung benötigen.





Lehrkörper

Die Lehrkräfte dieses Programms sind aktive Fachleute mit direkter Erfahrung in führenden Unternehmen der Technologiebranche. Diese Kombination aus akademischem Wissen und praktischer Erfahrung ermöglicht eine aktuelle Vermittlung von Inhalten, die auf die Anforderungen des Marktes abgestimmt sind. Darüber hinaus garantiert die Vielfalt der Fachgebiete innerhalb des Teams einen multidisziplinären Ansatz, der von Software-Engineering bis hin zu Innovationen in verteilten Systemen reicht. Auf diese Weise wird ein umfassendes Verständnis der digitalen Umgebung gefördert und der Erwerb von Kompetenzen ermöglicht, die den aktuellen Herausforderungen und den neuen Anforderungen der globalen Industrie gerecht werden.



66

Sie erhalten Zugang zu einem Lehrplan, der von einem renommierten, hochqualifizierten Dozententeam für Softwareentwicklung entworfen wurde und Ihnen ein effizientes Lernen garantiert"

Internationaler Gastsdirektor

Amey Patil gilt als internationale Referenz für die Anwendung digitaler Strategien und ist ein erfahrener Software-Ingenieur, der auf den Einsatz modernster technologischer Instrumente spezialisiert ist.

In diesem Sinne hat er bei renommierten Institutionen wie **Google** in Kalifornien gearbeitet. So war er für die Verwaltung mehrerer digitaler Infrastrukturen wie **Ads Creative Studio, Display & Video 360** oder **Google Analytics** zuständig. Unter seiner Leitung haben diese Produkte erhebliche Umsätze generiert, die das **strategische Wachstum** des Unternehmens vorangetrieben und gleichzeitig die **Nutzererfahrung** verbessert haben.

So basiert seine Arbeitsphilosophie auf der Konzeption und Umsetzung von **maßgeschneiderten Online-Lösungen** für Institutionen mit dem Ziel, sowohl deren Online-Präsenz als auch deren Werbeleistung zu optimieren. Auf diese Weise hat er zahlreiche **Marketingkampagnen** geleitet, die den **Return on Investment** maximiert und gleichzeitig den Ruf der Organisationen in der Gesellschaft gestärkt haben.

Darüber hinaus hat er verschiedene Initiativen zur Förderung von **Innovationen** im Bereich der Technologie vorangetrieben. Ein Beispiel dafür ist die Analyse, wie **künstliche Intelligenz** Unternehmen in die Lage versetzt, komplexe Aufgaben wie die Verarbeitung großer Mengen von Informationen zu automatisieren. Auf diese Weise hat er einer Vielzahl von Unternehmen geholfen, ihre betrieblichen Abläufe zu verbessern und fundierte Entscheidungen zu treffen. Außerdem hat er Unternehmen ermöglicht, **Trends** frühzeitig zu erkennen, um einzigartige Vorteile zu schaffen, mit denen sie sich auf dem Markt positionieren konnten.

Im Rahmen seines Engagements für herausragende Leistungen nimmt Amey Patil als Dozent an internationalen **wissenschaftlichen Konferenzen** teil. So gibt er sein umfassendes Wissen zu Themen wie der Einführung **agiler Methoden**, Systemen zur **Qualitätssicherung von Computeranwendungen** und den neuesten Fortschritten im **Datenbankmanagement** in Unternehmen weiter.



Hr. Patil, Amey

- Direktor für Softwareentwicklung bei Google in Kalifornien, USA
- Leiter der Ingenieurabteilung bei Dell Technologies, Kalifornien
- Softwareentwickler bei Mealpit, Kalifornien
- Qualitätssicherungsingenieur bei Amdocs, Zypern
- Spezialist für Google Ads und Google Analytics
- Masterstudiengang in Ingenieurwissenschaften der University of Maryland
- Hochschulabschluss in Telekommunikation an der Universität von Mumbai

“

*Dank TECH werden Sie
mit den besten Fachleuten
der Welt lernen können"*

09

Qualifizierung

Der Weiterbildender Masterstudiengang in Softwareentwicklung garantiert neben der präzisesten und aktuellsten Fortbildung auch den Zugang zu einem von der TECH Global University ausgestellten Diplom.



66

Schließen Sie dieses Programm erfolgreich ab
und erhalten Sie Ihren Universitätsabschluss
ohne lästige Reisen oder Formalitäten"

Mit diesem Programm erwerben Sie den von **TECH Global University**, der größten digitalen Universität der Welt, bestätigten eigenen Titel **Weiterbildender Masterstudiengang in Softwareentwicklung**.

TECH Global University ist eine offizielle europäische Universität, die von der Regierung von Andorra ([Amtsblatt](#)) öffentlich anerkannt ist. Andorra ist seit 2003 Teil des Europäischen Hochschulraums (EHR). Der EHR ist eine von der Europäischen Union geförderte Initiative, die darauf abzielt, den internationalen Ausbildungsräumen zu organisieren und die Hochschulsysteme der Mitgliedsländer dieses Raums zu vereinheitlichen. Das Projekt fördert gemeinsame Werte, die Einführung gemeinsamer Instrumente und die Stärkung der Mechanismen zur Qualitätssicherung, um die Zusammenarbeit und Mobilität von Studenten, Forschern und Akademikern zu verbessern.

Dieser eigene Abschluss der **TECH Global University** ist ein europäisches Programm zur kontinuierlichen Weiterbildung und beruflichen Fortbildung, das den Erwerb von Kompetenzen in seinem Wissensgebiet garantiert und dem Lebenslauf des Studenten, der das Programm absolviert, einen hohen Mehrwert verleiht.

Titel: Weiterbildender Masterstudiengang in Softwareentwicklung

Modalität: online

Dauer: 12 Monate

Akkreditierung: 60 ECTS



zukunft
gesundheit vertrauen menschen
erziehung information tutoren
garantie akkreditierung unterricht
institutionen technologie lernen
gemeinschaft verpflichtung
persönliche betreuung **tech** global university
wissen gegenwart qualität
online-Ausbildung
entwicklung institutionen
virtuelles Klassenzimmer



Weiterbildender Masterstudiengang Softwareentwicklung

- » Modalität: online
- » Dauer: 12 Monate
- » Qualifizierung: TECH Global University
- » Akkreditierung: 60 ECTS
- » Zeitplan: in Ihrem eigenen Tempo
- » Prüfungen: online

Weiterbildender Masterstudiengang Softwareentwicklung

