

Privater Masterstudiengang Erstellung von Schnittstellen und Netzwerkanwendungen



Privater Masterstudiengang Erstellung von Schnittstellen und Netzwerkanwendungen

- » Modalität: online
- » Dauer: 12 Monate
- » Qualifizierung: TECH Technische Universität
- » Aufwand: 16 Std./Woche
- » Zeitplan: in Ihrem eigenen Tempo
- » Prüfungen: online

Internetzugang: www.techtute.com/de/informatik/masterstudiengang/masterstudiengang-erstellung-schnittstellen-netzwerkanwendungen

Index

01

Präsentation

Seite 4

02

Ziele

Seite 8

03

Kompetenzen

Seite 14

04

Struktur und Inhalt

Seite 18

05

Methodik

Seite 32

06

Qualifizierung

Seite 40

01 Präsentation

Um auf dem Gebiet der Informatik wettbewerbsfähig zu sein, ist eine solide Kenntnis der wichtigsten Entwicklungen und Aktualisierungen im Bereich der Softwaretechnik, der mathematischen Grundlagen, der Statistik und anderer Bereiche erforderlich. In diesem Programm haben wir einen intensiven Spezialisierungsweg entwickelt, der den Studenten in die Lage versetzt, sich wissenschaftlich und technologisch auf dem Gebiet der Informatik mit Qualität, Sicherheit und dem ultimativen Ziel der Exzellenz weiterzubilden.



“

Das innovative Konzept der Telepraxis wird Ihnen die Möglichkeit geben, durch eine immersive Erfahrung zu lernen, die Ihnen eine schnellere Integration und einen viel realistischeren Blick auf die Inhalte ermöglicht: "Learning From an Expert"

In dieser Weiterbildung werden die notwendigen Konzepte für die Erstellung von Schnittstellen entwickelt, mit der Sicherheit, die die Beherrschung aller Wissensbereiche zu diesem Thema dem Spezialisten bietet. Während des gesamten Kurses werden innovative didaktische Ansätze vorgeschlagen, um ein vertieftes Verständnis der Architektur einer verteilten Anwendung zu erlangen. Es werden die besonderen Aspekte der Client-Server-Architektur vorgestellt und die Grundlagen und wichtigsten Entwicklungen der am weitesten verbreiteten Programmiersprachen behandelt, wobei zwischen den Sprachen unterschieden wird, was für den Profi von grundlegender Bedeutung ist.

Diese Grundkenntnisse sind auch der erste Schritt, um Zugang zu den Entwicklungskapazitäten dieser Art von Technologie zu erhalten.

Während dieser Weiterbildung wird ein reales Arbeitsszenario zur Verfügung gestellt, um die Eignung der Anwendung im Projekt zu beurteilen, indem die tatsächlichen Indikationen, die Art und Weise der Entwicklung und die Erwartungen an die Ergebnisse bewertet werden.

Durch Erfahrung kann man lernen, wie man das Wissen entwickelt, das man braucht, um in diesem Arbeitsbereich voranzukommen. Diese Fortbildung, die notwendigerweise Erfahrung voraussetzt, verbindet in diesem Bereich Fernstudium mit praktischer Fortbildung und bietet eine einzigartige Gelegenheit, Ihrem Lebenslauf den gewünschten Auftrieb zu geben.

Dieser **Privater Masterstudiengang in Erstellung von Schnittstellen und Netzwerkanwendungen** enthält das vollständigste und aktuellste Programm auf dem Markt. Die hervorstechendsten Merkmale sind:

- ◆ Neueste Technologie in der E-Learning-Software
- ◆ Intensiv visuelles Lehrsystem, unterstützt durch grafische und schematische Inhalte, die leicht zu erfassen und zu verstehen sind
- ◆ Entwicklung von Fallstudien, die von berufstätigen Experten vorgestellt werden
- ◆ Hochmoderne interaktive Videosysteme
- ◆ Unterricht unterstützt durch Telepraxis
- ◆ Systeme zur ständigen Aktualisierung und Überarbeitung
- ◆ Selbstgesteuertes Lernen: Vollständige Kompatibilität mit anderen Berufen
- ◆ Praktische Übungen zur Selbstbeurteilung und Überprüfung des Gelernten
- ◆ Hilfsgruppen und Bildungssynergien: Fragen an den Experten, Diskussions- und Wissensforen
- ◆ Kommunikation mit der Lehrkraft und individuelle Reflexionsarbeit
- ◆ Verfügbarkeit von Inhalten von jedem festen oder tragbaren Gerät mit Internetanschluss aus
- ◆ Datenbanken mit ergänzenden Unterlagen, die auch nach dem Kurs ständig verfügbar sind



Werden Sie einer der gefragtesten Fachleute der Gegenwart: bilden Sie sich in Computertechnik durch den vollständigsten und aktualisiertesten Privaten Masterstudiengang in Erstellung von Schnittstellen und Netzwerkanwendungen weiter"



Mit einem methodischen Konzept, das sich auf bewährte Lehrmethoden stützt, führt Sie dieser innovative Private Masterstudiengang in Erstellung von Schnittstellen und Netzwerkanwendungen durch verschiedene Lehransätze, um Ihnen ein dynamisches und effektives Lernen zu ermöglichen"

Das Dozententeam setzt sich aus Fachleuten aus verschiedenen Bereichen zusammen, die mit diesem Fachgebiet in Verbindung stehen. Auf diese Weise stellt TECH sicher, dass das angestrebte Ziel der Bildungsaktualisierung erreicht wird. Ein multidisziplinäres Team von qualifizierten und erfahrenen Fachleuten aus verschiedenen Bereichen, die Ihnen die theoretischen Kenntnisse auf effiziente Weise vermitteln, Ihnen aber vor allem das praktische Wissen aus ihrer eigenen Erfahrung zur Verfügung stellen: eine der besonderen Qualitäten dieser Weiterbildung.

Diese Beherrschung des Themas wird durch die Effizienz der methodischen Konzeption ergänzt. Es wurde von einem multidisziplinären Team von E-Learning-Experten entwickelt und integriert die neuesten Fortschritte in der Bildungstechnologie. Auf diese Weise können Sie mit einer Reihe komfortabler und vielseitiger multimedialer Werkzeuge studieren, die Ihnen die nötige Handlungskompetenz für Ihre Weiterbildung vermitteln.

Das Programm basiert auf problemorientiertem Lernen: ein Ansatz, der Lernen als einen eminent praktischen Prozess begreift. Um dies aus der Ferne zu erreichen, werden Telepraktika eingesetzt: Mit Hilfe eines innovativen interaktiven Videosystems wird der Student, der von einem Experten lernt, in der Lage sein, sich das Wissen so anzueignen, als würde er das Szenario, das er lernt, in diesem Moment erleben. Ein Konzept, das eine realistischere und nachhaltigere Integration und Konsolidierung des Lernens ermöglicht.

Ein privater Masterstudiengang, der Sie befähigt, in allen Bereichen der Erstellung von Schnittstellen und Netzwerkanwendungen mit der Kompetenz eines hochqualifizierten Spezialisten zu arbeiten"

Mit der Erfahrung von Fachleuten, die Ihnen ein reales, unmittelbares und konkretes Wissen über diesen Arbeitsbereich vermitteln.



02 Ziele

Ziel von TECH ist es, den Fachkräften aktuelle Informationen über die Erstellung von Schnittstellen und Netzwerkanwendungen zur Verfügung zu stellen. Dadurch soll der Systemingenieur neue Fähigkeiten für die Realisierung vollständigerer Computerprogramme erwerben. Ein Ziel, das in wenigen Monaten durch einen auf die aktuellen Bedürfnisse des Sektors ausgerichteten Lehrplan erreicht werden kann.



“

Erweitern Sie Ihr Wissen in den Bereichen Informatik und Softwaretechnik und machen Sie sich bereit für den Wettbewerb mit den Besten der Branche"



Allgemeine Ziele

- ◆ Ausbilden in Wissenschaft und Technik für die Praxis der Computertechnik
- ◆ Erlangen von umfassenden Kenntnissen auf dem Gebiet der Computerwissenschaften
- ◆ Erlangen von umfassenden Kenntnissen auf dem Gebiet der Computerstruktur
- ◆ Erwerben der notwendigen Kenntnisse in der Softwareentwicklung
- ◆ Wiederholen der mathematischen, statistischen und physikalischen Grundlagen, die für dieses Fach unerlässlich sind

“

Ein Weg zu Fortbildung und beruflichem Wachstum, der Ihnen zu mehr Wettbewerbsfähigkeit auf dem Arbeitsmarkt verhelfen wird"





Spezifische Ziele

Modul 1. Interaktion zwischen Mensch und Computer

- ◆ Aneignen solider Kenntnisse über die Interaktion zwischen Mensch und Computer und die Erstellung benutzbarer Schnittstellen
- ◆ Verstehen, wie wichtig die Benutzerfreundlichkeit von Anwendungen ist und warum wir sie bei der Entwicklung unserer Software berücksichtigen müssen
- ◆ Verstehen der verschiedenen Arten menschlicher Vielfalt und die damit verbundenen Einschränkungen und Wissen, wie man Schnittstellen an die spezifischen Bedürfnisse jeder Art von Vielfalt anpasst
- ◆ Kennenlernen des Prozesses des Schnittstellendesigns, von der Anforderungsanalyse bis zur Evaluierung, über die verschiedenen Zwischenstufen, die zur Realisierung einer geeigneten Schnittstelle notwendig sind
- ◆ Kennen der verschiedenen Zugänglichkeitsrichtlinien, der Normen, die sie festlegen, und der Hilfsmittel, die es uns ermöglichen, sie zu bewerten
- ◆ Die verschiedenen Methoden der Interaktion mit dem Computer durch Peripherie- und andere Geräte

Modul 2. Datenbanken

- ◆ Erlernen der verschiedenen Anwendungen und Zwecke von Datenbanksystemen sowie deren Betrieb und Architektur kennen
- ◆ Verstehen des relationalen Modells, von seiner Struktur und seinen Operationen bis hin zur erweiterten relationalen Algebra
- ◆ Lernen, was SQL-Datenbanken sind, wie sie funktionieren, wie man Daten definiert und wie man Abfragen erstellt, von den einfachsten bis zu den fortgeschrittensten und komplexesten
- ◆ Lernen, wie man Datenbanken mit Hilfe des *Entity-Relationship*-Modells entwirft, wie man Diagramme erstellt und die Eigenschaften des erweiterten ER-Modells

- ♦ Vertiefen des Entwurfs von relationalen Datenbanken, Analysieren der verschiedenen Normalformen und Zerlegungsalgorithmen
- ♦ Schaffen der Grundlagen für das Verständnis der Funktionsweise von NoSQL-Datenbanken sowie Einführung in die Datenbank Mongo DB

Modul 3. Entwicklung von Netzwerkanwendungen

- ♦ Kennen der Merkmale der Auszeichnungssprache HTML und ihrer Verwendung bei der Erstellung von Webseiten in Verbindung mit CSS-Stilvorlagen
- ♦ Lernen des Umgangs mit der browserorientierten Programmiersprache JavaScript und einiger ihrer wichtigsten Funktionen
- ♦ Verstehen der Konzepte der komponentenorientierten Programmierung und der Komponentenarchitektur
- ♦ Lernen, wie man das *Bootstrap Front-End Framework* für die Gestaltung von Websites verwendet
- ♦ Verstehen der Struktur des *Controller-View-Modells* bei der Entwicklung von dynamischen Websites
- ♦ Kennen der dienstleistungsorientierten Architektur und der Grundlagen des HTTP-Protokolls

Modul 4. Freie Software und Freies Wissen

- ♦ Lernen der Konzepte der freien Software und des freien Wissens sowie der verschiedenen damit verbundenen Lizenztypen
- ♦ Kennen der wichtigsten kostenlosen Tools für verschiedene Bereiche wie Betriebssysteme, *Business Management*, *Content Manager* und Erstellen von Multimedia-Inhalten
- ♦ Verstehen der Bedeutung und der Vorteile von freier Software in der Geschäftswelt, sowohl in Bezug auf ihre Eigenschaften als auch auf ihre Kosten
- ♦ Vertiefen der Kenntnisse des GNU/Linux-Betriebssystems sowie der verschiedenen existierenden Distributionen und wie man diese individuell anpasst
- ♦ Lernen der Funktionsweise und der Entwicklung von WordPress, da dieses CMS mehr als 35% der aktiven Websites in der Welt ausmacht, und mehr als 60% im besonderen Falle von CMS

- ♦ Verstehen, wie das Betriebssystem für Android-Mobilgeräte funktioniert, und Lernen der Grundlagen der Entwicklung mobiler Anwendungen sowohl nativ als auch mit plattformübergreifenden *Frameworks*

Modul 5. Fortgeschrittene Datenbanken

- ♦ Vorstellen der verschiedenen Datenbanksysteme, die derzeit auf dem Markt erhältlich sind
- ♦ Erlernen der Verwendung von XML und Datenbanken für das Web
- ♦ Verstehen der Funktionsweise fortgeschrittener Datenbanken, wie z. B. parallele und verteilte Datenbanken
- ♦ Verstehen der Bedeutung von Indizierung und Assoziierung in Datenbanksystemen
- ♦ Verstehen der Funktionsweise von transaktionalen Verarbeitungs- und Abfragesystemen
- ♦ Erwerben von Kenntnissen in Bezug auf nichtrelationale Datenbanken und *Data Mining*

Modul 6. Software-Entwicklung

- ♦ Verstehen des Software-Engineering-Rahmens und der Norm ISO/IEC 12207
- ♦ Kennenlernen der Merkmale des einheitlichen Softwareentwicklungsprozesses und der Planung im Kontext der agilen Softwareentwicklung
- ♦ Kennenlernen der verschiedenen Arten von verteiltem Softwaredesign und dienstorientierten Softwarearchitekturen
- ♦ Lernen der grundlegenden Konzepte für die Gestaltung grafischer Benutzeroberflächen
- ♦ Verstehen der Grundlagen der Entwicklung von Web-Applikationen
- ♦ Vertiefen der Strategien und Techniken des Softwaretests, der Faktoren für die Softwarequalität und der verschiedenen verwendeten Messgrößen

Modul 7. Fortgeschrittene Programmierung

- ♦ Vertiefen der Kenntnisse in der Programmierung, insbesondere in Bezug auf die objektorientierte Programmierung und die verschiedenen Arten von Beziehungen zwischen bestehenden Klassen
- ♦ Die verschiedenen Entwurfsmuster für objektorientierte Probleme kennenlernen
- ♦ Lernen der ereignisgesteuerte Programmierung und die Entwicklung von Benutzeroberflächen mit Qt kennen
- ♦ Aneignen der grundlegenden Kenntnisse über nebenläufige Programmierung, Prozesse und *Threads*
- ♦ Das Verwenden von *Threads* und Synchronisierung sowie das Lösen gängiger Probleme bei der gleichzeitigen Programmierung
- ♦ Verstehen der Bedeutung von Dokumentation und Tests bei der Softwareentwicklung

Modul 8. Wiederverwendung von Software

- ♦ Kennen des Gesamtbilds der Software-Wiederverwendungsstrategie
- ♦ Erlernen der verschiedenen Muster für die Wiederverwendung von Software, sowohl in Bezug auf Design, Erstellung, Struktur und Verhalten
- ♦ Einführen in das Konzept des *Framework* sowie Kennenlernen der wichtigsten Typen, wie z. B. derjenigen, die für die Gestaltung von grafischen Benutzeroberflächen, die Entwicklung von Webanwendungen und die Verwaltung der Persistenz von Objekten in Datenbanken bestimmt sind
- ♦ Verstehen der Funktionsweise des derzeit weit verbreiteten *Model-View-Controller* (MVC)-Musters

Modul 9. Künstliche Intelligenz und Wissensmodellierung

- ♦ Festlegen der Grundlagen der künstlichen Intelligenz und der Wissensmodellierung durch einen kurzen Blick auf die Geschichte der künstlichen Intelligenz bis zum heutigen Tag
- ♦ Verstehen der wesentlichen Konzepte der Suche in der künstlichen Intelligenz, sowohl der informierten als auch der uninformierten Suche
- ♦ Verstehen, wie künstliche Intelligenz in Spielen funktioniert
- ♦ Lernen der grundlegenden Konzepte von neuronalen Netzen und der Verwendung genetischer Algorithmen
- ♦ Aneignen geeigneter Mechanismen zur Darstellung von Wissen, insbesondere im Hinblick auf das semantische Web
- ♦ Verstehen der Funktionsweise von Expertensystemen und Systemen zur Entscheidungsunterstützung

Modul 10. Fortgeschrittene Softwaretechnik

- ♦ Tiefes Verstehen der Kenntnisse über die verschiedenen agilen Methoden, die in der Softwareentwicklung eingesetzt werden
- ♦ Entwickeln lernen mit Scrum, extremer Programmierung und wiederverwendungsbasierten Softwareentwicklungstechniken
- ♦ Verstehen verschiedener Systemarchitekturen und Softwareentwurfsmuster sowie der Architektur von *Cloud*-Anwendungen
- ♦ Erlernen der Durchführung von Softwaretests mit Methoden wie *Test Driven Development*, *Acceptance Test Driven Development*, *Behavior Driven Development*, BDD und *Cucumber*
- ♦ Vertiefen der Verbesserung des Softwareentwicklungsprozesses und der Softwarequalität anhand von ISO/IEC-Normen
- ♦ Einführen in das Konzept von DevOps und seine wichtigsten Praktiken

03 Kompetenzen

Dieses Programm in Erstellung von Schnittstellen und Netzwerkanwendungen wurde als hochqualifiziertes Instrument für Fachkräfte geschaffen. Sein intensiver Lehrplan wird wesentlich zur Entwicklung und Arbeit von Webprogrammierern und Webportalen beitragen, indem er die vorhandenen Internetressourcen und freie Software praxisnah und nützlich einsetzt.





“

Mit den Fähigkeiten, die Sie mit diesem privaten Masterstudiengang erwerben können, werden Sie in der Lage sein, mit der Entwicklung von Anwendungen zu beginnen und als Software-Ingenieur zu arbeiten, dank einer vollständigen und aktualisierten Weiterbildung"



Allgemeine Kompetenz

- ♦ Erwerben der notwendigen Fähigkeiten für die berufliche Praxis der Computertechnik mit der Kenntnis aller Faktoren, die notwendig sind, um sie mit Qualität und Solvenz auszuführen

“

Eine Gelegenheit, die für Fachleute geschaffen wurde, die ein intensives und effektives Programm suchen, um in ihrem Beruf einen bedeutenden Schritt nach vorne zu machen"





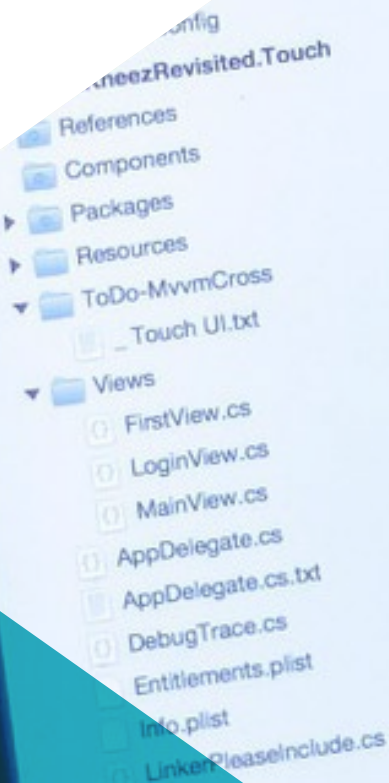
Spezifische Kompetenzen

- ◆ Erlangen eines tiefgreifenden Verständnisses für alle Facetten der Mensch-Computer-Interaktion und wie diese mit IT-Entwicklungen zusammenhängen
- ◆ Beherrschen der Nutzung von Datenbanken
- ◆ Entwickeln verschiedener Arten von Netzwerkanwendungen
- ◆ Beschreiben und Nutzen von freier Software und freiem Wissen, das im Internet verfügbar ist
- ◆ Arbeiten als Software-Ingenieur
- ◆ Kontrollieren der Nutzung fortgeschrittener Datenbanken
- ◆ Durchführen einer fortgeschrittenen Programmierung
- ◆ Wissen, wie Software wiederverwendet wird
- ◆ Erstellen von Schnittstellen und Netzwerkanwendungen
- ◆ Beherrschen der verschiedenen Systeme der fortgeschrittenen Softwareentwicklung

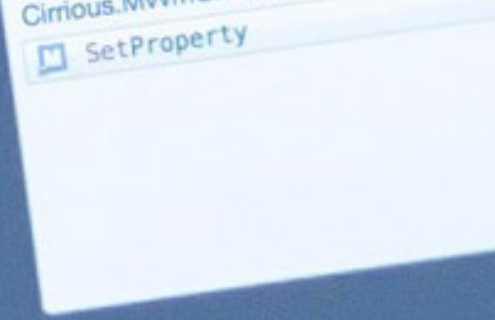
04 Struktur und Inhalt

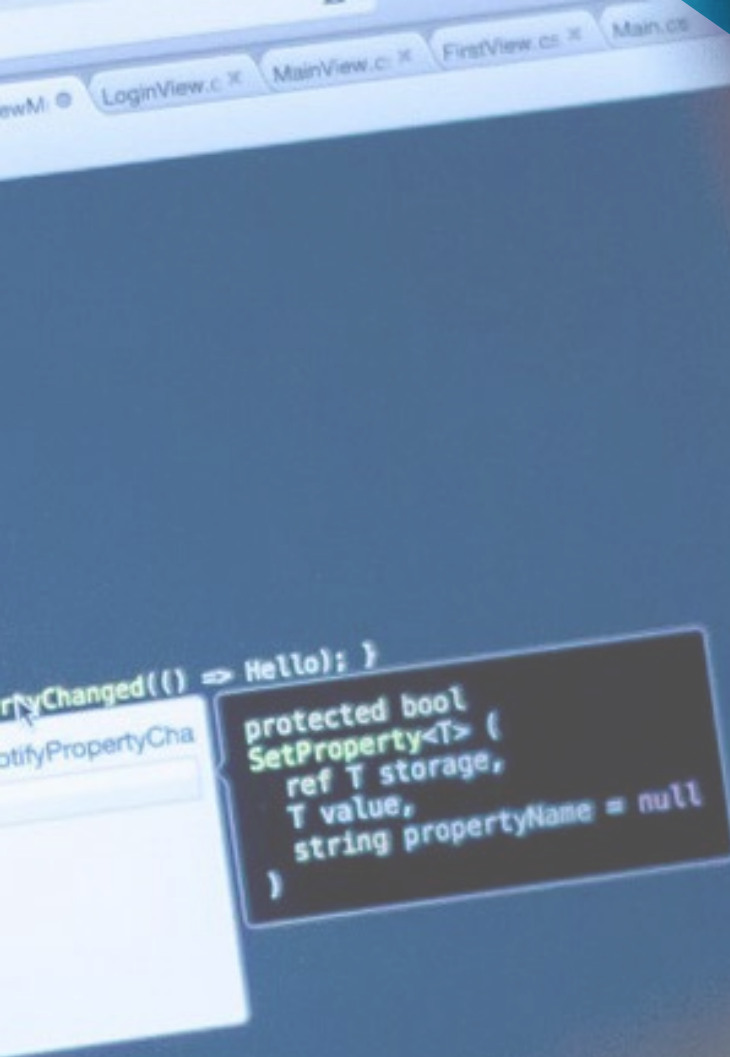
Die Inhalte dieses privaten Masterstudiengangs wurden von den verschiedenen Experten dieses Studiengangs mit einem klaren Ziel entwickelt: sicherzustellen, dass die Studenten alle notwendigen Fähigkeiten erwerben um echte Experten in diesem Bereich zu werden

Ein sehr komplettes und gut strukturiertes Programm, das sie zu höchsten Qualitäts- und Erfolgsstandards führen wird.



```
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
public class MainViewModel : MvxViewModel  
{  
    public MainViewModel ()  
    {  
        private string _hello = "Hello MOFO";  
        public string Hello  
        {  
            get { return _hello; }  
            set { set _hello = value; RaisePrope  
        }  
    }  
}
```





“

Ein sehr komplettes Lehrprogramm, das in vollständige didaktische Einheiten gegliedert ist, die auf der Grundlage der neuesten Entwicklungen des Sektors aktualisiert werden, und das auf ein Studium ausgerichtet ist, das mit Ihrem persönlichen und beruflichen Leben vereinbar ist"

Modul 1. Mensch-Computer-Interaktion

- 1.1. Einführung in die Mensch-Computer-Interaktion
 - 1.1.1. Was ist Mensch-Computer-Interaktion?
 - 1.1.2. Beziehung der Mensch-Computer-Interaktion zu anderen Disziplinen
 - 1.1.3. Die Benutzeroberfläche
 - 1.1.4. Benutzerfreundlichkeit und Zugänglichkeit
 - 1.1.5. Benutzererfahrung und benutzerzentriertes Design
- 1.2. Der Computer und die Interaktion: Benutzeroberfläche und Interaktionsparadigmen
 - 1.2.1. Interaktion
 - 1.2.2. Interaktionsparadigmen und Interaktionsstile
 - 1.2.3. Entwicklung von Benutzeroberflächen
 - 1.2.4. Klassische Benutzeroberflächen: WIMP/GUI, Befehle, Stimme, Virtuelle Realität
 - 1.2.5. Innovative Benutzeroberflächen: mobil, tragbar, kollaborativ, BCI
- 1.3. Der menschliche Faktor: psychologische und kognitive Aspekte
 - 1.3.1. Die Bedeutung des menschlichen Faktors in der Interaktion
 - 1.3.2. Menschliche Informationsverarbeitung
 - 1.3.3. Die Eingabe und Ausgabe von Informationen: visuell, auditiv und taktil
 - 1.3.4. Wahrnehmung und Aufmerksamkeit
 - 1.3.5. Wissen und mentale Modelle: Darstellung, Organisation und Erwerb
- 1.4. Der menschliche Faktor: sensorische und physische Einschränkungen
 - 1.4.1. Funktionelle Vielfalt, Behinderung und Beeinträchtigung
 - 1.4.2. Visuelle Vielfalt
 - 1.4.3. Akustische Vielfalt
 - 1.4.4. Kognitive Vielfalt
 - 1.4.5. Motorische Vielfalt
 - 1.4.6. Der Fall der digitalen Einwanderer
- 1.5. Der Designprozess (I): Anforderungsanalyse für die Gestaltung der Benutzeroberfläche
 - 1.5.1. Benutzerzentriertes Design
 - 1.5.2. Was ist eine Anforderungsanalyse?
 - 1.5.3. Sammeln von Informationen
 - 1.5.4. Analyse und Interpretation der Informationen
 - 1.5.5. Der Designprozess (II): Prototyping und Aufgabenanalyse



- 1.6. Analyse der Benutzerfreundlichkeit und Zugänglichkeit
 - 1.6.1. Konzeptioneller Entwurf
 - 1.6.2. Prototyping
 - 1.6.3. Hierarchische Aufgabenanalyse
- 1.7. Der Designprozess (III): Bewertung
 - 1.7.1. Bewertung im Designprozess: Ziele und Methoden
 - 1.7.2. Bewertungsmethoden ohne Benutzer
 - 1.7.3. Bewertungsmethoden mit Benutzern
 - 1.7.4. Bewertungsstandards und -normen
- 1.8. Zugänglichkeit: Definition und Leitlinien
 - 1.8.1. Barrierefreiheit und universelles Design
 - 1.8.2. WAI-Initiative und WCAG-Richtlinien
 - 1.8.3. WCAG-Richtlinien 2.0 und 2.1
- 1.9. Zugänglichkeit: Bewertung und Funktionsvielfalt
 - 1.9.1. Tools zur Bewertung der Barrierefreiheit im Web
 - 1.9.2. Zugänglichkeit und Funktionsvielfalt
- 1.10. Der Computer und die Interaktion: Peripheriegeräte und Zubehör
 - 1.10.1. Herkömmliche und Peripheriegeräte
 - 1.10.2. Alternative und Peripheriegeräte
 - 1.10.3. Handys und Tablets
 - 1.10.4. Funktionsvielfalt, Interaktion und Peripheriegeräte

Modul 2. Datenbanken

- 2.1. Anwendungen und Zwecke von Datenbanksystemen
 - 2.1.1. Anwendungen der verschiedenen Datenbanksysteme
 - 2.1.2. Zweck in verschiedenen Datenbanksystemen
 - 2.1.3. Übersicht der Daten
- 2.2. Datenbank und Architektur
 - 2.2.1. Relationale Datenbanken
 - 2.2.2. Datenbank-Design
 - 2.2.3. Objektbasierte und semistrukturierte Datenbanken
 - 2.2.4. Datenspeicherung und Abfragen
 - 2.2.5. Transaktionsmanagement
 - 2.2.6. *Data Mining* und Analyse
 - 2.2.7. Datenbank-Architektur

- 2.3. Das relationale Modell: Struktur, Operationen und erweiterte relationale Algebra
 - 2.3.1. Die Struktur von relationalen Datenbanken
 - 2.3.2. Grundlegende Operationen in der relationalen Algebra
 - 2.3.3. Andere Operationen der relationalen Algebra
 - 2.3.4. Erweiterte Operationen der relationalen Algebra
 - 2.3.5. Nullwerte
 - 2.3.6. Modifizierung der Datenbank
- 2.4. SQL (I)
 - 2.4.1. Was ist SQL?
 - 2.4.2. Definition der Daten
 - 2.4.3. Grundlegende Struktur der SQL-Abfragen
 - 2.4.4. Operationen auf Sets
 - 2.4.5. Aggregationsfunktionen
 - 2.4.6. Nullwerte
- 2.5. SQL (II)
 - 2.5.1. Verschachtelte Sub-Abfragen
 - 2.5.2. Komplexe Abfragen
 - 2.5.3. Ansichten
 - 2.5.4. *Cursors*
 - 2.5.5. Komplexe Abfragen
 - 2.5.6. *Triggers*
- 2.6. Datenbankdesign und das ER-Modell
 - 2.6.1. Überblick über den Entwurfsprozess
 - 2.6.2. Das *Entity-Relationship*-Modell
 - 2.6.3. Beschränkungen
- 2.7. *Entity-Relationship*-Diagramme
 - 2.7.1. *Entity-Relationship*-Diagramme
 - 2.7.2. Aspekte der Gestaltung von *Entity-Relationship*
 - 2.7.3. Gruppen von schwachen Entitäten
- 2.8. Das erweiterte *Entity-Relationship*-Modell
 - 2.8.1. Merkmale des erweiterten ER-Modells
 - 2.8.2. Datenbank-Design
 - 2.8.3. Reduktion auf relationale Schemata

- 2.9. Relationaler Datenbankentwurf
 - 2.9.1. Merkmale eines guten relationalen Designs
 - 2.9.2. Atomare Domänen und die erste Normalform (1NF)
 - 2.9.3. Dekomposition durch funktionale Abhängigkeiten
 - 2.9.4. Funktionale Abhängigkeitstheorie
 - 2.9.5. Zersetzungsalgorithmen
 - 2.9.6. Zerlegung unter Verwendung mehrwertiger Abhängigkeiten
 - 2.9.7. Weitere Normalformen
 - 2.9.8. Prozess der Datenbankentwicklung
- 2.10. NoSQL-Datenbanken
 - 2.10.1. Was sind NoSQL-Datenbanken?
 - 2.10.2. Analyse der verschiedenen NoSQL-Optionen und ihrer Eigenschaften
 - 2.10.3. Mongo DB

Modul 3. Entwicklung von Netzwerkanwendungen

- 3.1. HTML5-Auszeichnungssprachen
 - 3.1.1. Grundlegende Konzepte der HTML
 - 3.1.2. Neue HTML 5-Elemente
 - 3.1.3. Formulare: neue Steuerelemente
- 3.2. Einführung in CSS-Stylesheets
 - 3.2.1. Erste Schritte mit CSS
 - 3.2.2. Einführung in CSS3
- 3.3. Browser-Skriptsprache: JavaScript
 - 3.3.1. JavaScript-Grundlagen
 - 3.3.2. DOM
 - 3.3.3. Events
 - 3.3.4. jQuery
 - 3.3.5. Ajax
- 3.4. Konzept der komponentenorientierten Programmierung
 - 3.4.1. Kontext
 - 3.4.2. Komponenten und Schnittstellen
 - 3.4.3. Zustände einer Komponente

- 3.5. Architektur von Komponenten
 - 3.5.1. Zeitgenössische Architekturen
 - 3.5.2. Integration und Einsatz von Komponenten
- 3.6. *Framework Front-End: Bootstrap*
 - 3.6.1. Rasterdesign
 - 3.6.2. Formulare
 - 3.6.3. Komponenten
- 3.7. Model-View-Controller
 - 3.7.1. Methoden der Webentwicklung
 - 3.7.2. Entwurfsmuster: MVC
- 3.8. *Grid-Informationstechnologien*
 - 3.8.1. Erhöhte Computerressourcen
 - 3.8.2. Konzept der *Grid*-Technologie
- 3.9. Serviceorientierte Architektur
 - 3.9.1. SOA und Webdienste
 - 3.9.2. Topologie eines Webdienstes
 - 3.9.3. Plattformen für Webdienste
- 3.10. HTTP-Protokoll
 - 3.10.1. Nachrichten
 - 3.10.2. Dauerhafte Sitzungen
 - 3.10.3. Kryptographisches System
 - 3.10.4. Funktionsweise des HTTPS-Protokolls

Modul 4. Freie Software und freies Wissen

- 4.1. Einführung in Freie Software
 - 4.1.1. Geschichte der Freien Software
 - 4.1.2. „Freiheit“ in Software
 - 4.1.3. Lizenzen für die Nutzung von Software-Tools
 - 4.1.4. Geistiges Eigentum an Software
 - 4.1.5. Was ist die Motivation für die Verwendung freier Software?
 - 4.1.6. Mythen über Freie Software
 - 4.1.7. Top500
- 4.2. Freies Wissen und CC-Lizenzen
 - 4.2.1. Grundlegende Konzepte
 - 4.2.2. *Creative Commons* Lizenzen
 - 4.2.3. Andere Inhaltslizenzen
 - 4.2.4. Wikipedia und andere Projekte des freien Wissens
- 4.3. Wichtigste kostenlose Software-Tools
 - 4.3.1. Operative Systeme
 - 4.3.2. Office-Anwendungen
 - 4.3.3. Business-Management-Anwendungen
 - 4.3.4. Web-Content-Manager
 - 4.3.5. Tools zur Erstellung von Multimedia-Inhalten
 - 4.3.6. Andere Anwendungen
- 4.4. Das Unternehmen: Freie Software und ihre Kosten
 - 4.4.1. Freie Software: Ja oder Nein?
 - 4.4.2. Wahrheiten und Lügen über Freie Software
 - 4.4.3. Unternehmenssoftware auf der Grundlage freier Software
 - 4.4.4. Software Kosten
 - 4.4.5. FOSS-Modelle
- 4.5. Das GNU/Linux-Betriebssystem
 - 4.5.1. Architektur
 - 4.5.2. Grundlegende Verzeichnisstruktur
 - 4.5.3. Merkmale und Struktur des Dateisystems
 - 4.5.4. Interne Darstellung von Dateien
- 4.6. Das mobile Betriebssystem Android
 - 4.6.1. Geschichte
 - 4.6.2. Architektur
 - 4.6.3. Forks von Android
 - 4.6.4. Einführung in die Android-Entwicklung
 - 4.6.5. *Frameworks* für die Entwicklung mobiler Anwendungen
- 4.7. Websites mit WordPress erstellen
 - 4.7.1. WordPress Funktionen und Struktur
 - 4.7.2. Erstellung von wordpress.com-Websites
 - 4.7.3. Installieren und Konfigurieren von WordPress auf einem eigenen Server
 - 4.7.4. Installation von Plugins und Erweiterung von WordPress
 - 4.7.5. WordPress-Plugins erstellen
 - 4.7.6. WordPress-Themen erstellen
- 4.8. Freie Software Trends
 - 4.8.1. Cloud-Umgebungen
 - 4.8.2. Tools zur Überwachung
 - 4.8.3. Operative Systeme
 - 4.8.4. *Big Data* und *Open Data 2.0*
 - 4.8.5. Quantencomputing
- 4.9. Versionskontrolle
 - 4.9.1. Grundlegende Konzepte
 - 4.9.2. Git
 - 4.9.3. Cloud- und selbstgehostete Git-Dienste
 - 4.9.4. Andere Versionskontrollsysteme
- 4.10. Benutzerdefinierte GNU/Linux-Distributionen
 - 4.10.1. Wichtige Distributionen
 - 4.10.2. Von Debian abgeleitete Distributionen
 - 4.10.3. Erstellen von .deb-Paketen
 - 4.10.4. Ändern der Distribution
 - 4.10.5. Erzeugen von ISO-Images

Modul 5. Fortgeschrittene Datenbanken

- 5.1. Einführung in verschiedene Datenbanksysteme
 - 5.1.1. Historischer Rückblick
 - 5.1.2. Hierarchische Datenbanken
 - 5.1.3. Netzwerk-Datenbanken
 - 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. XML-Datenspeicherung
 - 5.2.4. XML-relationale 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 Konsultationen
 - 5.3.4. Parallelität zwischen den Konsultationen
 - 5.3.5. Paralleler Systementwurf
 - 5.3.6. Parallelverarbeitung 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 Assoziierung
 - 5.5.1. Geordnete Indizes
 - 5.5.2. Dichte und spärliche Indizes
 - 5.5.3. Multilevel-Indizes
 - 5.5.4. Aktualisierung des Indexes
 - 5.5.5. Statische Assoziation
 - 5.5.6. Wie man Indizes in Datenbanken verwendet



- 5.6. Einführung in die transaktionale Verarbeitung
 - 5.6.1. Zustände einer Transaktion
 - 5.6.2. Implementierung von Atomarität und Dauerhaftigkeit
 - 5.6.3. Sequenzierung
 - 5.6.4. Wiederherstellbarkeit
 - 5.6.5. Implementierung der Isolierung
- 5.7. Rückgewinnungssysteme
 - 5.7.1. Klassifizierung von Fehlern
 - 5.7.2. Strukturen zur Lagerung
 - 5.7.3. Rückforderung und Atomarität
 - 5.7.4. Rückforderung auf der Grundlage historischer Daten
 - 5.7.5. Gleichzeitige Transaktionen und Wiederherstellung
 - 5.7.6. Hohe Verfügbarkeit von Datenbanken
- 5.8. Ausführung und Bearbeitung von Abfragen
 - 5.8.1. Kosten für eine Beratung
 - 5.8.2. Auswahlverfahren
 - 5.8.3. Ordination
 - 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. Software-Entwicklung

- 6.1. Rahmen der Softwaretechnik
 - 6.1.1. Eigenschaften der Software
 - 6.1.2. Die wichtigsten Prozesse der Softwaretechnik
 - 6.1.3. Prozessmodelle für die Softwareentwicklung
 - 6.1.4. Standard-Referenzrahmen für den Softwareentwicklungsprozess: die Norm ISO/IEC 12207
- 6.2. Einheitlicher Softwareentwicklungsprozess
 - 6.2.1. Einheitlicher Prozess
 - 6.2.2. Dimensionen des einheitlichen Prozesses
 - 6.2.3. Anwendungsfallorientierter Entwicklungsprozess
 - 6.2.4. Grundlegende Arbeitsabläufe einheitlicher Prozesse
- 6.3. Planung im Rahmen der agilen Softwareentwicklung
 - 6.3.1. Eigenschaften der agilen Softwareentwicklung
 - 6.3.2. Unterschiedliche Planungszeithorizonte in der agilen Entwicklung
 - 6.3.3. Rahmen für die agile Entwicklung nach Scrum und Planung von Zeithorizonten
 - 6.3.4. Anwenderberichte als Planungs- und Schätzungseinheit
 - 6.3.5. Übliche Techniken zur Ableitung einer Schätzung
 - 6.3.6. Skalen für die Interpretation von Schätzungen
 - 6.3.7. *Planning Poker*
 - 6.3.8. Gängige Planungsarten: Lieferplanung und Iterationsplanung
- 6.4. Designstile für verteilte Software und dienstorientierte Softwarearchitekturen
 - 6.4.1. Kommunikationsmodelle in verteilten Softwaresystemen
 - 6.4.2. Zwischenschicht oder *Middleware*
 - 6.4.3. Architekturmuster für verteilte Systeme
 - 6.4.4. Allgemeiner Prozess zur Entwicklung von Softwarediensten
 - 6.4.5. Gestaltungsaspekte von Softwarediensten
 - 6.4.6. Zusammensetzung der Dienstleistungen
 - 6.4.7. Architektur der Webdienste
 - 6.4.8. Komponenten für Infrastruktur und SOA
- 6.5. Einführung in die modellgetriebene Softwareentwicklung
 - 6.5.1. Das Konzept des Modells
 - 6.5.2. Modellgestützte Softwareentwicklung
 - 6.5.3. MDA Rahmen für modellbasierte Entwicklung
 - 6.5.4. Elemente eines Transformationsmodells
- 6.6. Gestaltung der grafischen Benutzeroberfläche
 - 6.6.1. Grundlagen der Gestaltung der Benutzeroberfläche
 - 6.6.2. Architektonische Entwurfsmuster für interaktive Systeme: *Model-View-Controller (MVC)*
 - 6.6.3. Benutzererfahrung (UX *User Experience*)
 - 6.6.4. Benutzerzentriertes Design
 - 6.6.5. Prozess der Analyse und Gestaltung grafischer Benutzeroberflächen
 - 6.6.6. Benutzerfreundlichkeit der Benutzeroberfläche
 - 6.6.7. Zugänglichkeit der Benutzeroberflächen
- 6.7. Gestaltung von Web-Applikationen
 - 6.7.1. Eigenschaften von Web-Applikationen
 - 6.7.2. Benutzeroberfläche einer Web-Applikation
 - 6.7.3. Gestaltung der Navigation
 - 6.7.4. Grundlegendes Interaktionsprotokoll für Web-Applikationen
 - 6.7.5. Architekturstile für Web-Applikationen
- 6.8. Strategien und Techniken der Softwareprüfung und Faktoren der Softwarequalität
 - 6.8.1. Prüfstrategien
 - 6.8.2. Gestaltung von Testfällen
 - 6.8.3. Preis-Leistungs-Verhältnis
 - 6.8.4. Qualitätsmodelle
 - 6.8.5. ISO/IEC 25000-Normenfamilie (SQuaRE)
 - 6.8.6. Modell der Produktqualität (ISO 2501n)
 - 6.8.7. Modelle der Datenqualität (ISO 2501n)
 - 6.8.8. Software-Qualitätsmanagement

- 6.9. Einführung in Messgrößen in der Softwaretechnik
 - 6.9.1. Grundlegende Konzepte: Messgrößen, Metriken und Indikatoren
 - 6.9.2. Arten von Messgrößen in der Softwaretechnik
 - 6.9.3. Der Messprozess
 - 6.9.4. ISO 25024. Externe und verwendete Qualitätsmessgrößen
 - 6.9.5. Objektorientierte Messgrößen
- 6.10. Software-Wartung und Reengineering
 - 6.10.1. Wartungsprozess
 - 6.10.2. Standardrahmen für Wartungsprozesse. ISO/IEC 14764
 - 6.10.3. Prozessmodell für das Software-Reengineering
 - 6.10.4. *Reverse Engineering*

Modul 7. Fortgeschrittene Programmierung

- 7.1. Einführung in die objektorientierte Programmierung
 - 7.1.1. Einführung in die objektorientierte Programmierung
 - 7.1.2. Klassen-Design
 - 7.1.3. Einführung in UML für die Modellierung von Problemen
- 7.2. Beziehungen zwischen Klassen
 - 7.2.1. Abstraktion und Vererbung
 - 7.2.2. Fortgeschrittene Konzepte der Vererbung
 - 7.2.3. Polymorphismen
 - 7.2.4. Zusammensetzung und Aggregation
- 7.3. Einführung in Design Patterns für objektorientierte Probleme
 - 7.3.1. Was sind Entwurfsmuster?
 - 7.3.2. *Factory*-Muster
 - 7.3.3. *Singleton*-Muster
 - 7.3.4. *Observer*-Muster
 - 7.3.5. *Composite*-Muster
- 7.4. Ausnahmen
 - 7.4.1. Was sind Ausnahmen?
 - 7.4.2. Abfangen und Behandlung von Ausnahmen
 - 7.4.3. Einführen von Ausnahmen
 - 7.4.4. Erstellung von Ausnahmen

- 7.5. Benutzeroberflächen
 - 7.5.1. Einführung in Qt
 - 7.5.2. Positionierung
 - 7.5.3. Was sind Ereignisse?
 - 7.5.4. Ereignisse: Definition und Erfassung
 - 7.5.5. Entwicklung von Benutzeroberflächen
- 7.6. Einführung in die gleichzeitige Programmierung
 - 7.6.1. Einführung in die gleichzeitige Programmierung
 - 7.6.2. Der Prozess und das *Thread*-Konzept
 - 7.6.3. Interaktion zwischen Prozessen oder *Threads*
 - 7.6.4. *Threads* in C++
 - 7.6.5. Vor- und Nachteile der gleichzeitigen Programmierung
- 7.7. *Thread*-Verwaltung und Synchronisierung
 - 7.7.1. Lebenszyklus eines *Threads*
 - 7.7.2. Die Klasse *Thread*
 - 7.7.3. Planung des *Threads*
 - 7.7.4. Gruppen von *Threads*
 - 7.7.5. *Daemon Threads*
 - 7.7.6. Synchronisierung
 - 7.7.7. Verriegelungsmechanismen
 - 7.7.8. Kommunikationsmechanismen
 - 7.7.9. Monitore
- 7.8. Häufige Probleme bei der gleichzeitigen Programmierung
 - 7.8.1. Das Erzeuger-Verbraucher-Problem
 - 7.8.2. Das Problem von Lesern und Schreibern
 - 7.8.3. Das Philosophenproblem
- 7.9. Software-Dokumentation und -Tests
 - 7.9.1. Warum ist es wichtig, Software zu dokumentieren?
 - 7.9.2. Design-Dokumentation
 - 7.9.3. Verwendung von Tools zur Dokumentation

- 7.10. Software-Tests
 - 7.10.1. Einführung in das Testen von Software
 - 7.10.2. Arten von Tests
 - 7.10.3. Einheitstest
 - 7.10.4. Integrationstests
 - 7.10.5. Validierungstest
 - 7.10.6. Systemprüfung

Modul 8. Wiederverwendung von Software

- 8.1. Überblick über die Wiederverwendung von Software
 - 8.1.1. Was ist die Wiederverwendung von Software?
 - 8.1.2. Vor- und Nachteile der Wiederverwendung von Software
 - 8.1.3. Wichtigste Techniken der Wiederverwendung von Software
- 8.2. Einführung in die Entwurfsmuster
 - 8.2.1. Was ist ein Entwurfsmuster?
 - 8.2.2. Katalog der wichtigsten Entwurfsmuster
 - 8.2.3. Wie man Muster zur Lösung von Designproblemen einsetzt
 - 8.2.4. Wie man das beste Entwurfsmuster auswählt
- 8.3. Erzeugungsmuster
 - 8.3.1. Erzeugungsmuster
 - 8.3.2. *Abstract Factory*-Muster
 - 8.3.3. Beispielimplementierung des *Abstract Factory*-Musters
 - 8.3.4. *Builder*-Muster
 - 8.3.5. Beispielimplementierung des *Builder*
 - 8.3.6. *Abstract Factory*-Muster vs. *Builder*
- 8.4. Erzeugungsmuster (II)
 - 8.4.1. *Factory Method*-Muster
 - 8.4.2. *Factory Method* vs. *Abstract Factory*
 - 8.4.3. *Singleton*-Muster
- 8.5. Strukturmuster
 - 8.5.1. Strukturmuster
 - 8.5.2. *Adapter*-Muster
 - 8.5.3. *Bridge*-Muster

- 8.6. Strukturmuster (II)
 - 8.6.1. *Composite*-Muster
 - 8.6.2. *Decorator*-Muster
- 8.7. Strukturmuster (III)
 - 8.7.1. *Facade*-Muster
 - 8.7.2. *Proxy*-Muster
- 8.8. Verhaltensmuster
 - 8.8.1. Konzept der Verhaltensmuster
 - 8.8.2. Verhaltensmuster: Kette der Verantwortung
 - 8.8.3. Verhaltensmuster Ordnung
- 8.9. Verhaltensmuster (II)
 - 8.9.1. *Interpreter*-Muster
 - 8.9.2. *Iterator*-Muster
 - 8.9.3. Beobachtermuster
 - 8.9.4. Strategiemuster
- 8.10. *Frameworks*
 - 8.10.1. *Framework*-Konzepte
 - 8.10.2. Entwicklung mit *Frameworks*
 - 8.10.3. *Model View Controller*-Muster
 - 8.10.4. *Framework* für die Gestaltung grafischer Benutzeroberflächen
 - 8.10.5. *Frameworks* für die Entwicklung Web-Applikationen
 - 8.10.6. *Frameworks* für die Verwaltung der Objektpersistenz in Datenbanken

Modul 9. Künstliche Intelligenz und Wissensmodellierung

- 9.1. Einführung in Künstliche Intelligenz und Wissensmodellierung
 - 9.1.1. Kurze Geschichte der Künstlichen Intelligenz
 - 9.1.2. Künstliche Intelligenz heute
 - 9.1.3. Wissensmodellierung
- 9.2. Suche
 - 9.2.1. Allgemeine Suchkonzepte
 - 9.2.2. Uninformierte Suche
 - 9.2.3. Informierte Suche

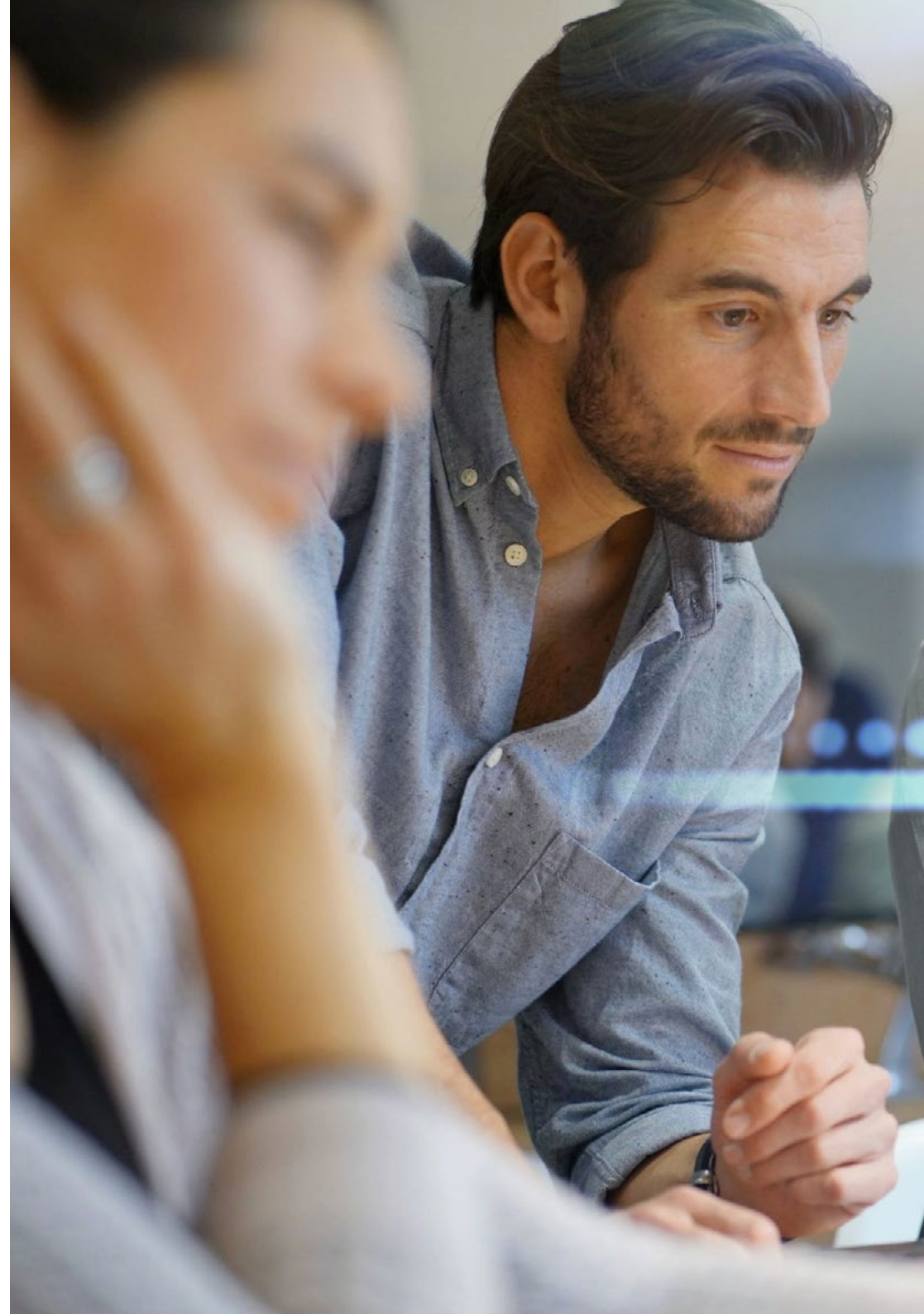
- 9.3. Boolesche Erfüllbarkeit, Erfüllbarkeit von Nebenbedingungen und automatische Planung
 - 9.3.1. Boolesche Erfüllbarkeit
 - 9.3.2. Probleme mit der Erfüllung von Einschränkungen
 - 9.3.3. Automatische Planung und PDDL
 - 9.3.4. Planung als heuristische Suche
 - 9.3.5. Planung mit SAT
- 9.4. Künstliche Intelligenz in Spielen
 - 9.4.1. Spieltheorie
 - 9.4.2. Minimax und Alpha-Beta-Beschneidung
 - 9.4.3. Simulation: Monte Carlo
- 9.5. Überwachtes und unüberwachtes Lernen
 - 9.5.1. Einführung in das maschinelle Lernen
 - 9.5.2. Klassifizierung
 - 9.5.3. Regression
 - 9.5.4. Validierung der Ergebnisse
 - 9.5.5. Gruppieren (*Clustering*)
- 9.6. Neuronale Netzwerke
 - 9.6.1. Biologische Grundlagen
 - 9.6.2. Berechnungsmodell
 - 9.6.3. Überwachte und nicht überwachte neuronale Netzwerke
 - 9.6.4. Einfaches Perzeptron
 - 9.6.5. Mehrschichtiges Perzeptron
- 9.7. Genetische Algorithmen
 - 9.7.1. Geschichte
 - 9.7.2. Biologische Grundlage
 - 9.7.3. Problem-Kodierung
 - 9.7.4. Erzeugung der Ausgangspopulation
 - 9.7.5. Hauptalgorithmus und genetische Operatoren
 - 9.7.6. Bewertung von Personen: Fitness
- 9.8. Thesauri, Vokabularien, Taxonomien
 - 9.8.1. Wortschatz
 - 9.8.2. Taxonomie
 - 9.8.3. Thesauri
 - 9.8.4. Ontologien

- 9.9. Wissensrepräsentation: Semantisches Web
 - 9.9.1. Semantisches Web
 - 9.9.2. Spezifizierungen: RDF, RDFS und OWL
 - 9.9.3. Schlussfolgerung/Begründung
 - 9.9.4. Verknüpfte Daten
- 9.10. Expertensysteme und DSS
 - 9.10.1. Experten-Systeme
 - 9.10.2. Systeme zur Entscheidungshilfe

Modul 10. Fortgeschrittene Softwaretechnik

- 10.1. Einführung in Agile Methoden
 - 10.1.1. Prozessmodelle und Methodologien
 - 10.1.2. Agilität und agile Prozesse
 - 10.1.3. Das Agile Manifest
 - 10.1.4. Einige agile Methoden
 - 10.1.5. Agil vs. Traditionell
- 10.2. Scrum
 - 10.2.1. Ursprünge und Philosophie von Scrum
 - 10.2.2. Scrum-Werte
 - 10.2.3. Scrum -Prozessablauf
 - 10.2.4. Scrum-Rollen
 - 10.2.5. Scrum-Artefakte
 - 10.2.6. Scrum-Ereignisse
 - 10.2.7. Anwenderberichte
 - 10.2.8. Scrum-Erweiterungen
 - 10.2.9. Agile Schätzungen
 - 10.2.10. Scrum-Skalierung
- 10.3. Extreme Programmierung
 - 10.3.1. Grundprinzip und Überblick über XP
 - 10.3.2. Der Lebenszyklus von XP
 - 10.3.3. Die fünf Grundwerte
 - 10.3.4. Die zwölf Kernpraktiken von XP
 - 10.3.5. Rollen der Teilnehmer
 - 10.3.6. Industrie XP
 - 10.3.7. Kritische Beurteilung von XP

- 10.4. Softwareentwicklung auf der Grundlage von Wiederverwendung
 - 10.4.1. Wiederverwendung von Software
 - 10.4.2. Stufen der Wiederverwendung von Codes
 - 10.4.3. Spezifische Techniken der Wiederverwendung
 - 10.4.4. Komponentenbasierte Entwicklung
 - 10.4.5. Vorteile und Probleme der Wiederverwendung
 - 10.4.6. Planung der Wiederverwendung
- 10.5. Systemarchitektur und Software-Entwurfsmuster
 - 10.5.1. Architektonisches Design
 - 10.5.2. Allgemeine architektonische Muster
 - 10.5.3. Fehlertolerante Architekturen
 - 10.5.4. Architekturen verteilter Systeme
 - 10.5.5. Entwurfsmuster
 - 10.5.6. Gamma-Muster
 - 10.5.7. Muster für die Interaktionsgestaltung
- 10.6. Architektur von Cloud-Anwendungen
 - 10.6.1. Grundlagen des *Cloud Computing*
 - 10.6.2. Qualität von Cloud-Anwendungen
 - 10.6.3. Architektonische Stile
 - 10.6.4. Entwurfsmuster
- 10.7. Software-Tests: TDD, ATDD und BDD
 - 10.7.1. Überprüfung und Validierung von Software
 - 10.7.2. Software-Tests
 - 10.7.3. *Test Driven Development* (TDD)
 - 10.7.4. *Acceptance Test Driven Development* (ATDD)
 - 10.7.5. *Behavior Driven Development* (BDD)
 - 10.7.6. BDD und *Cucumber*
- 10.8. Verbesserung von Software-Prozessen
 - 10.8.1. Verbesserung von Software-Prozessen
 - 10.8.2. Der Prozess der Prozessverbesserung
 - 10.8.3. Reifegradmodelle
 - 10.8.4. CMMI-Modell
 - 10.8.5. CMMI V2.0
 - 10.8.6. CMMI und Agile





- 10.9. Qualität von Softwareprodukten: SQuaRE
 - 10.9.1. Qualität der Software
 - 10.9.2. Qualitätsmodelle für Softwareprodukte
 - 10.9.3. ISO/IEC 25000-Familie
 - 10.9.4. ISO/IEC 25010: Qualitätsmodell und Qualitätsmerkmale
 - 10.9.5. ISO/IEC 25012: Datenqualität
 - 10.9.6. ISO/IEC 25020: Messung der Softwarequalität
 - 10.9.7. ISO/IEC 25022, 25023 und 25024: Messgrößen für die Software- und Datenqualität
 - 10.9.8. ISO/IEC 25040: Bewertung der Software
 - 10.9.9. Der Prozess der Zertifizierung
- 10.10. Einführung in DevOps
 - 10.10.1. DevOps-Konzept
 - 10.10.2. Grundlegende Praktiken

“

Eine einzigartige, wichtige und entscheidende Erfahrung, die Ihre berufliche Entwicklung fördert”

05 Methodik

Dieses Fortbildungsprogramm bietet eine andere Art des Lernens. Unsere Methodik wird durch eine zyklische Lernmethode entwickelt: **das Relearning**. Dieses Lehrsystem wird z. B. an den renommiertesten medizinischen Fakultäten der Welt angewandt und wird von wichtigen Publikationen wie dem **New England Journal of Medicine** als eines der effektivsten angesehen.





“

Entdecken Sie Relearning, ein System, das das herkömmliche lineare Lernen hinter sich lässt und Sie durch zyklische Lehrsysteme führt: eine Art des Lernens, die sich als äußerst effektiv erwiesen hat, insbesondere in Fächern, die Auswendiglernen erfordern"

Fallstudie zur Kontextualisierung aller Inhalte

Unser Programm bietet eine revolutionäre Methode zur Entwicklung von Fähigkeiten und Kenntnissen. Unser Ziel ist es, Kompetenzen in einem sich wandelnden, wettbewerbsorientierten und sehr anspruchsvollen Umfeld zu stärken.

“

Mit TECH werden Sie eine Art des Lernens erleben, die an den Grundlagen der traditionellen Universitäten auf der ganzen Welt rüttelt“



Sie werden Zugang zu einem Lernsystem haben, das auf Wiederholung basiert, mit natürlichem und progressivem Unterricht während des gesamten Lehrplans.



Der Student wird durch gemeinschaftliche Aktivitäten und reale Fälle lernen, wie man komplexe Situationen in realen Geschäftsumgebungen löst.

Eine innovative und andersartige Lernmethode

Dieses TECH-Programm ist ein von Grund auf neu entwickeltes, intensives Lehrprogramm, das die anspruchsvollsten Herausforderungen und Entscheidungen in diesem Bereich sowohl auf nationaler als auch auf internationaler Ebene vorsieht. Dank dieser Methodik wird das persönliche und berufliche Wachstum gefördert und ein entscheidender Schritt in Richtung Erfolg gemacht. Die Fallmethode, die Technik, die diesem Inhalt zugrunde liegt, gewährleistet, dass die aktuellste wirtschaftliche, soziale und berufliche Realität berücksichtigt wird.

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

Die Fallmethode ist das am weitesten verbreitete Lernsystem an den besten Informatikschulen der Welt, seit es sie gibt. Die Fallmethode wurde 1912 entwickelt, damit Jurastudenten das Recht nicht nur auf der Grundlage theoretischer Inhalte erlernen. Sie bestand darin, ihnen reale komplexe Situationen zu präsentieren, damit sie fundierte Entscheidungen treffen und Werturteile darüber fällen konnten, wie diese zu lösen sind. Sie wurde 1924 als Standardlehrmethode in Harvard etabliert.

Was sollte eine Fachkraft in einer bestimmten Situation tun? Mit dieser Frage konfrontieren wir Sie in der Fallmethode, einer handlungsorientierten Lernmethode.

Während des gesamten Kurses werden die Studenten mit mehreren realen Fällen konfrontiert. Sie müssen ihr gesamtes Wissen integrieren, recherchieren, argumentieren und ihre Ideen und Entscheidungen verteidigen.

Relearning Methodology

TECH kombiniert die Methodik der Fallstudien effektiv mit einem 100%igen Online-Lernsystem, das auf Wiederholung basiert und in jeder Lektion verschiedene didaktische Elemente kombiniert.

Wir ergänzen die Fallstudie mit der besten 100%igen Online-Lehrmethode: Relearning.

*Im Jahr 2019 erzielten wir die besten
Lernergebnisse aller spanischsprachigen
Online-Universitäten der Welt.*

Bei TECH lernen Sie mit einer hochmodernen Methodik, die darauf ausgerichtet ist, die Führungskräfte der Zukunft zu spezialisieren. Diese Methode, die an der Spitze der weltweiten Pädagogik steht, wird Relearning genannt.

Unsere Universität ist die einzige in der spanischsprachigen Welt, die für die Anwendung dieser erfolgreichen Methode zugelassen ist. Im Jahr 2019 ist es uns gelungen, die Gesamtzufriedenheit unserer Studenten (Qualität der Lehre, Qualität der Materialien, Kursstruktur, Ziele...) in Bezug auf die Indikatoren der besten spanischsprachigen Online-Universität zu verbessern.



In unserem Programm ist das Lernen kein linearer Prozess, sondern erfolgt in einer Spirale (lernen, verlernen, vergessen und neu lernen). Daher wird jedes dieser Elemente konzentrisch kombiniert. Mit dieser Methode wurden mehr als 650.000 Hochschulabsolventen mit beispiellosem Erfolg in so unterschiedlichen Bereichen wie Biochemie, Genetik, Chirurgie, internationales Recht, Managementfähigkeiten, Sportwissenschaft, Philosophie, Recht, Ingenieurwesen, Journalismus, Geschichte, Finanzmärkte und -instrumente fortgebildet. Dies alles in einem sehr anspruchsvollen Umfeld mit einer Studentenschaft mit hohem sozioökonomischem Profil und einem Durchschnittsalter von 43,5 Jahren.

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.

Nach den neuesten wissenschaftlichen Erkenntnissen der Neurowissenschaften wissen wir nicht nur, wie wir Informationen, Ideen, Bilder und Erinnerungen organisieren, sondern auch, dass der Ort und der Kontext, in dem wir etwas gelernt haben, von grundlegender Bedeutung dafür sind, dass wir uns daran erinnern und es im Hippocampus speichern können, um es in unserem Langzeitgedächtnis zu behalten.

Auf diese Weise sind die verschiedenen Elemente unseres Programms im Rahmen des so genannten Neurocognitive Context-Dependent E-Learning mit dem Kontext verbunden, in dem der Teilnehmer seine berufliche Praxis entwickelt.



Dieses Programm bietet die besten Lehrmaterialien, die sorgfältig für Fachleute aufbereitet sind:



Studienmaterial

Alle didaktischen Inhalte werden von den Fachleuten, 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 das audiovisuelle Format angewendet, um die Online-Arbeitsmethode von TECH zu schaffen. All dies mit den neuesten Techniken, die in jedem einzelnen der Materialien, die dem Studenten zur Verfügung gestellt werden, qualitativ hochwertige Elemente bieten.



Meisterklassen

Die Nützlichkeit der Expertenbeobachtung ist wissenschaftlich belegt.

Das sogenannte Learning from an Expert festigt das Wissen und das Gedächtnis und schafft Vertrauen für zukünftige schwierige Entscheidungen.



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.



Weitere Lektüren

Aktuelle Artikel, Konsensdokumente und internationale Leitfäden, u. a. In der virtuellen Bibliothek von TECH hat der Student Zugang zu allem, was er für seine Fortbildung benötigt.





Case Studies

Sie werden eine Auswahl der besten Fallstudien vervollständigen, die speziell für diese Qualifizierung ausgewählt wurden. Die Fälle werden von den besten Spezialisten der internationalen Szene präsentiert, analysiert und betreut.



Interaktive Zusammenfassungen

Das TECH-Team präsentiert die Inhalte auf attraktive und dynamische Weise in multimedialen Pillen, die Audios, Videos, Bilder, Diagramme und konzeptionelle Karten enthalten, um das Wissen zu vertiefen.

Dieses einzigartige Bildungssystem für die Präsentation multimedialer Inhalte wurde von Microsoft als "Europäische Erfolgsgeschichte" ausgezeichnet.



Testing & Retesting

Die Kenntnisse des Studenten werden während des gesamten Programms regelmäßig durch Bewertungs- und Selbsteinschätzungsaktivitäten und -übungen beurteilt und neu bewertet, so dass der Student überprüfen kann, wie er seine Ziele erreicht.



06

Qualifizierung

Der Privater Masterstudiengang in Erstellung von Schnittstellen und Netzwerkanwendungen garantiert neben der präzisesten und aktuellsten Fortbildung auch den Zugang zu einem von der TECH Technologischen Universität ausgestellten Diplom.



“

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

Dieser **Privater Masterstudiengang in Erstellung von Schnittstellen und Netzwerkanwendungen** enthält das vollständigste und aktuellste Programm auf dem Markt.

Sobald der Student die Prüfungen bestanden hat, erhält er/sie per Post* mit Empfangsbestätigung das entsprechende Diplom, ausgestellt von der **TECH Technologischen Universität**.

Das von **TECH Technologische Universität** ausgestellte Diplom drückt die erworbene Qualifikation aus und entspricht den Anforderungen, die in der Regel von Stellenbörsen, Auswahlprüfungen und Berufsbildungsausschüssen verlangt werden.

Titel: Privater Masterstudiengang in Erstellung von Schnittstellen und Netzwerkanwendungen

Anzahl der offiziellen Arbeitsstunden: **1.500 Std.**



*Haager Apostille. Für den Fall, dass der Student die Haager Apostille für sein Papierdiplom beantragt, wird TECH EDUCATION die notwendigen Vorkehrungen treffen, um diese gegen eine zusätzliche Gebühr zu beschaffen.

zukunft

gesundheit vertrauen menschen
erziehung information tutoren
garantie akkreditierung unterricht
institutionen technologie lernen
gemeinschaft verpflichtung
persönliche betreuung innovation
wissen gegenwart qualität
online-Ausbildung
entwicklung instituten
virtuelles Klassenzimmer

tech technologische
universität

Privater Masterstudiengang
Erstellung von Schnittstellen
und Netzwerkanwendungen

- » Modalität: online
- » Dauer: 12 Monate
- » Qualifizierung: TECH Technologische Universität
- » Aufwand: 16 Std./Woche
- » Zeitplan: in Ihrem eigenen Tempo
- » Prüfungen: online

Privater Masterstudiengang Erstellung von Schnittstellen und Netzwerkanwendungen