Weiterbildender Masterstudiengang

Informatik und Programmiersprachen

Akkredditierung/Mitgliedschaft

Association
for Computing
Machinery

tech global university

tech global university

Weiterbildender Masterstudiengang
Informatik und Programmiersprachen

- 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-informatik-programmiersprachen

Index

01

Präsentation des Programms

Die ständige Weiterentwicklung der künstlichen Intelligenz, des maschinellen Lernens und der Sprachverarbeitung hat die Welt der Informatik weltweit verändert. Diese Fortschritte erfordern Fachleute, die sowohl die theoretischen Grundlagen als auch die praktischen Anwendungen von Computersprachen verstehen.
Laut UNESCO werden in naher Zukunft 90% aller Arbeitsplätze fortgeschrittene digitale Kompetenzen erfordern. Diese Tatsache hat den Bedarf an spezialisierten Studiengängen erhöht, die Kenntnisse in Algorithmen, Datenstrukturen und Sprachsemantik vermitteln. In diesem Zusammenhang präsentiert TECH einen innovativen, vollständig online absolvierten Studiengang in Informatik und Sprachen, der darauf ausgerichtet ist, eine solide und aktuelle akademische Ausbildung in einem digitalen und flexiblen Umfeld zu bieten.

Ein umfassendes, zu 100% online durchgeführtes Programm, das exklusiv von TECH angeboten wird und durch unsere Mitgliedschaft in der Association for Computing Machinery eine internationale Perspektive bietet"

In einem technologischen Umfeld, das zunehmend von Automatisierung, künstlicher Intelligenz und natürlicher Sprachverarbeitung geprägt ist, ist das Verständnis, wie Computersprachen aufgebaut sind, interpretiert und optimiert werden, zu einer zentralen Notwendigkeit geworden, um in vielen Bereichen des Wissens und der Industrie Fortschritte zu erzielen. Die Interaktion zwischen Mensch und Maschine, die Effizienz von Computersystemen und die Entwicklung neuer digitaler Werkzeuge hängen in hohem Maße von der Beherrschung dieser Sprachen ab.

Dieses Programm bietet einen umfassenden Überblick über die Grundlagen der Informatik und der Programmiersprachen und vermittelt Fachkenntnisse, die das logische Denken, die Lösung komplexer Probleme und die Entwicklung innovativer Lösungen fördern. Es werden Themen wie formale Semantik, Sprachtheorie, Automaten, Kompilierung und andere grundlegende Elemente behandelt, die für das Verständnis der inneren Funktionsweise von Computersystemen erforderlich sind.

Darüber hinaus eröffnen sich durch die Beherrschung dieser Inhalte Möglichkeiten in hochdynamischen Bereichen wie künstliche Intelligenz, Entwicklung von Programmiersprachen, Quantencomputing oder *Software-Engineering*. Diese Spezialisierung stellt einen strategischen Schritt für diejenigen dar, die ihre Karriere im akademischen, technischen oder beruflichen Bereich vorantreiben möchten, insbesondere in Kontexten, in denen Profile mit einer soliden theoretischen Grundlage und analytischen Fähigkeiten geschätzt werden. Die erworbene konzeptionelle Tiefe ermöglicht es nicht nur, bestehende Technologien anzuwenden, sondern auch die Technologien der Zukunft zu entwerfen.

Die Online-Modalität dieses Programms ermöglicht den Zugriff auf aktuelle Inhalte von hohem akademischem Niveau von jedem Ort aus, ohne dabei Abstriche bei der Qualität oder den Anforderungen zu machen. Diese Flexibilität erleichtert die Vereinbarkeit mit anderen beruflichen oder akademischen Projekten und fördert selbstständiges, rigoroses und auf kritische Analyse ausgerichtetes Lernen. Durch eine interaktive Plattform, multimediale Materialien und die Begleitung durch Experten wird eine bereichernde Erfahrung garantiert, die den aktuellen Anforderungen des digitalen und akademischen Sektors gerecht wird. Darüber hinaus wird ein renommierter internationaler Gastdirektor 10 ausführliche *Masterclasses* anbieten.

Da TECH Mitglied der **Association for Computing Machinery (ACM)** ist, hat der Student außerdem Zugang zu exklusiven und aktuellen Ressourcen wie wissenschaftlichen Publikationen, Fachkursen und internationalen Konferenzen. Darüber hinaus hat er die Möglichkeit, sein Netzwerk zu erweitern und Kontakte zu Experten aus den Bereichen Technologie, künstliche Intelligenz, Datenwissenschaft und anderen wichtigen Disziplinen der Branche zu knüpfen.

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

- Die Entwicklung von Fallstudien, die von Experten für Informatik und Programmiersprachen vorgestellt werden
- Der anschauliche, schematische und äußerst praxisnahe Inhalt vermittelt alle für die berufliche Praxis unverzichtbaren wissenschaftlichen und praktischen Informationen
- Die praktischen Übungen, bei denen der Selbstbewertungsprozess zur Verbesserung des Lernens durchgeführt 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 Zugriffs auf die Inhalte von jedem festen oder tragbaren Gerät mit Internetanschluss

Ein renommierter internationaler Gastdirektor wird zehn exklusive Masterclasses zu den neuesten Trends in den Bereichen Informatik und Programmiersprachen anbieten“

Sie werden sich eingehend mit der Theorie der Automaten und formalen Sprachen befassen, um zu verstehen, wie Computersysteme intern funktionieren“

Sie werden mit fortschrittlichen Datenstrukturen arbeiten, um effiziente, skalierbare und leistungsorientierte Lösungen zu entwickeln.

Sie werden komplexe Algorithmen entwerfen, um anspruchsvolle rechnerische Probleme in wichtigen Technologiebereichen zu lösen.

Zu den Dozenten gehören Fachleute aus dem Bereich der Informatik, 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.

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“

Die beste Online-Universität der Welt laut FORBES

Das renommierte, auf Wirtschaft und Finanzen spezialisierte Magazin Forbes hat TECH als „beste Online-Universität der Welt" ausgezeichnet. Dies wurde kürzlich in einem Artikel in der digitalen Ausgabe des Magazins festgestellt, in dem die Erfolgsgeschichte dieser Einrichtung „dank ihres akademischen Angebots, der Auswahl ihrer Lehrkräfte und einer innovativen Lernmethode, die auf die Ausbildung der Fachkräfte der Zukunft abzielt", hervorgehoben wird.

Die besten internationalen Top-Lehrkräfte

Der Lehrkörper der TECH besteht aus mehr als 6.000 Professoren von höchstem internationalen Ansehen. Professoren, Forscher und Führungskräfte multinationaler Unternehmen, darunter Isaiah Covington, Leistungstrainer der Boston Celtics, Magda Romanska, leitende Forscherin am Harvard MetaLAB, Ignacio Wistuba, Vorsitzender der Abteilung für translationale Molekularpathologie am MD Anderson Cancer Center, und D.W. Pine, Kreativdirektor des TIME Magazine, um nur einige zu nennen.

Die größte digitale Universität der Welt

TECH ist die weltweit größte digitale Universität. Wir sind die größte Bildungseinrichtung mit dem besten und umfangreichsten digitalen Bildungskatalog, der zu 100% online ist und die meisten Wissensgebiete abdeckt. Wir bieten weltweit die größte Anzahl eigener Abschlüsse sowie offizieller Grund- und Aufbaustudiengänge an. Insgesamt sind wir mit mehr als 14.000 Hochschulabschlüssen in elf verschiedenen Sprachen die größte Bildungseinrichtung der Welt.

Die umfassendsten Lehrpläne in der Universitätslandschaft

TECH bietet die vollständigsten Lehrpläne in der Universitätslandschaft an, mit Lehrplänen, die grundlegende Konzepte und gleichzeitig die wichtigsten wissenschaftlichen Fortschritte in ihren spezifischen wissenschaftlichen Bereichen abdecken. Darüber hinaus werden diese Programme ständig aktualisiert, um den Studenten die akademische Avantgarde und die gefragtesten beruflichen Kompetenzen zu garantieren. Auf diese Weise verschaffen die Abschlüsse der Universität ihren Absolventen einen bedeutenden Vorteil, um ihre Karriere erfolgreich voranzutreiben.

Eine einzigartige Lernmethode

TECH ist die erste Universität, die *Relearning* in allen ihren Studiengängen einsetzt. Es handelt sich um die beste Online-Lernmethodik, die mit internationalen Qualitätszertifikaten renommierter Bildungseinrichtungen ausgezeichnet wurde. Darüber hinaus wird dieses disruptive akademische Modell durch die „Fallmethode" ergänzt, wodurch eine einzigartige Online-Lehrstrategie entsteht. Es werden auch innovative Lehrmittel eingesetzt, darunter ausführliche Videos, Infografiken und interaktive Zusammenfassungen.

Die offizielle Online-Universität der NBA

TECH ist die offizielle Online-Universität der NBA. Durch eine Vereinbarung mit der größten Basketball-Liga bietet sie ihren Studenten exklusive Universitätsprogramme sowie eine breite Palette von Bildungsressourcen, die sich auf das Geschäft der Liga und andere Bereiche der Sportindustrie konzentrieren. Jedes Programm hat einen einzigartig gestalteten Lehrplan und bietet außergewöhnliche Gastredner: Fachleute mit herausragendem Sporthintergrund, die ihr Fachwissen zu den wichtigsten Themen zur Verfügung stellen.

Führend in Beschäftigungsfähigkeit

TECH ist es gelungen, die führende Universität im Bereich der Beschäftigungsfähigkeit zu werden. 99% der Studenten finden innerhalb eines Jahres nach Abschluss eines Studiengangs der Universität einen Arbeitsplatz in dem von ihnen studierten Fachgebiet. Ähnlich viele erreichen einen unmittelbaren Karriereaufstieg. All dies ist einer Studienmethodik zu verdanken, die ihre Wirksamkeit auf den Erwerb praktischer Fähigkeiten stützt, die für die berufliche Entwicklung absolut notwendig sind.

Google Partner Premier

Der amerikanische Technologieriese hat TECH mit dem Logo Google Partner Premier ausgezeichnet. Diese Auszeichnung, die nur 3% der Unternehmen weltweit erhalten, unterstreicht die effiziente, flexible und angepasste Erfahrung, die diese Universität den Studenten bietet. Die Anerkennung bestätigt nicht nur die maximale Präzision, Leistung und Investition in die digitalen Infrastrukturen der TECH, sondern positioniert diese Universität auch als eines der modernsten Technologieunternehmen der Welt.

Die von ihren Studenten am besten bewertete Universität

Die Studenten haben TECH auf den wichtigsten Bewertungsportalen als die am besten bewertete Universität der Welt eingestuft, mit einer Höchstbewertung von 4,9 von 5 Punkten, die aus mehr als 1.000 Bewertungen hervorgeht. Diese Ergebnisse festigen die Position der TECH als internationale Referenzuniversität und spiegeln die Exzellenz und die positiven Auswirkungen ihres Bildungsmodells wider.

03

Lehrplan

In einem Umfeld, in dem Fortschritte in den Bereichen Programmiersprachen, Automatentheorie und formale Semantik die Entwicklung neuer Technologien vorantreiben, ist die eingehende Untersuchung dieser Grundlagen von entscheidender Bedeutung für Innovationen in Bereichen wie Quantencomputing, künstliche Intelligenz und Cybersicherheit. Aus diesem Grund umfasst dieser Lehrplan aktuelle und strategisch strukturierte Inhalte, die ein Verständnis der internen Logik von Computersystemen ermöglichen. Auf diese Weise wird eine kritische und fortschrittliche Sichtweise auf die Funktionsweise von Computern gefördert, die im Einklang mit den digitalen Veränderungen steht, die die Standards der Informatik weltweit neu definieren.

Sie werden domänenspezifische Sprachen entwerfen und implementieren, die an konkrete technische Anforderungen angepasst sind“

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. ADT-Liste: Definition
- 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. Sortieralgorithmen
- 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. Binärbäume
- 3.4.3. Baumtraversierungen
- 3.4.4. Darstellung von Ausdrücken
- 3.4.5. Binäre Suchbäume
- 3.4.6. Balancierte Binärbäume

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. Fortgeschrittener Algorithmusentwurf

4.1. Analyse von rekursiven und *Divide-and-Conquer*-Algorithmen
- 4.1.1. Aufstellen und Lösen von homogenen und nichthomogenen Rekursionsgleichungen
- 4.1.2. Allgemeine Beschreibung der *Divide-and-Conquer*-Strategie

4.2. Amortisierte Analyse
- 4.2.1. Aggregierte Analyse
- 4.2.2. Die Buchhaltungsmethode
- 4.2.3. Die Potentialmethode

4.3. Dynamische Programmierung und Algorithmen für NP-Probleme
- 4.3.1. Merkmale der dynamischen Programmierung
- 4.3.2. Umkehr: *Backtracking*
- 4.3.3. Verzweigung und Beschneidung

4.4. Kombinatorische Optimierung
- 4.4.1. Problemdarstellung
- 4.4.2. 1D-Optimierung

4.5. Randomisierte Algorithmen
- 4.5.1. Beispiele für randomisierte Algorithmen
- 4.5.2. Das Buffonsche Theorem
- 4.5.3. Monte-Carlo-Algorithmus
- 4.5.4. Las-Vegas-Algorithmus

4.6. Lokale Suche und Kandidatensuche
- 4.6.1. *Gradient Ascent*
- 4.6.2. *Hill Climbing*
- 4.6.3. *Simulated Annealing*
- 4.6.4. *Tabu Search*
- 4.6.5. Kandidatensuche

4.7. Formale Verifizierung von Programmen
- 4.7.1. Spezifikation von funktionalen Abstraktionen
- 4.7.2. Die Sprache der Prädikatenlogik erster Stufe
- 4.7.3. Hoare's formales System

4.8. Verifizierung von iterativen Programmen
- 4.8.1. Regeln des formalen Hoare-Systems
- 4.8.2. Konzept der invarianten Iterationen

4.9. Numerische Methoden
- 4.9.1. Die Methode der Halbierung
- 4.9.2. Newton-Raphson-Methode
- 4.9.3. Die Sekantenmethode

4.10. Parallele Algorithmen
- 4.10.1. Parallele binäre Operationen
- 4.10.2. Parallele Operationen mit Diagrammen
- 4.10.3. Parallelität in Teilen und Erobern
- 4.10.4. Parallelität in der dynamischen Programmierung

Modul 5. Fortgeschrittene Programmierung

5.1. Einführung in die objektorientierte Programmierung
- 5.1.1. Einführung in die objektorientierte Programmierung
- 5.1.2. Klassendesign
- 5.1.3. Einführung in UML für die Modellierung von Problemen

5.2. Beziehungen zwischen Klassen
- 5.2.1. Abstraktion und Vererbung
- 5.2.2. Fortgeschrittene Konzepte der Vererbung
- 5.2.3. Polymorphismen
- 5.2.4. Zusammensetzung und Aggregation

5.3. Einführung in Entwurfsmuster für objektorientierte Probleme
- 5.3.1. Was sind Entwurfsmuster
- 5.3.2. *Factory*-Muster
- 5.3.3. *Singleton*-Muster
- 5.3.4. *Observer*-Muster
- 5.3.5. *Composite*-Muster

5.4. Ausnahmen
- 5.4.1. Was sind Ausnahmen?
- 5.4.2. Abfangen und Behandlung von Ausnahmen
- 5.4.3. Auslösen von Ausnahmen
- 5.4.4. Erstellung von Ausnahmen

5.5. Benutzeroberflächen
- 5.5.1. Einführung in Qt
- 5.5.2. Positionierung
- 5.5.3. Was sind Ereignisse?
- 5.5.4. Ereignisse: Definition und Erfassung
- 5.5.5. Entwicklung von Benutzeroberflächen

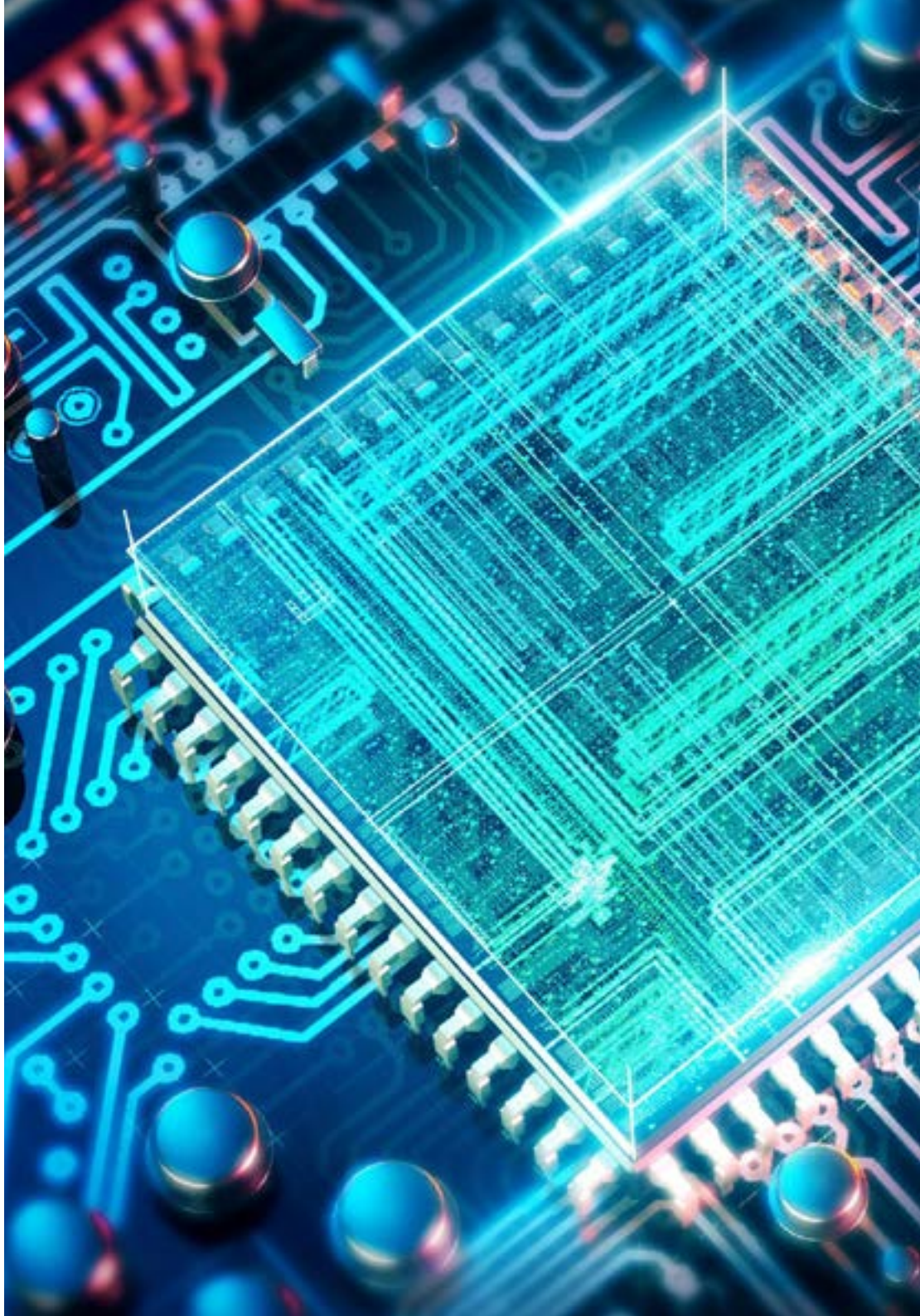
5.6. Einführung in die nebenläufige Programmierung
- 5.6.1. Einführung in die nebenläufige Programmierung
- 5.6.2. Das Konzept von Prozessen und *Threads*
- 5.6.3. Interaktion zwischen Prozessen oder *Threads*
- 5.6.4. *Threads* in C++
- 5.6.5. Vor- und Nachteile der nebenläufigen Programmierung

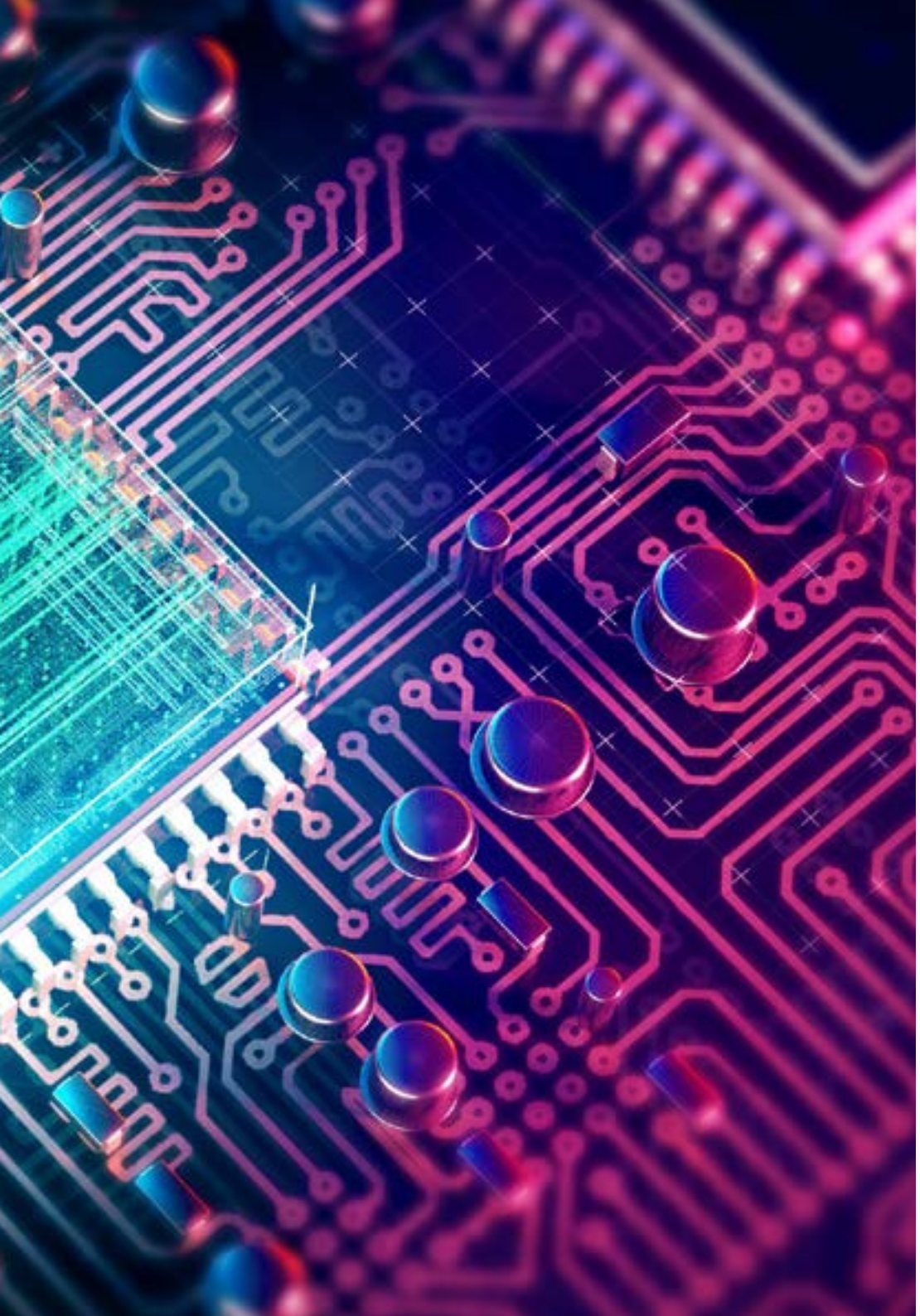
5.7. *Thread*-Verwaltung und Synchronisation
- 5.7.1. Lebenszyklus eines *Threads*
- 5.7.2. Die Klasse *Thread*
- 5.7.3. *Thread*-Planung
- 5.7.4. *Thread*-Gruppen
- 5.7.5. *Daemon*-Threads
- 5.7.6. Synchronisation
- 5.7.7. Sperrmechanismen
- 5.7.8. Kommunikationsmechanismen
- 5.7.9. Monitore

5.8. Häufige Probleme in der nebenläufigen Programmierung
- 5.8.1. Das Erzeuger-Verbraucher-Problem
- 5.8.2. Das Leser-Schreiber-Problem
- 5.8.3. Das Problem der speisenden Philosophen

5.9. Softwaredokumentation und -tests
- 5.9.1. Warum ist es wichtig, Software zu dokumentieren?
- 5.9.2. Entwurfsdokumentation
- 5.9.3. Verwendung von Tools zur Dokumentation

5.10. Softwaretests
- 5.10.1. Einführung in Softwaretests
- 5.10.2. Arten von Tests
- 5.10.3. Unit-Test
- 5.10.4. Integrationstests
- 5.10.5. Validierungstest
- 5.10.6. Systemprüfung





Modul 6. Theoretische Informatik

6.1. Verwendete mathematische Konzepte
- 6.1.1. Einführung in die Aussagenlogik
- 6.1.2. Relationenlehre
- 6.1.3. Abzählbare und überabzählbare Mengen

6.2. Formale Sprachen und Grammatiken und Einführung in Turingmaschinen
- 6.2.1. Formale Sprachen und Grammatiken
- 6.2.2. Problem der Entscheidung
- 6.2.3. Die Turingmaschine

6.3. Erweiterungen für Turingmaschinen, beschränkte Turingmaschinen und Computer
- 6.3.1. Programmiertechniken für Turingmaschinen
- 6.3.2. Erweiterungen für Turingmaschinen
- 6.3.3. Beschränkte Turingmaschinen
- 6.3.4. Turingmaschinen und Computer

6.4. Unentscheidbarkeit
- 6.4.1. Nicht rekursiv aufzählbare Sprache
- 6.4.2. Ein unentscheidbares rekursiv aufzählbares Problem

6.5. Andere unentscheidbare Probleme
- 6.5.1. Unentscheidbare Probleme für Turingmaschinen
- 6.5.2. Postkorrespondenz-Problem (PCP)

6.6. Unlösbare Probleme
- 6.6.1. Die Klassen P und NP
- 6.6.2. Ein NP-vollständiges Problem
- 6.6.3. Problem der eingeschränkten Erfüllbarkeit
- 6.6.4. Andere NP-vollständige Probleme

6.7. *Co-NP- und PSPACE*-Probleme
- 6.7.1. Komplementär zu NP-Sprachen
- 6.7.2. Probleme, die mit polynomiellem Speicher lösbar sind
- 6.7.3. *PSPACE*-vollständige Probleme

6.8. Sprachklassen mit Randomisierung
- 6.8.1. Probabilistische Turingmaschinen
- 6.8.2. Die Klassen RP und ZPP
- 6.8.3. Primzahltest
- 6.8.4. Komplexität des Primzahltests

6.9. Andere Klassen und Grammatiken
- 6.9.1. Probabilistische endliche Automaten
- 6.9.2. Zelluläre Automaten
- 6.9.3. McCulloch-Pitts-Zellen
- 6.9.4. Lindenmayer-Grammatiken

6.10. Fortgeschrittene Rechensysteme
- 6.10.1. *Membrane Computing*: P-Systeme
- 6.10.2. DNA-Computing
- 6.10.3. Quantencomputing

Modul 7. Automatentheorie und formale Sprachen

7.1. Einführung in die Automatentheorie
- 7.1.1. Warum Automatentheorie studieren?
- 7.1.2. Einführung in formale Beweise
- 7.1.3. Andere Formen des Nachweises
- 7.1.4. Mathematische Induktion
- 7.1.5. Alphabete, Zeichenketten und Sprachen

7.2. Deterministische endliche Automaten (DEA)
- 7.2.1. Einführung in endliche Automaten
- 7.2.2. Deterministische endliche Automaten (DEA)

7.3. Nichtdeterministische endliche Automaten
- 7.3.1. Nichtdeterministische endliche Automaten
- 7.3.2. Äquivalenz zwischen DEA und NEA
- 7.3.3. Endliche Automaten mit Є-Übergängen

7.4. Reguläre Sprachen und reguläre Ausdrücke (I)
- 7.4.1. Reguläre Sprachen und reguläre Ausdrücke
- 7.4.2. Endliche Automaten und reguläre Ausdrücke

7.5. Reguläre Sprachen und reguläre Ausdrücke (II)
- 7.5.1. Umwandlung regulärer Ausdrücke in Automaten
- 7.5.2. Anwendungen regulärer Ausdrücke
- 7.5.3. Algebra der regulären Ausdrücke

7.6. *Pumping-Lemma* und Abschluss von regulären Sprachen
- 7.6.1. *Pumping-Lemma*
- 7.6.2. Abschlusseigenschaften von regulären Sprachen

7.7. Äquivalenz und Minimierung von Automaten
- 7.7.1. Äquivalenz endlicher Automaten
- 7.7.2. Minimierung endlicher Automaten

7.8. Kontextfreie Grammatiken (CFG)
- 7.8.1. Kontextfreie Grammatiken
- 7.8.2. Ableitungsbäume
- 7.8.3. Anwendungen kontextfreier Grammatiken
- 7.8.4. Mehrdeutigkeit in Grammatiken und Sprachen

7.9. Kellerautomaten und kontextfreie Grammatiken
- 7.9.1. Definition von Kellerautomaten
- 7.9.2. Von einem Kellerautomaten akzeptierte Sprachen
- 7.9.3. Äquivalenz zwischen Kellerautomaten und kontextfreien Grammatiken
- 7.9.4. Deterministischer Kellerautomat

7.10. Normalformen, *Pumping-Lemma* von CFGs und Eigenschaften kontextfreier Sprachen
- 7.10.1. Normalformen kontextfreier Grammatiken
- 7.10.2. *Pumping-Lemma*
- 7.10.3. Abschlusseigenschaften der Sprachen
- 7.10.4. Entscheidungseigenschaften kontextfreier Sprachen

Modul 8. Sprachprozessoren

8.1. Einführung in den Kompilierungsprozess
- 8.1.1. Zusammenstellung und Interpretation
- 8.1.2. Laufzeitumgebung eines Compilers
- 8.1.3. Analyseprozess
- 8.1.4. Syntheseprozess

8.2. Lexikalischer Analysator
- 8.2.1. Was ist ein lexikalischer Analysator?
- 8.2.2. Implementierung des lexikalischen Analysators
- 8.2.3. Semantische Aktionen
- 8.2.4. Fehlerbehebung
- 8.2.5. Implementierungsaspekte

8.3. Syntaxanalyse
- 8.3.1. Was ist ein *Parser*?
- 8.3.2. Vorläufige Konzepte
- 8.3.3. *Top-down-Parser*
- 8.3.4. *Bottom-up-Parser*

8.4. *Top-down-* und *Bottom-up*-Syntaxanalyse
- 8.4.1. LL(1)-*Parser*
- *8.4.2. LR(0)-Parser*
- 8.4.3. Beispiel eines *Parser*

8.5. Fortgeschrittene Bottom-up-Syntaxanalyse
- 8.5.1. SLR-*Parser*
- 8.5.2. LR(1)-*Parser*
- 8.5.3. LR(k)-*Parser*
- 8.5.4. LALR-*Parser*

8.6. Semantische Analyse (I)
- 8.6.1. Syntaxgesteuerte Übersetzung
- 8.6.2. Symboltabelle

8.7. Semantische Analyse (II)
- 8.7.1. Typüberprüfung
- 8.7.2. Das Typ-Subsystem
- 8.7.3. Typäquivalenz und Typkonversionen

8.8. Codegenerierung und Laufzeitumgebung
- 8.8.1. Entwurfsaspekte
- 8.8.2. Laufzeitumgebung
- 8.8.3. Speicherorganisation
- 8.8.4. Speicherzuweisung

8.9. Zwischencodegenerierung
- 8.9.1. Syntaxgesteuerte Übersetzung
- 8.9.2. Zwischendarstellungen
- 8.9.3. Beispiele für Übersetzungen

8.10. Codeoptimierung
- 8.10.1. Registerzuweisung
- 8.10.2. Eliminierung toter Zuweisungen
- 8.10.3. Ausführung zur Kompilierzeit
- 8.10.4. Neuordnung von Ausdrücken
- 8.10.5. Schleifenoptimierung

Modul 9. Computergrafik und Visualisierung

9.1. Farbtheorie
- 9.1.1. Eigenschaften von Licht
- 9.1.2. Farbenmodelle
- 9.1.3. Der CIE-Standard
- 9.1.4. *Profiling*

9.2. Ausgabe-Primitive
- 9.2.1. Der Videotreiber
- 9.2.2. Algorithmen zum Zeichnen von Linien
- 9.2.3. Algorithmen zum Zeichnen von Kreisen
- 9.2.4. Algorithmen zum Füllen

9.3. 2D-Transformationen und 2D-Koordinatensysteme und 2D-Clipping
- 9.3.1. Geometrische Grundtransformationen
- 9.3.2. Homogene Koordinaten
- 9.3.3. Inverse Transformation
- 9.3.4. Komposition von Transformationen
- 9.3.5. Andere Transformationen
- 9.3.6. Koordinatenverschiebung
- 9.3.7. 2D-Koordinatensysteme
- 9.3.8. Koordinatenverschiebung
- 9.3.9. Normalisierung
- 9.3.10. Algorithmen zum Trimmen

9.4. 3D-Transformationen
- 9.4.1. Übertragung
- 9.4.2. Rotation
- 9.4.3. Skalierung
- 9.4.4. Reflexion
- 9.4.5. Scherung

9.5. Anzeige und Änderung von 3D-Koordinaten
- 9.5.1. 3D-Koordinatensysteme
- 9.5.2. Sichtbarkeit
- 9.5.3. Koordinatenwechsel
- 9.5.4. Projektion und Normalisierung

9.6. Projektion und 3D-Clipping
- 9.6.1. Orthogonale Projektion
- 9.6.2. Schrägparallele Projektion
- 9.6.3. Perspektivische Projektion
- 9.6.4. 3D-Clipping-Algorithmen

9.7. Entfernen von verdeckten Flächen
- 9.7.1. *Back-Face Removal*
- 9.7.2. *Z-buffer*
- 9.7.3. Painter-Algorithmus
- 9.7.4. Warnock-Algorithmus
- 9.7.5. Erkennung verdeckter Linien

9.8. Interpolation und parametrische Kurven
- 9.8.1. Interpolation und polynomielle Approximation
- 9.8.2. Parametrische Darstellung
- 9.8.3. Lagrange-Polynom
- 9.8.4. Natürliche kubische *Splines*
- 9.8.5. Basis-Funktionen
- 9.8.6. Matrix-Darstellung

9.9. Bézier-Kurven
- 9.9.1. Algebraische Konstruktion
- 9.9.2. Matrix-Formular
- 9.9.3. Zusammensetzung
- 9.9.4. Geometrische Konstruktion
- 9.9.5. Algorithmus zum Zeichnen

9.10. *B-Splines*
- 9.10.1. Das Problem der lokalen Kontrolle
- 9.10.2. Gleichmäßige kubische *B-Splines*
- 9.10.3. Basisfunktionen und Kontrollpunkte
- 9.10.4. Ableitung zum Ursprung und Multiplizität
- 9.10.5. Matrix-Darstellung
- 9.10.6. Ungleichmäßige *B-Splines*

Modul 10. Bio-inspiriertes Computing

10.1. Einführung in das bio-inspirierte Computing
- 10.1.1. Einführung in das bio-inspirierte Computing

10.2. Algorithmen zur sozialen Anpassung
- 10.2.1. Bio-inspiriertes Computing auf der Grundlage von Ameisenkolonien
- 10.2.2. Varianten von Ameisenkolonie-Algorithmen
- 10.2.3. Cloud-basiertes Computing auf Partikelebene

10.3. Genetische Algorithmen
- 10.3.1. Allgemeine Struktur
- 10.3.2. Implementierungen der wichtigsten Operatoren

10.4. Strategien zur Exploration und Ausbeutung des Suchraums für genetische Algorithmen
- 10.4.1. CHC-Algorithmus
- 10.4.2. Multimodale Probleme

10.5. Evolutionäre Berechnungsmodelle (I)
- 10.5.1. Evolutionäre Strategien
- 10.5.2. Evolutionäre Programmierung
- 10.5.3. Algorithmen auf der Grundlage der differentiellen Evolution

10.6. Evolutionäre Berechnungsmodelle (II)
- 10.6.1. Evolutionäre Modelle auf der Grundlage der Schätzung von Verteilungen (EDA)
- 10.6.2. Genetische Programmierung

10.7. Evolutionäre Programmierung angewandt auf Lernprobleme
- 10.7.1. Regelbasiertes Lernen
- 10.7.2. Evolutionäre Methoden bei Instanzauswahlproblemen

10.8. Mehrzielprobleme
- 10.8.1. Konzept der Dominanz
- 10.8.2. Anwendung evolutionärer Algorithmen auf Mehrzielprobleme

10.9. Neuronale Netze (I)
- 10.9.1. Einführung in neuronale Netze
- 10.9.2. Praktisches Beispiel mit neuronalen Netzen

10.10. Neuronale Netze
- 10.10.1. Anwendungsfälle neuronaler Netze in der medizinischen Forschung
- 10.10.2. Anwendungsfälle neuronaler Netze in der Wirtschaft
- 10.10.3. Anwendungsfälle neuronaler Netze in der künstlichen Bildverarbeitung

Sie werden Fähigkeiten zur formalen und semantischen Analyse von Programmen entwickeln und die Effizienz der Software erheblich verbessern“

04

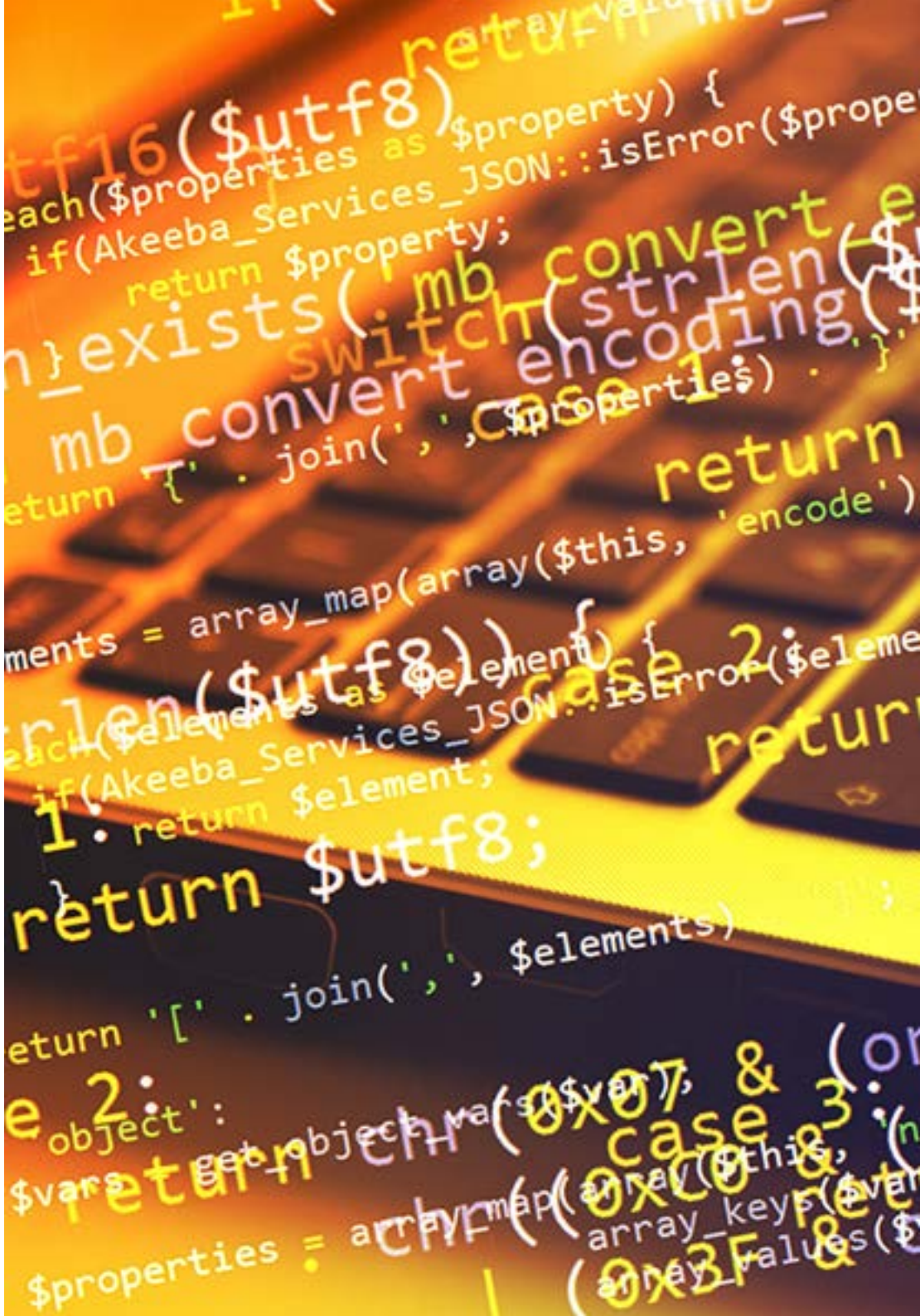
Lehrziele

Dieser weiterbildende Masterstudiengang zielt darauf ab, die theoretischen und praktischen Grundlagen zu vertiefen, die dem Entwurf, der Analyse und der Optimierung von Computersprachen zugrunde liegen. Zu diesem Zweck kombiniert er das Studium grammatikalischer und semantischer Strukturen sowie Computermodelle mit Werkzeugen, die bei der Entwicklung intelligenter Systeme zum Einsatz kommen. Darüber hinaus fördert er abstraktes Denken und die rigorose Lösung komplexer Probleme, die in technologisch anspruchsvollen Umgebungen von entscheidender Bedeutung sind. Dank dieser ganzheitlichen Perspektive wird die Innovationsfähigkeit in aufstrebenden Bereichen wie der Verarbeitung natürlicher Sprache, formalen Sprachen und der Architektur von Compilern gefördert.

Sie werden das computergestützte Denken in technologische Innovationsprojekte integrieren, wodurch Sie in der Lage sein werden, verschiedenen Unternehmen fortschrittliche Lösungen anzubieten“

Allgemeine Ziele

- Umfassendes Verstehen der theoretischen Grundlagen von Programmiersprachen und ihrer Anwendung in fortgeschrittenen und spezialisierten Computerumgebungen
- Kritisches Analysieren der formalen Strukturen, die dem Entwurf und der Entwicklung von Computersprachen in verschiedenen Kontexten zugrunde liegen
- Kompetentes Anwenden von Computermodellen zur effizienten Lösung komplexer Probleme im Bereich der aktuellen Informatik
- Beherrschen der wesentlichen Prinzipien der formalen Semantik und ihrer direkten Beziehung zur präzisen Ausführung von Computerprogrammen
- Gründliches Untersuchen von Automaten, Grammatiken und formalen Sprachen aus einer umfassenden computerwissenschaftlichen und wissenschaftlichen Perspektive
- Integrieren grundlegender Kenntnisse der Computertheorie mit aktuellen und zukünftigen technologischen Entwicklungen in diesem Bereich
- Systematisches Bewerten der Leistung und Effizienz verschiedener Sprachen, Compiler und fortgeschrittener Entwicklungsumgebungen
- Entwerfen innovativer Computerlösungen auf der Grundlage einer detaillierten Analyse logischer, syntaktischer und semantischer Strukturen
- Entwickeln spezialisierter Kompetenzen für die angewandte Forschung im Bereich der Informatik und der formalen Sprachen
- Kritisches Interpretieren der Auswirkungen von Computersprachen auf die ständige Weiterentwicklung moderner intelligenter Systeme



Spezifische Ziele

Modul 1. Grundlagen der Programmierung

- Identifizieren der grundlegenden Elemente einer Programmiersprache
- Anwenden von Kontrollstrukturen bei der Problemlösung
- Verwenden von Variablen, Operatoren und Funktionen in einfachen Programmen
- Entwickeln von strukturierten Programmen nach bewährten Verfahren

Modul 2. Datenstruktur

- Implementieren linearer Strukturen wie Listen und Stapel
- Verwenden von Bäumen und Graphen in computergestützten Darstellungen
- Bewerten der Effizienz verschiedener Datenstrukturen
- Auswählen geeigneter Strukturen entsprechend der gestellten Aufgabe

Modul 3. Algorithmen und Komplexität

- Analysieren der zeitlichen und räumlichen Effizienz von Algorithmen
- Klassifizieren von Algorithmen nach ihrer rechnerischen Komplexität
- Lösen von Problemen unter Verwendung grundlegender algorithmischer Paradigmen
- Vergleichen von Lösungsalternativen hinsichtlich ihrer Leistungsfähigkeit

Modul 4. Fortgeschrittener Algorithmusentwurf

- Anwenden dynamischer und *Greedy*-Programmiertechniken
- Entwerfen von Algorithmen für komplexe Optimierungsprobleme
- Verwenden fortgeschrittener Strukturen beim Erstellen von Algorithmen
- Bewerten der Leistung fortgeschrittener algorithmischer Lösungen

Modul 5. Fortgeschrittene Programmierung

- Entwickeln von Programmen unter Verwendung objektorientierter Programmierkonzepte
- Integrieren von Fehlerbehandlung und *Debugging* in komplexe Projekte
- Implementieren von Entwurfsmustern in skalierbaren Anwendungen
- Verwenden von Bibliotheken und *Frameworks* zur Beschleunigung der Entwicklung

Modul 6. Theoretische Informatik

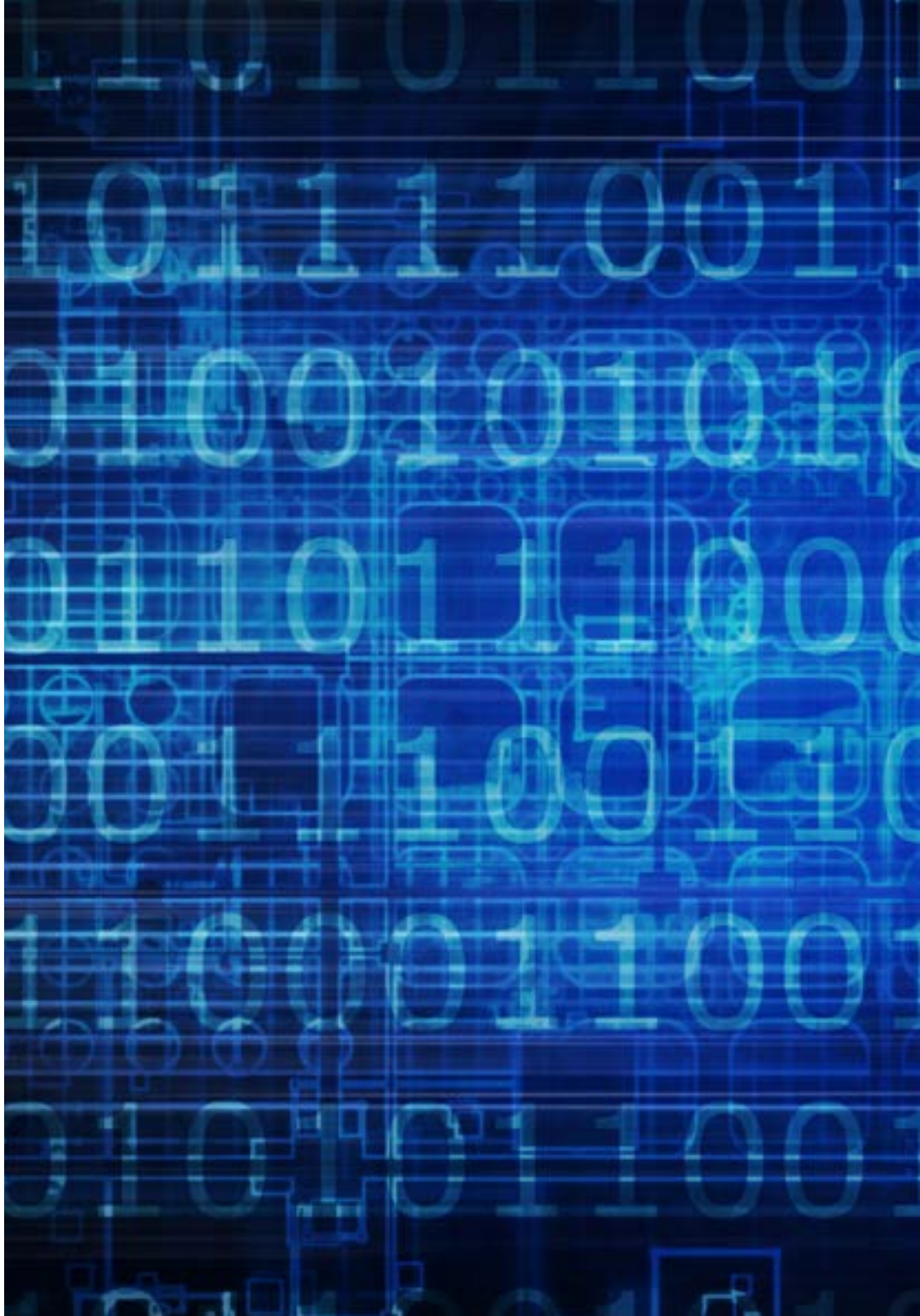
- Verstehen der Grenzen der Informatik anhand theoretischer Modelle
- Untersuchen nicht berechenbarer Probleme und ihrer Klassifizierung
- Analysieren der Klassen P und NP in der Informatiktheorie
- Anwenden theoretischer Konzepte bei der Lösung abstrakter Probleme

Modul 7. Automatentheorie und formale Sprachen

- Entwerfen von deterministischen und nicht deterministischen endlichen Automaten
- Generieren von regulären und kontextfreien Grammatiken
- Analysieren von formalen Sprachen mittels Automaten und regulären Ausdrücken
- Beziehen von Sprachtypen auf formale Hierarchien

Modul 8. Sprachprozessoren

- Implementieren grundlegender lexikalischer und syntaktischer Analysatoren
- Verstehen des Übersetzungsvorgangs in Compilern
- Entwerfen von Grammatiken für bestimmte Programmiersprachen
- Integrieren von Compiler-Phasen in eine funktionale Umgebung



Modul 9. Computergrafik und Visualisierung

- Verwenden von Grafikbibliotheken zur visuellen Darstellung von Daten
- Verstehen der Grundlagen der computergestützten Grafikmodellierung
- Implementieren von Rendering-Techniken in zwei und drei Dimensionen
- Entwerfen interaktiver Systeme zur Informationsvisualisierung

Modul 10. Bio-inspiriertes Computing

- Anwenden evolutionärer Algorithmen zur Lösung komplexer Probleme
- Verstehen der Grundlagen neuronaler Netze und ihres Trainings
- Einsetzen bio-inspirierter Strategien im Kontext der künstlichen Intelligenz
- Vergleichen bio-inspirierter Methoden mit traditionellen Algorithmen

Entwickeln Sie sich in aufstrebenden Bereichen wie Quantencomputing, PLN und fortschrittlicher Cybersicherheit weiter. Machen Sie sich unentbehrlich!“

05

Karrieremöglichkeiten

Die stetigen Fortschritte in den Bereichen künstliche Intelligenz, formale Sprachen und *Software-Engineering* haben die Beschäftigungsmöglichkeiten im Bereich der Informatik erheblich erweitert. Eine fundierte Spezialisierung in diesen Bereichen ermöglicht daher den Zugang zu stark nachgefragten Berufsbildern wie Sprachdesigner, Compiler-Entwickler, Forscher in der Informatiktheorie oder Experte für algorithmische Analyse. Darüber hinaus ist diese Art der Fortbildung von entscheidender Bedeutung für den Einstieg in strategische Bereiche wie Cybersicherheit, natürliche Sprachverarbeitung oder Quantencomputing, in denen fundierte Kenntnisse der Grundlagen der Informatik zunehmend geschätzt werden.

Sie werden Computerprozesse unter Verwendung modernster formaler und mathematischer Systeme modellieren“

Profil des Absolventen

Dieses Programm fördert die Entwicklung eines technischen und analytischen Profils, das in der Lage ist, die Prinzipien der modernen Computerlogik zu verstehen und anzuwenden. Dank einer integrativen Sichtweise zwischen Theorie und Praxis wird sich der Absolvent durch seine Fähigkeit auszeichnen, effiziente Lösungen zu entwerfen, komplexe Systeme zu modellieren und innovative Werkzeuge in verschiedenen technologischen Umgebungen zu entwickeln. Darüber hinaus macht ihn seine Beherrschung formaler Sprachen, fortgeschrittener Algorithmen und Computerstrukturen zu einem hochkompetenten Fachmann in akademischen, wissenschaftlichen oder industriellen Umgebungen, in denen ein tiefgreifendes Verständnis des internen Verhaltens digitaler Systeme erforderlich ist.

Sie werden bio-inspirierte Computertechniken in Innovationsprojekte einbringen, die natürliche und evolutionäre Prozesse nachahmen.

- **Kritisches Denken und Lösung komplexer Probleme:** Fähigkeit, Situationen aus einer logischen Perspektive zu analysieren und fundierte Lösungen vorzuschlagen
- **Effektive technische Kommunikation:** Fähigkeit, computertechnische Konzepte klar und präzise auszudrücken, sowohl mündlich als auch schriftlich
- **Zusammenarbeit in multidisziplinären Umgebungen:** Bereitschaft, sich zu integrieren und in Teams mit unterschiedlichen Profilen aus dem Technologiebereich mitzuwirken
- **Anpassungsfähigkeit an Veränderungen und kontinuierliches Lernen:** Bereitschaft, das Wissen in einem von ständiger Innovation geprägten Sektor auf dem neuesten Stand zu halten

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

1. **Softwareingenieur mit Spezialisierung auf Compiler:** Entwirft und implementiert Übersetzer für Programmiersprachen und optimiert deren Leistung und Funktionalität.
2. **Architekt von Programmiersprachen:** Entwickelt neue Sprachen oder passt bestehende Sprachen an die spezifischen Anforderungen von Technologiebranchen an.
3. **Forscher im Bereich der theoretischen Informatik:** Erforscht Computermodelle, algorithmische Komplexität und formale Sprachen, um abstrakte Probleme zu lösen, die auf reale Systeme anwendbar sind.
4. **Entwickler intelligenter Systeme:** Erstellt Lösungen auf der Grundlage von Computerlogik und natürlicher Sprachverarbeitung für fortgeschrittene Anwendungen.
5. **Algorithmus-Analyst:** Bewertet und optimiert Algorithmen, um ihre Effizienz in Kontexten mit hoher Rechenleistung zu verbessern.
6. **Spezialist für natürliche Sprachverarbeitung:** Entwirft Systeme, die in der Lage sind, menschliche Sprache in digitalen Umgebungen zu interpretieren, zu generieren oder zu übersetzen.
7. **Ingenieur für Computergrafik:** Entwickelt interaktive visuelle Lösungen, Simulationen und hochwertige Grafikumgebungen für verschiedene Branchen.
8. **Berater für Computerstrukturen:** Berät bei der Auswahl und Implementierung von Modellen und Strukturen für das Design effizienter Software.

> *Sie werden interaktive Grafiklösungen und nützliche Visualisierungen in der Softwareentwicklung implementieren“*

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 Informatik und Programmiersprachen 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.

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 ein und derselben 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.

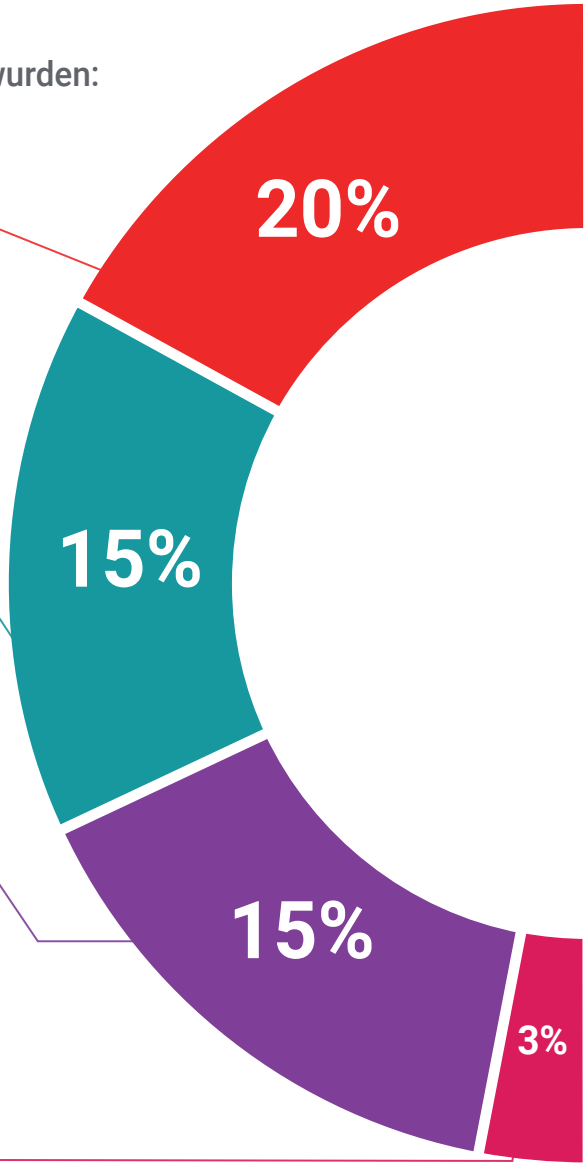
Interaktive Zusammenfassungen

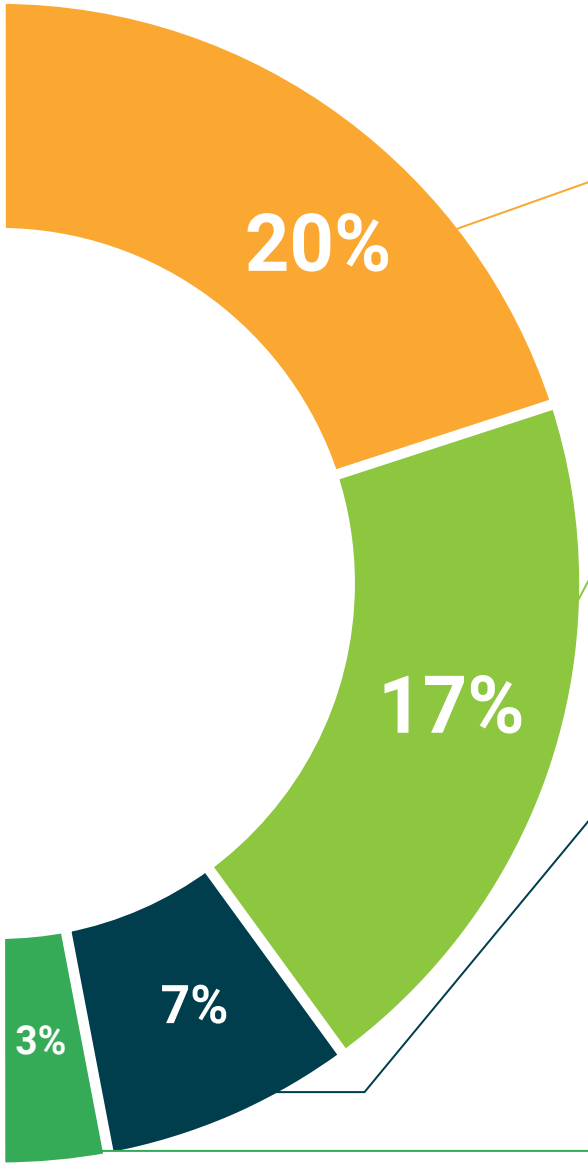
Wir präsentieren die Inhalte auf attraktive und dynamische Weise in multimedialen Pillen, Audios, Videos, Bilder, Diagramme und konzeptionelle Karten enthalten, um das Wissen zu festigen.

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.





Case Studies

Sie werden eine Auswahl der besten *case studies* zu diesem Thema bearbeiten. Die Fälle werden von den besten Spezialisten der internationalen Szene präsentiert, analysiert und betreut.

Testing & Retesting

Während des gesamten Programms werden Ihre Kenntnisse in regelmäßigen Abständen getestet und wiederholt. Wir tun dies auf 3 der 4 Ebenen der Millerschen Pyramide.

Meisterklassen

Die Nützlichkeit der Expertenbeobachtung ist wissenschaftlich belegt.

Das sogenannte *Learning from an Expert* stärkt das Wissen und das Gedächtnis und schafft Vertrauen in unsere zukünftigen schwierigen Entscheidungen.

Kurzanleitungen zum Vorgehen

TECH bietet die wichtigsten Inhalte des Kurses in Form von Arbeitsblättern oder Kurzanleitungen an. Ein synthetischer, praktischer und effektiver Weg, um dem Studenten zu helfen, in seinem Lernen voranzukommen.

08

Lehrkörper

Die Lehrkräfte dieses Programms sind Experten mit langjähriger akademischer und beruflicher Erfahrung in Schlüsselbereichen der Informatik und der formalen Sprachen. Dank ihrer Erfahrung in der angewandten Forschung und in der Entwicklung fortschrittlicher technologischer Lösungen bieten sie einen aktuellen und fundierten Einblick in die Branche. Darüber hinaus ermöglicht ihre Beteiligung an internationalen Projekten die Übertragung realer Fälle und innovativer Methoden in den Unterricht, was den Lernprozess erheblich bereichert. Diese Kombination aus fundiertem Wissen und praktischer Ausrichtung garantiert eine akademische Erfahrung auf hohem Niveau, die den Anforderungen des aktuellen wissenschaftlichen und technologischen Umfelds entspricht.

Sie werden jederzeit vom Dozententeam beraten, das sich aus renommierten Spezialisten für Informatik und Programmiersprachen zusammensetzt“

Internationaler Gastdirektor

Dr. Jeremy Gibbons gilt als **internationale Eminenz** für seine Beiträge im Bereich der **Programmiermethodik** und ihrer Anwendungen im **Software Engineering**. Seit mehr als zwei Jahrzehnten treibt dieser mit dem Fachbereich Informatik der Universität von Oxford verbundene Experte **verschiedene Entwicklungsprojekte** voran, deren greifbarste Ergebnisse von Informatikern in verschiedenen Teilen der Welt angewendet werden.

Seine Arbeit umfasst Bereiche wie **generische Programmierung**, formale Methoden, computergestützte Biologie, Bioinformatik und Algorithmenentwurf mit Haskell. Letzteres wurde in Zusammenarbeit mit seinem Mentor, Dr. Richard Bird, umfassend entwickelt.

In seiner Funktion als **Direktor** der **Programmieralgebra-Forschungsgruppe** hat Gibbons die Fortschritte bei den **funktionalen Programmiersprachen** und der **Mustertheorie in der Programmierung** vorangetrieben. Gleichzeitig wurden Anwendungen seiner Innovationen mit dem Gesundheitswesen in Verbindung gebracht, wie seine Zusammenarbeit mit **CancerGrid** und **Datatype Generic Programming** beweist. Diese und andere Initiativen wiederum spiegeln sein Interesse an der Lösung praktischer Probleme in der **Krebsforschung** und der **klinischen Informatik** wider.

Gibbons hat sich auch als **Chefredakteur** von **wissenschaftlichen Veröffentlichungen** in The Journal of Functional Programming und The Programming Journal: The Art, Science, and Engineering of Programming einen Namen gemacht. Im Rahmen dieser Aufgaben hat er sich intensiv um die **Verbreitung von Wissen** gekümmert. Darüber hinaus hatte er mehrere Lehrstühle inne, die mit renommierten Einrichtungen wie der Universität Oxford Brookes und der Universität von Auckland, Neuseeland, verbunden sind.

Außerdem ist dieser Spezialist Mitglied der Arbeitsgruppe 2.1 über Algorithmische Sprachen und Berechnungen der **International Federation for Information Processing (IFIP)**. Bei dieser Organisation ist er für die Wartung der Programmiersprachen ALGOL 60 und ALGOL 68 zuständig.

Dr. Gibbons, Jeremy

- Direktor des Software-Engineering-Programms an der Universität von Oxford, UK
- Stellvertretender Leiter des Informatiklabors und der Fakultät für Informatik an der Universität von Oxford
- Professor am Kellogg College, an der Universität Oxford Brookes und an der Universität von Auckland in Neuseeland
- Direktor der Forschungsgruppe Programmieralgebra
- Redakteur und Chef der Zeitschriften The Art, Science, and Engineering of Programming und Journal of Functional Programming
- Promotion in Computerwissenschaften an der Universität von Oxford
- Masterstudiengang in Informatik an der Universität von Edinburgh
- Mitglied von: Arbeitsgruppe 2.1 über Algorithmische Sprachen und Berechnungen der Internationalen Föderation für Informationsverarbeitung (IFIP)

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

09

Qualifizierung

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

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 Informatik und Programmiersprachen**.

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 Ausbildungsrahmen 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.

TECH ist Mitglied der **Association for Computing Machinery (ACM)**, dem internationalen Netzwerk, das die wichtigsten Akteure im Bereich der Informatik und Informationswissenschaften zusammenbringt. Diese Auszeichnung unterstreicht ihr Engagement für akademische Exzellenz, technologische Innovation und die Fortbildung von Fachkräften im digitalen Bereich.

Akkredditierung/Mitgliedschaft

Titel: **Weiterbildender Masterstudiengang in Informatik und Programmiersprachen**

Modalität: **online**

Dauer: **12 Monate**

Akkreditierung: **60 ECTS**

tech global university

Hr./Fr. _____, mit der Ausweis-Nr. _____ hat erfolgreich bestanden und den folgenden Abschluss erworben:

Weiterbildender Masterstudiengang in Informatik und Programmiersprachen

Es handelt sich um einen eigenen Abschluss mit einer Dauer von 1.800 Stunden, was 60 ECTS entspricht, mit Anfangsdatum am dd/mm/aaaa und Enddatum am dd/mm/aaaa.

TECH Global University ist eine von der Regierung Andorras am 31. Januar 2024 offiziell anerkannte Universität, die dem Europäischen Hochschulraum (EHR) angehört.

Andorra la Vella, den 28. Februar 2024

Dr. Pedro Navarro Illana
Rektor

einzigartiger Code TECH: AFWOR23S · techtitute.com/titol

Weiterbildender Masterstudiengang in Informatik und Programmiersprachen

Allgemeiner Aufbau des Lehrplans

Fachkategorie	ECTS
Obligatorisch (OB)	60
Wahlfach(OP)	0
Externes Praktikum (PR)	0
Masterarbeit (TFM)	0
	Summe 60

Allgemeiner Aufbau des Lehrplans

Kurs	Modul	ECTS	Kategorie
1°	Grundlagen der Programmierung	6	OB
1°	Datenstruktur	6	OB
1°	Algorithmen und Komplexität	6	OB
1°	Fortgeschrittener Algorithmusentwurf	6	OB
1°	Fortgeschrittene Programmierung	6	OB
1°	Theoretische Informatik	6	OB
1°	Automatentheorie und formale Sprachen	6	OB
1°	Sprachprozessoren	6	OB
1°	Computergrafik und Visualisierung	6	OB
1°	Bio-inspiriertes Computing	6	OB

Dr. Pedro Navarro Illana
Rektor

tech global university

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

tech global university

Weiterbildender Masterstudiengang

Informatik und Programmiersprachen

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

Weiterbildender Masterstudiengang
Informatik und Programmiersprachen

Akkredditierung/Mitgliedschaft

Association
for Computing
Machinery

tech global university