

Privater Masterstudiengang Künstliche Intelligenz und Wissensmodellierung



Privater Masterstudiengang Künstliche Intelligenz und Wissensmodellierung

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

Internetzugang: www.techtitude.com/de/informatik/masterstudiengang/masterstudiengang-kunstliche-intelligenz-wissensmodellierung

Index

01

Präsentation

Seite 4

02

Ziele

Seite 8

03

Kompetenzen

Seite 14

04

Struktur und Inhalt

Seite 18

05

Methodik

Seite 30

06

Qualifizierung

Seite 38

01

Präsentation

Das Programm wurde entwickelt, um Fachleute aus dem Ingenieurwesen in die spannende Welt der Künstlichen Intelligenz und der Wissensmodellierung einzuführen. Durch eine hochkompetente Weiterbildung werden sie in der Lage sein, einen soliden und solventen Schritt in diesem Bereich zu machen und die persönlichen und beruflichen Fähigkeiten zu erlangen, die notwendig sind, um als Experte auf diesem Gebiet zu praktizieren. Ein komplettes und effektives Programm, das sie auf die höchste Stufe der Kompetenz bringen wird.



“

Werden Sie eine der gefragtesten Fachkräfte der Gegenwart. Lassen Sie sich mit diesem sehr umfassenden Masterstudiengang in Künstlicher Intelligenz und Wissensmodellierung weiterbilden"

Entwicklungen, die auf künstlicher Intelligenz basieren, haben zahlreiche Anwendungen im Bereich der Technik erreicht. Von der Automatisierung zahlreicher Abläufe in Industrie und Unternehmen bis hin zur Prozesssteuerung selbst. Das bedeutet, dass Ingenieure diese komplexen Techniken kennen und beherrschen müssen.

Dieses grundlegende Wissen ist auch der erste Schritt, um Zugang zur Entwicklungskapazität dieser Art von Technologie zu erhalten.

Während dieser Fortbildung wird ein reales Arbeitsszenario angeboten, um die Eignung der Anwendung im eigenen Projekt beurteilen zu können. Dabei werden die tatsächlichen Indikationen, die Art und Weise der Entwicklung und die Erwartungen an die Ergebnisse bewertet.

Durch Erfahrung wird erlernt, wie man sich das notwendige Wissen aneignet, um in diesem Arbeitsbereich voranzukommen. Dieses Erlernen, das notwendigerweise Erfahrung voraussetzt, wird durch Fernunterricht und praktischen Unterricht miteinander in Einklang gebracht und bietet eine einzigartige Möglichkeit, Ihrem Lebenslauf den gewünschten Auftrieb zu geben.

Dieser **Privater Masterstudiengang in Künstliche Intelligenz und Wissensmodellierung** 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
- ◆ Die Inhalte sind von jedem festen oder tragbaren Gerät mit Internetanschluss abrufbar
- ◆ Datenbanken mit ergänzenden Unterlagen, die auch nach dem Kurs ständig verfügbar sind



Schließen Sie sich mit dieser hocheffektiven Fortbildung der Elite an und erschließen Sie sich neue Wege für Ihr berufliches Fortkommen"

“

Ein privater Masterstudiengang, der es Ihnen ermöglichen wird, in allen Bereichen der Künstlichen Intelligenz und der Wissensmodellierung mit der Sicherheit einer hochqualifizierten Fachkraft zu arbeiten”

Unser Lehrkörper setzt sich aus Fachleuten aus verschiedenen Bereichen zusammen, die mit diesem Fachgebiet in Verbindung stehen. Auf diese Weise stellen wir sicher, dass das angestrebte Ziel der didaktischen Aktualisierung erreicht wird. Ein multidisziplinäres Team von Fachleuten, die in verschiedenen Umgebungen ausgebildet und erfahren sind, wird Ihnen die theoretischen Kenntnisse effizient vermitteln, 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 Gestaltung ergänzt. Die Methode 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 Multimedia-Tools lernen, die Ihnen die nötige Handlungsfähigkeit für Ihre Fortbildung bieten.

Das Design dieses Programms basiert auf problemorientiertem Lernen: ein Ansatz, der Lernen als einen eminent praktischen Prozess begreift. Um dies aus der Ferne zu erreichen, wird die Telepraxis eingesetzt. Mit Hilfe eines innovativen interaktiven Videosystems und dem Learning From an Expert können Sie sich Wissen aneignen, als ob Sie sich in der Situation befänden, die Sie gerade lernen. Ein Konzept, das es ermöglichen wird, das Lernen auf eine realistischere und dauerhaftere Weise zu integrieren und zu festigen.

Mit einem methodischen Konzept, das auf bewährten Lehrtechniken basiert, führt Sie dieser innovative Private Masterstudiengang in Künstliche Intelligenz und Wissensmodellierung durch verschiedene Lehransätze, um Ihnen ein dynamisches und effektives Lernen zu ermöglichen.

Unser innovatives 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”



02 Ziele

Das Ziel ist es, hochqualifizierte Fachkräfte für die Berufspraxis zu spezialisieren
Ein Ziel, das im Übrigen auf globaler Ebene durch die Förderung einer menschlichen
Entwicklung ergänzt wird, die den Grundstein für eine bessere Gesellschaft legt. Dieses
Ziel wird erreicht, indem Fachleuten geholfen wird, ein viel höheres Maß an Kompetenz
und Kontrolle zu erlangen. Ein Vorhaben, welches sie in wenigen Monaten mit einer
Fortbildung von hoher Intensität und Präzision erreichen können.



“

Wenn es Ihr Ziel ist, Ihre Fähigkeiten auf neue Wege des Erfolgs und der Entwicklung auszurichten, dann ist dieses Programm das Richtige für Sie: eine Fortbildung, die auf Spitzenleistungen abzielt"

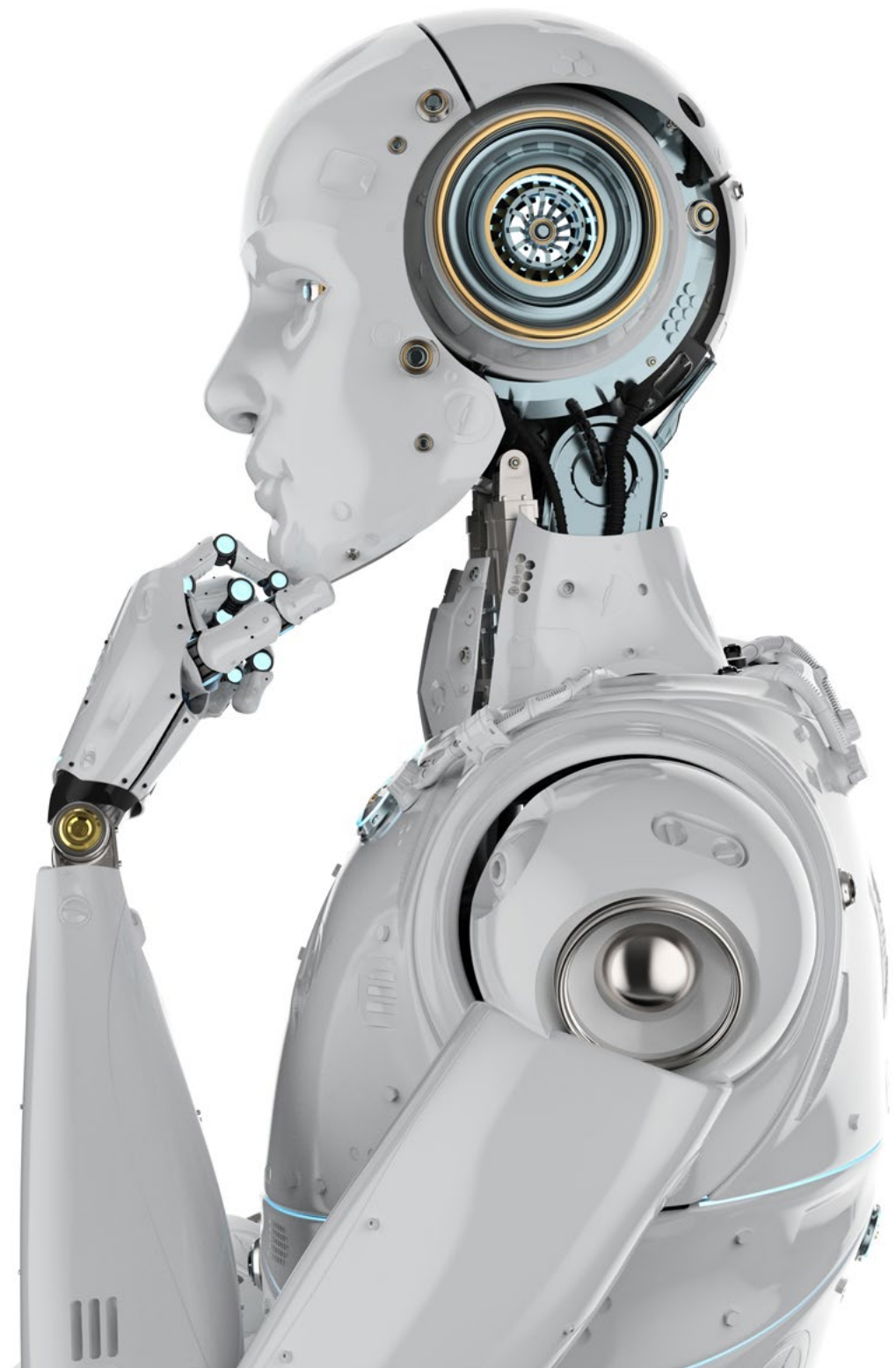


Allgemeine Ziele

- ◆ Fortbilden 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



Lassen Sie sich die Gelegenheit nicht entgehen und bringen Sie sich auf den neuesten Stand im Bereich Künstliche Intelligenz und Wissensmodellierung"





Spezifische Ziele

Modul 1. Grundlagen der Programmierung

- ◆ Verstehen der grundlegenden Struktur eines Computers, von Software und allgemeinen Programmiersprachen
- ◆ Lernen, wie man Algorithmen entwirft und interpretiert, die die notwendige Grundlage für die Softwareentwicklung sind
- ◆ Verstehen der wesentlichen Elemente eines Computerprogramms, wie z. B. Datentypen, Operatoren, Ausdrücke, Anweisungen, E/A und Steueranweisungen
- ◆ Verstehen der verschiedenen Datenstrukturen, die in allgemeinen Programmiersprachen zur Verfügung stehen, sowohl statisch als auch dynamisch, und Erwerben der wesentlichen Kenntnisse für den Umgang mit Dateien
- ◆ Kennen der verschiedenen Softwaretesttechniken und der Bedeutung der Erstellung einer guten Dokumentation zusammen mit einem guten Quellcode
- ◆ Erlernen der grundlegenden Konzepte der Programmiersprache C++, einer der am häufigsten verwendeten Sprachen der Welt

Modul 2. Datenstruktur

- ◆ Lernen der Grundlagen der Programmierung in der Sprache C++, einschließlich Klassen, Variablen, bedingte Ausdrücke und Objekte
- ◆ Verstehen von abstrakten Datentypen, linearen Datenstrukturtypen, einfachen und komplexen hierarchische Datenstrukturen und deren Implementierung in C++
- ◆ Verstehen der Funktionsweise von fortgeschrittenen Datenstrukturen, die nicht den üblichen entsprechen
- ◆ Verstehen der Theorie und Praxis im Zusammenhang mit der Verwendung von Prioritätsheaps und Prioritätswarteschlangen
- ◆ Lernen der Funktionsweise von Hash-Tabellen als abstrakte Datentypen und Funktionen
- ◆ Verstehen der Graphentheorie sowie fortgeschrittene Graph-Algorithmen und Konzepte

Modul 3. Algorithmen und Komplexität

- ◆ Erlernen der wichtigsten Strategien für den Entwurf von Algorithmen sowie der verschiedenen Methoden und Maße für die Berechnung von Algorithmen
- ◆ Kennen der wichtigsten Sortieralgorithmen, die in der Softwareentwicklung verwendet werden
- ◆ Verstehen, wie verschiedene Algorithmen mit Bäumen, *Heaps* und Graphen arbeiten
- ◆ Verstehen der Funktionsweise von *Greedy*-Algorithmen, ihrer Strategie und Beispiele für ihre Anwendung bei den wichtigsten bekannten Problemen. Kennen der Verwendung von *Greedy*-Algorithmen auf Graphen
- ◆ Kennen der wichtigsten Strategien der Suche nach minimalen Pfaden, mit der Annäherung an wesentliche Probleme des Feldes und Algorithmen zu deren Lösung
- ◆ Verstehen der Backtracking-Technik und ihrer wichtigsten Anwendungen sowie anderer alternativer Techniken

Modul 4. Fortgeschrittener Algorithmenentwurf

- ◆ Vertiefen in fortgeschrittenes Algorithmen-Design, Analysieren von rekursiven und Divide-and-Conquer-Algorithmen sowie Durchführen von amortisierten Analysen
- ◆ Verstehen der Konzepte der dynamischen Programmierung und Algorithmen für NP-Probleme
- ◆ Verstehen der Funktionsweise der kombinatorischen Optimierung, sowie der verschiedenen Randomisierungsalgorithmen und parallelen Algorithmen
- ◆ Kennen und Verstehen der Funktionsweise der verschiedenen lokalen und Kandidaten-Suchmethoden
- ◆ Erlernen der Mechanismen der formalen Verifikation von Programmen und iterativen Programmen, einschließlich der Logik erster Ordnung und des formalen Systems von *Hoare*
- ◆ Lernen der Funktionsweise einiger der wichtigsten numerischen Methoden wie die Bisektionsmethode, die Newton-Raphson-Methode und die Sekantenmethode

Modul 5. Computerlogik

- ◆ Lernen der Grundlagen der Berechnungslogik, wofür sie verwendet wird und die Gründe für ihre Verwendung
- ◆ Kennen der verschiedenen Strategien der Formalisierung und Deduktion in der Aussagenlogik, einschließlich des natürlichen Schlussfolgerns, der axiomatischen Deduktion und der natürlichen Deduktion, sowie der primitiven Regeln der Aussagenlogik
- ◆ Erwerben von fortgeschrittenen Kenntnissen in der Aussagenlogik, die sich mit ihrer Semantik und den wichtigsten Anwendungen dieser Logik wie z. B. logischen Schaltkreisen, befassen
- ◆ Verstehen der Prädikatenlogik sowohl für den Kalkül der natürlichen Deduktion von Prädikaten als auch für Formalisierungs- und Deduktionsstrategien für die Prädikatenlogik
- ◆ Verstehen der Grundlagen der natürlichen Sprache und ihres deduktiven Mechanismus
- ◆ Einführen in die logische Programmierung mit der Sprache PROLOG

Modul 6. 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
- ◆ Erlernen der grundlegenden Konzepte von neuronalen Netzwerken 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 7. Intelligente Systeme

- ◆ Erlernen aller Konzepte im Zusammenhang mit der Agententheorie und der Agentenarchitektur sowie deren Argumentationsprozess
- ◆ Verstehen der Theorie und Praxis, die hinter den Konzepten von Information und Wissen stehen, sowie der verschiedenen Arten der Darstellung von Wissen
- ◆ Verstehen der Theorie der Ontologien, sowie Lernen von Sprachen für Ontologien und Software für die Erstellung von Ontologien
- ◆ Erlernen unterschiedlicher Modelle der Wissensdarstellung, wie Vokabulare, Taxonomien, Thesauri, Mind Maps und andere
- ◆ Verstehen, wie semantische Reasoner, wissensbasierte Systeme und Expertensysteme funktionieren
- ◆ Wissen, wie das semantische Web funktioniert, seinen aktuellen und zukünftigen Stand, sowie Anwendungen, die auf dem semantischen Web basieren

Modul 8. Maschinelles Lernen und Data Mining

- ◆ Einführen in die Prozesse der Wissensentdeckung und grundlegende Konzepte des maschinellen Lernens
- ◆ Lernen der Methoden zur Datenexploration und -vorverarbeitung sowie verschiedener auf Entscheidungsbäumen basierender Algorithmen
- ◆ Verstehen der Funktionsweise der Bayesschen Methoden sowie der Regressions- und kontinuierlichen Antwortmethoden
- ◆ Verstehen der verschiedenen Klassifizierungsregeln und der Bewertung von Klassifikatoren, durch Erlernen der Verwendung von Konfusionsmatrizen und numerischer Bewertung, der Kappa-Statistik und der ROC-Kurve
- ◆ Erwerben grundlegender Kenntnisse in den Bereichen *Text Mining*, Verarbeitung natürlicher Sprache (NLP) und *Clustering*
- ◆ Vertiefen der Kenntnisse über neuronale Netze, von einfachen neuronalen Netzen bis hin zu rekurrenten neuronalen Netzen



Modul 9. Multi-Agenten-Systeme und rechnergestützte Wahrnehmung

- ◆ Verstehen der grundlegenden und fortgeschrittenen Konzepte im Zusammenhang mit Agenten und Multiagentensystemen
- ◆ Untersuchen des FIPA-Standards für Agenten unter Berücksichtigung der Kommunikation zwischen Agenten, der Agentenverwaltung und der Architektur sowie anderer Aspekte
- ◆ Vertiefen der Kenntnisse über die JADE-Plattform (*Java Agent Development Framework*) und Lernen, sowohl grundlegende als auch fortgeschrittene Konzepte darin zu programmieren, einschließlich Fragen der Kommunikation und der Agentenerkennung
- ◆ Vermitteln der Grundlagen der Verarbeitung natürlicher Sprache, wie z. B. der automatischen Spracherkennung und der Computerlinguistik
- ◆ Verstehen der Funktionsweise von Computer Vision, digitaler Bildanalyse, Bildtransformation und -segmentierung in allen Einzelheiten

Modul 10. Bio-inspiriertes Computing

- ◆ Einführen in das Konzept des bio-inspirierten Computings sowie Verständnis für die Funktionsweise der verschiedenen Arten von sozialen Anpassungsalgorithmen und genetischen Algorithmen
- ◆ Vertiefen des Studiums der verschiedenen Modelle der evolutionären Berechnung, Kennen ihrer Strategien, Programmierung, Algorithmen und Modelle, die auf der Schätzung von Verteilungen basieren
- ◆ Verstehen der wichtigsten Strategien zur Erkundung und Ausnutzung des Raums für genetische Algorithmen
- ◆ Verstehen der Funktionsweise der evolutionären Programmierung bei Lernproblemen und Mehrzielproblemen
- ◆ Lernen der wesentlichen Konzepte im Zusammenhang mit neuronalen Netzwerken und Verstehen, wie sie in realen Anwendungsfällen in so unterschiedlichen Bereichen wie medizinische Forschung, Wirtschaft und Computer Vision funktionieren

03

Kompetenzen

Dieser Private Masterstudiengang in Künstlicher Intelligenz und Wissensmodellierung wurde als hochqualifiziertes Fortbildungsinstrument für Fachleute entwickelt. Die intensive Weiterbildung wird Sie darauf vorbereiten, in allen Bereichen der Künstlichen Intelligenz mit der Sicherheit eines Experten auf diesem Gebiet zu arbeiten.



“

Der Private Masterstudiengang in Künstlicher Intelligenz und Wissensmodellierung wird Ihnen die wesentlichen persönlichen und beruflichen Fähigkeiten vermitteln, um in jeder beruflichen Situation in diesem Bereich eine angemessene Rolle zu spielen”

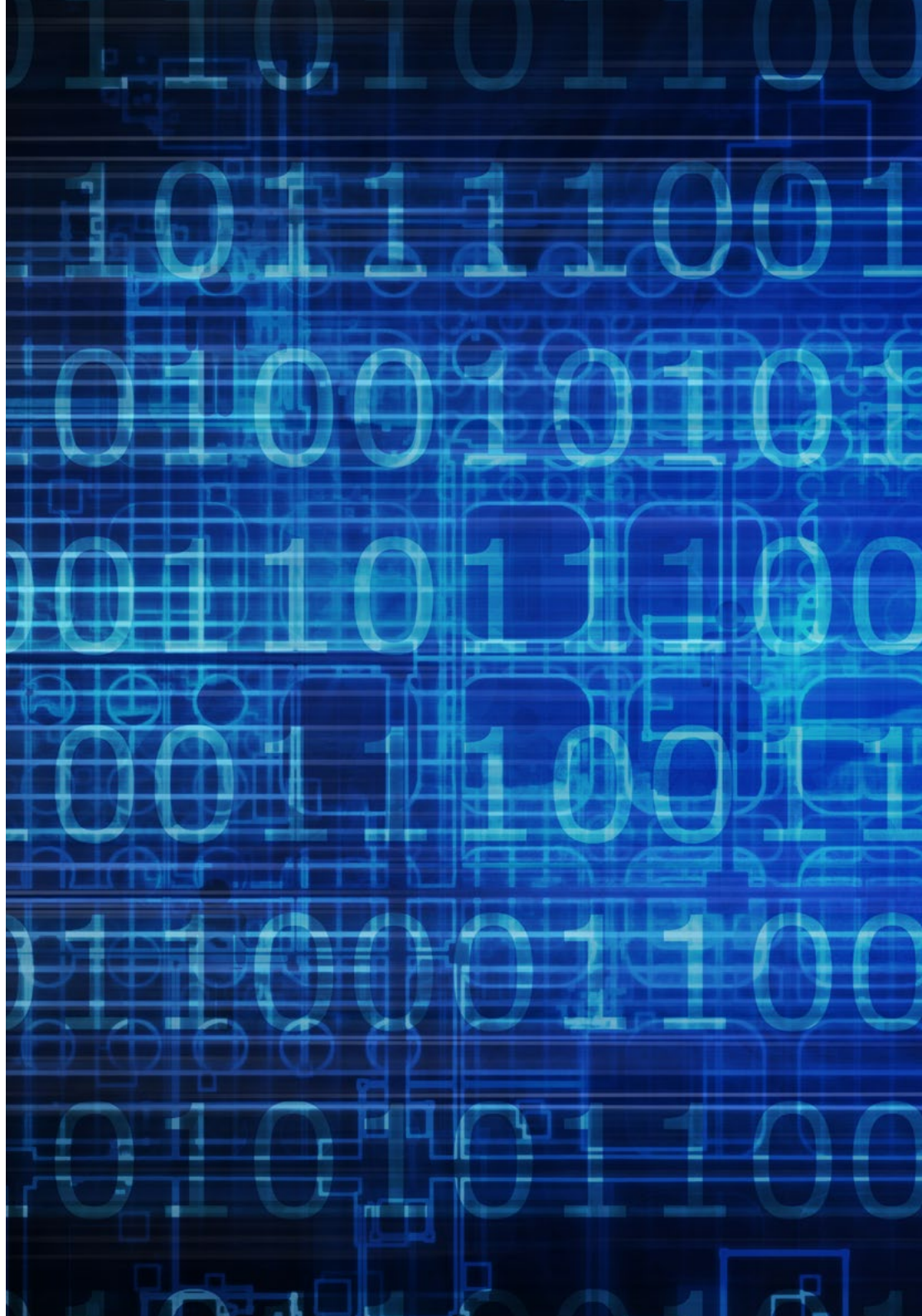


Allgemeine Kompetenz

- ♦ 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 einzigartige, wichtige
und entscheidende
Fortbildungserfahrung, die Ihre
berufliche Entwicklung fördert"*





Spezifische Kompetenzen

- ◆ Entwickeln der Programmierung auf dem Gebiet der künstlichen Intelligenz unter Berücksichtigung aller Faktoren der Entwicklung künstlicher Intelligenz
- ◆ Beherrschen der Datenstrukturen der C++-Programmierung
- ◆ Erstellen grundlegender und fortgeschrittener Algorithmen
- ◆ Verstehen der Berechnungslogik und deren Anwendung bei der Gestaltung von Projekten
- ◆ Erwerben von Kenntnissen über künstliche Intelligenz, ihre Anwendungen und Entwicklungen sowie Erstellen eigener Projekte
- ◆ Wissen, was sie sind, wie sie funktionieren und wie man mit intelligenten Systemen arbeitet
- ◆ Beherrschen der grundlegenden Konzepte des maschinellen Lernens
- ◆ Kennen von JADE, FIPA, Computer Vision und anderen Multiagentensystemen
- ◆ Verstehen von Algorithmen des bio-inspirierten Computings und Strategien für deren Einsatz

04

Struktur und Inhalt

Die Inhalte dieses privaten Masterstudiengangs wurden von verschiedenen Experten auf dem Gebiet mit einem klaren Ziel entwickelt: sicherzustellen, dass die Studenten alle notwendigen Fähigkeiten erwerben, um echte Experten in allen Bereichen der Künstlichen Intelligenz zu werden.

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



MA
LEA



“

Ein sehr komplettes Lehrprogramm, das in hervorragend ausgearbeitete didaktische Einheiten gegliedert ist, ausgerichtet auf ein Lernen, das mit dem persönlichen und beruflichen Leben kompatibel ist"

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. Algorithmusentwurf
 - 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 grundlegender Daten
 - 1.3.5. Betreiber
 - 1.3.6. Ausdrücke
 - 1.3.7. Sätze
 - 1.3.8. Dateneingabe und -ausgabe
- 1.4. Kontrollsätze
 - 1.4.1. Sätze
 - 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. Kennung des Geltungsbereichs
- 1.6. Statische Datenstrukturen
 - 1.6.1. Arrays
 - 1.6.2. Matrizen. Polyeder
 - 1.6.3. Suchen und Sortieren
 - 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. Multiple 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. Vorteile und Nachteile
 - 1.9.4. Überlegungen
 - 1.9.5. Rekursiv-iterative Umwandlung
 - 1.9.6. Der Rekursionsstapel

- 1.10. Prüfung und Dokumentation
 - 1.10.1. Programm-Tests
 - 1.10.2. White Box-Tests
 - 1.10.3. Black Box-Tests
 - 1.10.4. Test-Tools
 - 1.10.5. Programm-Dokumentation

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. C++ Stapel und Warteschlange
- 2.4. Hierarchische Datenstrukturen
 - 2.4.1. ADT-Baum
 - 2.4.2. Pfade
 - 2.4.3. n-äre Bäume
 - 2.4.4. Binäre Bäume
 - 2.4.5. Binäre Suchbäume
- 2.5. Hierarchische Datenstrukturen: Komplexe Bäume
 - 2.5.1. Perfekt ausbalancierte oder minimal hohe Bäume
 - 2.5.2. Bäume mit mehreren Pfaden
 - 2.5.3. Bibliografische Referenzen
- 2.6. Heaps und Prioritätswarteschlange
 - 2.6.1. ADT-Heaps
 - 2.6.2. ADT-Prioritätswarteschlange
- 2.7. Hash-Tabellen
 - 2.7.1. ADT in *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. Graph
 - 2.8.1. ADT-Graph
 - 2.8.2. Arten von Graphen
 - 2.8.3. Grafische Darstellung und Grundoperationen
 - 2.8.4. Graphen-Design
- 2.9. Graph-Algorithmen und fortgeschrittene Graph-Konzepte
 - 2.9.1. Graph-Probleme
 - 2.9.2. Pfad-Algorithmen
 - 2.9.3. Such- oder Pfad-Algorithmen
 - 2.9.4. Andere Algorithmen
- 2.10. Andere Datenstrukturen
 - 2.10.1. Sets
 - 2.10.2. Parallele Arrays
 - 2.10.3. Symboltabellen
 - 2.10.4. Tries

Modul 3. Algorithmen und Komplexität

- 3.1. Einführung in Algorithmus-Design-Strategien
 - 3.1.1. Rekursion
 - 3.1.2. Aufteilen und erobern
 - 3.1.3. Andere Strategien
- 3.2. Effizienz und Analyse von Algorithmen
 - 3.2.1. Maßnahmen zur Steigerung der Effizienz
 - 3.2.2. Messung der Eingabegröße
 - 3.2.3. Messung der Ausführungszeit
 - 3.2.4. Schlimmster, bester und durchschnittlicher Fall
 - 3.2.5. Asymptotische Notation
 - 3.2.6. Kriterien für die mathematische Analyse von nicht-rekursiven 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. Blase sortieren
 - 3.3.3. Sortieren nach Auswahl
 - 3.3.4. Reihenfolge der Insertion
 - 3.3.5. Sortierung zusammenführen (*Merge_Sort*)
 - 3.3.6. Schnelle Sortierung (*Quick_Sort*)
- 3.4. Algorithmen mit Bäumen
 - 3.4.1. Konzept des Baumes
 - 3.4.2. Binäre Bäume
 - 3.4.3. Baumpfade
 - 3.4.4. Ausdrücke darstellen
 - 3.4.5. Geordnete binäre Bäume
 - 3.4.6. Ausgeglichene binäre Bäume
- 3.5. Algorithmen mit *Heaps*
 - 3.5.1. *Heaps*
 - 3.5.2. Der *Heapsort*-Algorithmus
 - 3.5.3. Prioritätswarteschlangen



- 3.6. Graph-Algorithmen
 - 3.6.1. Vertretung
 - 3.6.2. Lauf in Breite
 - 3.6.3. Lauf in Tiefe
 - 3.6.4. Topologische Anordnung
 - 3.7. Greedy-Algorithmen
 - 3.7.1. Die Greedy-Strategie
 - 3.7.2. Elemente der Greedy-Strategie
 - 3.7.3. Währungsumtausch
 - 3.7.4. Das Problem des Reisenden
 - 3.7.5. Problem mit dem Rucksack
 - 3.8. Minimale Pfadsuche
 - 3.8.1. Das Problem des minimalen Pfades
 - 3.8.2. Negative Bögen und Zyklen
 - 3.8.3. Dijkstra-Algorithmus
 - 3.9. Greedy-Algorithmen auf Graphen
 - 3.9.1. Der minimal aufspannende Baum
 - 3.9.2. Algorithmus von Prim
 - 3.9.3. Algorithmus von Kruskal
 - 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 Teilen-und-Erobern-Algorithmen
 - 4.1.1. Aufstellen und Lösen von homogenen und nicht-homogenen Rekursionsgleichungen
 - 4.1.2. Überblick über die Strategie des Teilens und Eroberns
 - 4.2. Amortisierte Analyse
 - 4.2.1. Aggregierte Analyse
 - 4.2.2. Die Buchhaltungsmethode
 - 4.2.3. Die potenzielle Methode
 - 4.3. Dynamische Programmierung und Algorithmen für NP-Probleme
 - 4.3.1. Merkmale der dynamischen Programmierung
 - 4.3.2. Rückverfolgung: Backtracking
 - 4.3.3. Verzweigung und Beschneidung
 - 4.4. Kombinatorische Optimierung
 - 4.4.1. Problemdarstellung
 - 4.4.2. 1D-Optimierung
 - 4.5. Randomisierungsalgorithmen
 - 4.5.1. Beispiele für Randomisierungsalgorithmen
 - 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. Garcient *Ascent*
 - 4.6.2. Hill *Climbing*
 - 4.6.3. Simulated *Annealing*
 - 4.6.4. Tabu *Search*
 - 4.6.5. Suche mit Kandidaten
 - 4.7. Formale Überprüfung von Programmen
 - 4.7.1. Spezifikation von funktionalen Abstraktionen
 - 4.7.2. Die Sprache der Logik erster Ordnung
 - 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. Die 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. Computerlogik

- 5.1. Rechtfertigung der Logik
 - 5.1.1. Studienobjekt der Logik
 - 5.1.2. Wozu dient die Logik?
 - 5.1.3. Komponenten und Arten der Argumentation
 - 5.1.4. Komponenten einer logischen Berechnung
 - 5.1.5. Semantik
 - 5.1.6. Rechtfertigung für die Existenz einer Logik
 - 5.1.7. Wie kann man überprüfen, ob eine Logik angemessen ist?
- 5.2. Kalkül der natürlichen Deduktion von Aussagen
 - 5.2.1. Formale Sprache
 - 5.2.2. Deduktiver Mechanismus
- 5.3. Formalisierung und Deduktionsstrategien für Aussagenlogik
 - 5.3.1. Strategien zur Formalisierung
 - 5.3.2. Natürliche Argumentation
 - 5.3.3. Gesetze und Regeln
 - 5.3.4. Axiomatische Deduktion und natürliche Deduktion
 - 5.3.5. Das Kalkül der natürlichen Deduktion
 - 5.3.6. Primitive Regeln der Aussagenlogik
- 5.4. Semantik der Aussagenlogik
 - 5.4.1. Wahrheitstabellen
 - 5.4.2. Äquivalenz
 - 5.4.3. Tautologien und Widersprüche
 - 5.4.4. Validierung von Aussagesätzen
 - 5.4.5. Validierung mit Hilfe von Wahrheitstabellen
 - 5.4.6. Validierung mit semantischen Bäumen
 - 5.4.7. Validierung durch Widerlegung
- 5.5. Anwendungen der Aussagenlogik: Logische Schaltungen
 - 5.5.1. Grundlegende Gatter
 - 5.5.2. Schaltkreise
 - 5.5.3. Mathematische Modelle von Schaltkreisen
 - 5.5.4. Minimierung
 - 5.5.5. Zweite kanonische Form und minimale Form im Produkt der Summen
 - 5.5.6. Andere Gatter

- 5.6. Natürlicher Prädikatenschlusskalkül
 - 5.6.1. Formale Sprache
 - 5.6.2. Deduktiver Mechanismus
- 5.7. Formalisierungsstrategien für Prädikatenlogik
 - 5.7.1. Einführung in die Formalisierung der Prädikatenlogik
 - 5.7.2. Formalisierungsstrategien mit Quantoren
- 5.8. Deduktionsstrategien für Prädikatenlogik
 - 5.8.1. Grund für eine Auslassung
 - 5.8.2. Präsentation der neuen Regeln
 - 5.8.3. Prädikatenlogik als natürliches Deduktionskalkül
- 5.9. Anwendungen der Prädikatenlogik: Einführung in die logische Programmierung
 - 5.9.1. Informelle Präsentation
 - 5.9.2. Elemente von Prolog
 - 5.9.3. Neubewertung und Ausschluss
- 5.10. Mengenlehre, Prädikatenlogik und ihre Semantik
 - 5.10.1. Intuitionistische Mengenlehre
 - 5.10.2. Einführung in die Automatentheorie

Modul 6. Künstliche Intelligenz und Wissensmodellierung

- 6.1. Einführung in Künstliche Intelligenz und Wissensmodellierung
 - 6.1.1. Kurze Geschichte der Künstlichen Intelligenz
 - 6.1.2. Künstliche Intelligenz heute
 - 6.1.3. Wissensmodellierung
- 6.2. Suche
 - 6.2.1. Allgemeine Suchkonzepte
 - 6.2.2. Uninformierte Suche
 - 6.2.3. Informierte Suche
- 6.3. Boolesche Erfüllbarkeit, Erfüllbarkeit von Nebenbedingungen und automatische Planung
 - 6.3.1. Boolesche Erfüllbarkeit
 - 6.3.2. Probleme mit der Erfüllung von Einschränkungen
 - 6.3.3. Automatische Planung und PDDL
 - 6.3.4. Planung als heuristische Suche
 - 6.3.5. Planung mit SAT

- 6.4. Künstliche Intelligenz in Spielen
 - 6.4.1. Spieltheorie
 - 6.4.2. Minimax und Alpha-Beta-Beschneidung
 - 6.4.3. Simulation: Monte Carlo
- 6.5. Überwachtes und unüberwachtes Lernen
 - 6.5.1. Einführung in das maschinelle Lernen
 - 6.5.2. Klassifizierung
 - 6.5.3. Regression
 - 6.5.4. Validierung der Ergebnisse
 - 6.5.5. Gruppieren (*Clustering*)
- 6.6. Neuronale Netzwerke
 - 6.6.1. Biologische Grundlagen
 - 6.6.2. Berechnungsmodell
 - 6.6.3. Überwachte und nicht überwachte neuronale Netzwerke
 - 6.6.4. Einfaches Perzeptron
 - 6.6.5. Mehrschichtiges Perzeptron
- 6.7. Genetische Algorithmen
 - 6.7.1. Geschichte
 - 6.7.2. Biologische Grundlage
 - 6.7.3. Problem-Kodierung
 - 6.7.4. Erzeugung der Ausgangspopulation
 - 6.7.5. Hauptalgorithmus und genetische Operatoren
 - 6.7.6. Bewertung von Personen: *Fitness*
- 6.8. Thesauri, Vokabularien, Taxonomien
 - 6.8.1. Wortschatz
 - 6.8.2. Taxonomie
 - 6.8.3. Thesauri
 - 6.8.4. Ontologien
- 6.9. Wissensrepräsentation: Semantisches Web
 - 6.9.1. Semantisches Web
 - 6.9.2. Spezifizierungen: RDF, RDFS und OWL
 - 6.9.3. Schlussfolgerung/Begründung
 - 6.9.4. Verknüpfte Daten

- 6.10. Expertensysteme und DSS
 - 6.10.1. Experten-Systeme
 - 6.10.2. Systeme zur Entscheidungshilfe

Modul 7. Intelligente Systeme

- 7.1. Agententheorie
 - 7.1.1. Geschichte des Konzepts
 - 7.1.2. Definition von Agent
 - 7.1.3. Agenten in der künstlichen Intelligenz
 - 7.1.4. Agenten in der Softwareentwicklung
- 7.2. Agent-Architekturen
 - 7.2.1. Der Denkprozess eines Agenten
 - 7.2.2. Reaktive Wirkstoffe
 - 7.2.3. Deduktive Agenten
 - 7.2.4. Hybride Agenten
 - 7.2.5. Vergleich
- 7.3. Informationen und Wissen
 - 7.3.1. Unterscheidung zwischen Daten, Informationen und Wissen
 - 7.3.2. Bewertung der Datenqualität
 - 7.3.3. Methoden der Datenerfassung
 - 7.3.4. Methoden zur Datenerhebung
 - 7.3.5. Methoden zum Wissenserwerb
- 7.4. Darstellung von Wissen
 - 7.4.1. Die Bedeutung der Wissensdarstellung
 - 7.4.2. Definition der Wissensrepräsentation durch ihre Rollen
 - 7.4.3. Merkmale einer Wissensrepräsentation
- 7.5. Ontologien
 - 7.5.1. Einführung in Metadaten
 - 7.5.2. Philosophisches Konzept der Ontologie
 - 7.5.3. Computergestütztes Konzept der Ontologie
 - 7.5.4. Bereichsontologien und Ontologien auf höherer Ebene
 - 7.5.5. Wie erstellt man eine Ontologie?

- 7.6. Ontologiesprachen und Software für die Erstellung von Ontologien
 - 7.6.1. RDF-Tripel, Turtle und N3
 - 7.6.2. RDF-Schema
 - 7.6.3. OWL
 - 7.6.4. SPARQL
 - 7.6.5. Einführung in die verschiedenen Tools für die Erstellung von Ontologien
 - 7.6.6. Installation und Verwendung von Protégé
- 7.7. Das semantische Web
 - 7.7.1. Der aktuelle Stand und die Zukunft des semantischen Webs
 - 7.7.2. Anwendungen des Semantischen Webs
- 7.8. Andere Modelle der Wissensdarstellung
 - 7.8.1. Wortschatz
 - 7.8.2. Globale Sicht
 - 7.8.3. Taxonomie
 - 7.8.4. Thesauri
 - 7.8.5. Folksonomien
 - 7.8.6. Vergleich
 - 7.8.7. Mind Map
- 7.9. Bewertung und Integration von Wissensrepräsentationen
 - 7.9.1. Logik nullter Ordnung
 - 7.9.2. Logik erster Ordnung
 - 7.9.3. Beschreibende Logik
 - 7.9.4. Beziehung zwischen verschiedenen Arten von Logik
 - 7.9.5. Prolog: Programmierung auf Basis der Logik erster Ordnung
- 7.10. Semantische *Reasoner*, wissensbasierte Systeme und Expertensysteme
 - 7.10.1. Konzept des *Reasoners*
 - 7.10.2. Anwendungen eines *Reasoners*
 - 7.10.3. Wissensbasierte Systeme
 - 7.10.4. MYCIN, Geschichte der Expertensysteme
 - 7.10.5. Elemente und Architektur von Expertensystemen
 - 7.10.6. Erstellung von Expertensystemen

Modul 8. Maschinelles Lernen und *Data Mining*

- 8.1. Einführung in die Prozesse der Wissensentdeckung und in die grundlegenden Konzepte des maschinellen Lernens
 - 8.1.1. Schlüsselkonzepte von Prozessen der Wissensentdeckung
 - 8.1.2. Historische Perspektive der Wissensentdeckungsprozesse
 - 8.1.3. Phasen des Wissensentdeckungsprozesses
 - 8.1.4. Techniken, die bei der Wissensentdeckung eingesetzt werden
 - 8.1.5. Merkmale guter Modelle für maschinelles Lernen
 - 8.1.6. Arten von Informationen zum maschinellen Lernen
 - 8.1.7. Grundlegende Lernkonzepte
 - 8.1.8. Grundlegende Konzepte des unüberwachten Lernens
- 8.2. Datenexploration und Vorverarbeitung
 - 8.2.1. Datenverarbeitung
 - 8.2.2. Datenverarbeitung im Datenanalysefluss
 - 8.2.3. Datentypen
 - 8.2.4. Datenumwandlung
 - 8.2.5. Anzeige und Untersuchung von kontinuierlichen Variablen
 - 8.2.6. Anzeige und Erkundung kategorialer Variablen
 - 8.2.7. Korrelation Maßnahmen
 - 8.2.8. Die häufigsten grafischen Darstellungen
 - 8.2.9. Einführung in die multivariate Analyse und Dimensionsreduktion
- 8.3. Entscheidungsbaum
 - 8.3.1. ID3 Algorithmus
 - 8.3.2. C4.5 Algorithmus
 - 8.3.3. Übertraining und Beschneidung
 - 8.3.4. Analyse der Ergebnisse
- 8.4. Bewertung von Klassifikatoren
 - 8.4.1. Konfusionsmatrizen
 - 8.4.2. Numerische Bewertungsmatrizen
 - 8.4.3. Kappa-Statistik
 - 8.4.5. Die ROC-Kurve

- 8.5. Klassifizierungsregeln
 - 8.5.1. Maßnahmen zur Bewertung von Regeln
 - 8.5.2. Einführung in die grafische Darstellung
 - 8.5.3. Sequentieller Überlagerungsalgorithmus
- 8.6. Neuronale Netze
 - 8.6.1. Grundlegende Konzepte
 - 8.6.2. Einfache neuronale Netze
 - 8.6.3. *Backpropagation*-Algorithmus
 - 8.6.4. Einführung in rekurrente neuronale Netze
- 8.7. Bayessche Methoden
 - 8.7.1. Grundlegende Konzepte der Wahrscheinlichkeit
 - 8.7.2. Bayes-Theorem
 - 8.7.3. Naive Bayes
 - 8.7.4. Einführung in Bayessche Netzwerke
- 8.8. Regressions- und kontinuierliche Antwortmodelle
 - 8.8.1. Einfache lineare Regression
 - 8.8.2. Multiple lineare Regression
 - 8.8.3. Logistische Regression
 - 8.8.4. Regressionsbäume
 - 8.8.5. Einführung in Support Vector Machines (SVM)
 - 8.8.6. Maße für die Anpassungsgüte
- 8.9. *Clustering*
 - 8.9.1. Grundlegende Konzepte
 - 8.9.2. Hierarchisches Clustering
 - 8.9.3. Probabilistische Methoden
 - 8.9.4. EM-Algorithmus
 - 8.9.5. B-Würfel-Methode
 - 8.9.6. Implizite Methoden
- 8.10. *Text Mining* und natürliche Sprachverarbeitung (NLP)
 - 8.10.1. Grundlegende Konzepte
 - 8.10.2. Erstellung eines Korpus
 - 8.10.3. Deskriptive Analyse
 - 8.10.4. Einführung in die Stimmungsanalyse

Modul 9. Multi-Agenten-Systeme und rechnergestützte Wahrnehmung

- 9.1. Agenten und Multiagentensysteme
 - 9.1.1. Konzept des Agenten
 - 9.1.2. Architekturen
 - 9.1.3. Kommunikation und Koordination
 - 9.1.4. Programmiersprachen und Tools
 - 9.1.5. Agent-Anwendungen
 - 9.1.6. Das FIPA
- 9.2. Der Agentenstandard: FIPA
 - 9.2.1. Kommunikation zwischen Akteuren
 - 9.2.2. Die Verwaltung der Akteure
 - 9.2.3. Abstrakte Architektur
 - 9.2.4. Andere Spezifikationen
- 9.3. Die JADE-Plattform
 - 9.3.1. Software-Agenten nach JADE
 - 9.3.2. Architektur
 - 9.3.3. Installation und Ausführung
 - 9.3.4. JADE-Pakete
- 9.4. Grundlegende Programmierung mit JADE
 - 9.4.1. Die Verwaltungskonsole
 - 9.4.2. Grundlegende Erstellung von Agenten
- 9.5. Fortgeschrittene Programmierung mit JADE
 - 9.5.1. Erweiterte Agentenerstellung
 - 9.5.2. Kommunikation zwischen Agenten
 - 9.5.3. Agent-Entdeckung
- 9.6. Computer Vision
 - 9.6.1. Digitale Bildverarbeitung und -analyse
 - 9.6.2. Bildanalyse und Computer Vision
 - 9.6.3. Bildverarbeitung und menschliches Sehen
 - 9.6.4. System zur Bilderfassung
 - 9.6.5. Bildtraining und Wahrnehmung

- 9.7. Digitale Bildanalyse
 - 9.7.1. Phasen des Bildanalyseprozesses
 - 9.7.2. Vorverarbeitung
 - 9.7.3. Grundlegende Operationen
 - 9.7.4. Räumliche Filterung
- 9.8. Digitale Bildtransformation und Bildsegmentierung
 - 9.8.1. Fourier-Transformationen
 - 9.8.2. Frequenzfilterung
 - 9.8.3. Grundlegende Konzepte
 - 9.8.4. Schwellenwert
 - 9.8.5. Kontur-Erkennung
- 9.9. Form-Erkennung
 - 9.9.1. Merkmalsextraktion
 - 9.9.2. Klassifizierungsalgorithmen
- 9.10. Verarbeitung natürlicher Sprache
 - 9.10.1. Automatische Spracherkennungsfunktion
 - 9.10.2. Computerlinguistik
- 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. Multi-Objektive Probleme
 - 10.8.1. Konzept der Dominanz
 - 10.8.2. Anwendung evolutionärer Algorithmen auf multikriterielle Probleme
- 10.9. Neuronale Netze (I)
 - 10.9.1. Einführung in neuronale Netzwerke
 - 10.9.2. Praktisches Beispiel mit neuronalen Netzwerken
- 10.10. Neuronale Netze
 - 10.10.1. Anwendungsbeispiele für neuronale Netze in der medizinischen Forschung
 - 10.10.2. Anwendungsbeispiele für neuronale Netze in der Wirtschaft
 - 10.10.3. Anwendungsfälle für neuronale Netze in der industriellen Bildverarbeitung

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. Explorations-Ausbeutungsraum-Strategien für genetische Algorithmen
 - 10.4.1. CHC-Algorithmus
 - 10.4.2. Multimodale Probleme

- AboutWindow.cpp
- auth.c
- auth.h
 - _auth.h
 - get_access_token.c
- client.c
- client.h
 - _client.h
 - client_block(int, int)
 - client_chunk(int, int)
 - client_connect(char)
 - client_disable()
 - client_enable()
 - client_light(int, int, i)
 - client_login(const cl)
 - client_position(float)
 - client_recv()
 - client_send(char *)
 - client_sign(int, int, i)
 - client_start()
 - client_stop()
 - client_talk(const cha)
 - client_version(int)
 - DEFAULT_PORT
 - get_client_enabled()
- config.h
- ConnectionDialog.cpp
- _make_sphere(float
- _make_sphere(float

```

        continue;
    }
    float du = (tiles[i] % 16) * s;
    float dv = (tiles[i] / 16) * s;
    int flip = ao[i][0] + ao[i][3] > ao[i][1] + ao[i][2];
    for (int v = 0; v < 6; v++) {
        int j = flip ? flipped[i][v] : indices[i][v];
        *(d++) = x + n * positions[i][j][0];
        *(d++) = y + n * positions[i][j][1];
        *(d++) = z + n * positions[i][j][2];
        *(d++) = normals[i][0];
        *(d++) = normals[i][1];
        *(d++) = normals[i][2];
        *(d++) = du + (uvs[i][j][0] ? b : a);
        *(d++) = dv + (uvs[i][j][1] ? b : a);
        *(d++) = ao[i][j];
        *(d++) = light[i][j];
    }
}

void make_cube(
    float *data, float ao[6][4], float light[6][4],
    int left, int right, int top, int bottom, int front, int back,
    float x, float y, float z, float n, int w)
{
    int wleft = blocks[w][0];
    int wright = blocks[w][1];
    int wtop = blocks[w][2];
    int wbottom = blocks[w][3];
    int wfront = blocks[w][4];
    int wback = blocks[w][5];
    make_cube_faces(
        data, ao, light,
        left, right, top, bottom, front, back,
        wleft, wright, wtop, wbottom, wfront, wback,
        x, y, z, n);
}

```


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 aufgibt 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 die Grundlagen der traditionellen Universitäten in der ganzen Welt verschiebt”



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



Die Studenten lernen durch gemeinschaftliche Aktivitäten und reale Fälle die Lösung komplexer Situationen in realen Geschäftsumgebungen.

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 die Jurastudenten das Recht nicht nur anhand theoretischer Inhalte erlernen, sondern ihnen reale, komplexe Situationen vorlegen, damit sie fundierte Entscheidungen treffen und Werturteile darüber fällen können, wie diese zu lösen sind. Sie wurde 1924 als Standardlehrmethode in Harvard eingeführt.

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 Studierenden mit mehreren realen Fällen konfrontiert. Sie müssen Ihr gesamtes Wissen integrieren, recherchieren, argumentieren und Ihre Ideen und Entscheidungen verteidigen.

Relearning Methodik

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 auszubilden. 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 Online-Universität in Spanisch 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 ausgebildet. 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 Ihr Fachgebiet 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 neurokognitiven kontextabhängigen E-Learnings 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 TECH-Online-Arbeitsmethode zu schaffen. Und das alles mit den neuesten Techniken, die dem Studenten qualitativ hochwertige Stücke aus jedem einzelnen Material zur Verfügung stellen.



Meisterklassen

Die Nützlichkeit der Expertenbeobachtung ist wissenschaftlich belegt.

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



Fertigkeiten und Kompetenzen Praktiken

Sie werden Aktivitäten durchführen, um spezifische Kompetenzen und Fertigkeiten in jedem Fachbereich zu entwickeln. Praktiken und Dynamiken 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 haben die Studenten Zugang zu allem, was sie für ihre Ausbildung benötigen.





Fallstudien

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.



Prüfung und Nachprüfung

Die Kenntnisse der Studenten werden während des gesamten Programms regelmäßig durch Bewertungs- und Selbsteinschätzungsaktivitäten und -übungen beurteilt und neu bewertet, so dass die Studenten überprüfen können, wie sie ihre Ziele erreichen.



06

Qualifizierung

Der Privater Masterstudiengang in Künstliche Intelligenz und Wissensmodellierung garantiert neben der strengsten und aktuellsten Ausbildung 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"*

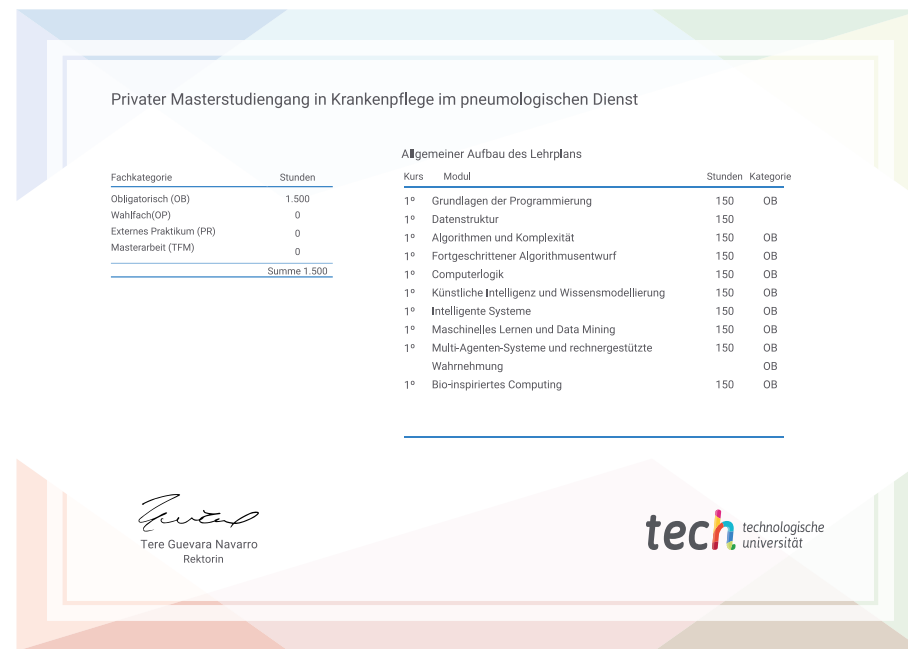
Dieser **Privater Masterstudiengang in Künstliche Intelligenz und Wissensmodellierung** enthält das vollständigste und aktuellste Programm auf dem Markt.

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 Künstliche Intelligenz und Wissensmodellierung**

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 institutionen

virtuelles Klassenzimmer

tech technologische universität

Privater Masterstudiengang Künstliche Intelligenz und Wissensmodellierung

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

Privater Masterstudiengang Künstliche Intelligenz und Wissensmodellierung