Специализированная магистратура

Искусственный интеллект в программировании

tech технологический университет

tech технологический университет

Специализированная магистратура
Искусственный интеллект в программировании

- Формат: **онлайн**
- Продолжительность: **12 месяцев**
- Учебное заведение: **TECH Технологический университет**
- Расписание: **по своему усмотрению**
- Экзамены: **онлайн**

Веб-доступ: www.techtitute.com/ru/information-technology/professional-master-degree/master-artificial-intelligence-programming

Оглавление

01 Презентация — стр. 4

02 Цели — стр. 8

03 Компетенции — стр. 18

04 Руководство курса — стр. 22

05 Структура и содержание — стр. 26

06 Методология — стр. 44

07 Квалификация — стр. 52

01

Презентация

Искусственный интеллект (ИИ) занял прочное место в мире программирования благодаря своей способности автоматизировать сложные задачи, принимать решения на основе данных и учиться на шаблонах. На самом деле ИИ предлагает инструменты и методы, позволяющие создавать более умные и эффективные системы. От алгоритмов машинного обучения, повышающих точность программ, до разработки автономных систем, способных принимать решения в режиме реального времени, — ИИ радикально изменил способы разработки и выполнения кода. В связи с этим TECH разработал академическую программу, которая даст студентам возможность погрузиться в новейшие достижения в этой области, используя революционную методологию *Relearning*.

Программа "Искусственный интеллект в программировании" позволит вам получить целостное представление о том, как искусственный интеллект влияет и улучшает каждый этап разработки программного обеспечения"

Важность искусственного интеллекта в программировании заключается в его способности улучшать и автоматизировать процессы, оптимизировать разработку программного обеспечения и повышать эффективность решения сложных задач. Его способность анализировать большие объемы данных и находить оптимальные решения привела к значительным достижениям в таких областях, как оптимизация алгоритмов, создание более интуитивных интерфейсов и решение сложных задач в различных областях.

Именно поэтому TECH разработал эту Специализированную магистратуру, которая является стратегическим решением для расширения профессиональных возможностей и карьерного роста специалистов в области компьютерных технологий. Программа будет посвящена повышению производительности разработки программного обеспечения с помощью искусственного интеллекта, изучению методов и инструментов для автоматизации процессов, оптимизации кода и ускорения создания интеллектуальных приложений.

Кроме того, программа будет посвящена важнейшей роли искусственного интеллекта в области *QA-тестирования*, внедрению алгоритмов и методов искусственного интеллекта для повышения качества, точности и охвата тестов, более эффективного обнаружения и исправления ошибок. Кроме того, это позволит глубже интегрировать возможности машинного обучения и обработки естественного языка в веб-разработку, создавая интеллектуальные сайты, которые адаптируются и обеспечивают персонализированный пользовательский опыт.

Также будут рассмотрены методы искусственного интеллекта для улучшения удобства использования, взаимодействия и функциональности мобильных приложений, для создания интеллектуальных и предсказуемых приложений, адаптирующихся к поведению пользователя. В программе будет проведен глубокий анализ архитектуры программного обеспечения ИИ, включая различные модели, которые облегчат интеграцию алгоритмов ИИ и их развертывание в производственных средах.

Для того чтобы вырастить высококвалифицированных специалистов в области искусственного интеллекта, TECH разработал комплексную программу, основанную на уникальной методологии *Relearning*. Такой подход позволит студентам закрепить свое понимание путем повторения ключевых понятий.

Данная **Специализированная магистратура в области искусственного интеллекта в программировании** содержит самую полную и современную образовательную программу на рынке. Основными особенностями обучения являются:

- Разбор практических кейсов, представленных экспертами в области искусственного интеллекта в программировании
- Наглядное, схематичное и исключительно практичное содержание курса предоставляет научную и практическую информацию по тем дисциплинам, которые необходимы для осуществления профессиональной деятельности
- Практические упражнения для самооценки, контроля и улучшения успеваемости
- Особое внимание уделяется инновационным методологиям
- Теоретические занятия, вопросы эксперту, дискуссионные форумы по спорным темам и самостоятельная работа
- Учебные материалы курса доступны с любого стационарного или мобильного устройства с выходом в интернет

Вы будете руководить инновационными проектами, адаптированными к требованиям постоянно развивающегося технологического рынка. Чего вы ждете, чтобы поступить?"

Вы погрузитесь в основы архитектуры программного обеспечения, включая производительность, масштабируемость и ремонтопригодность, благодаря самым инновационным мультимедийным ресурсам"

Хотите специализироваться на искусственном интеллекте? В рамках этой программы вы освоите оптимизацию процесса развертывания и интеграцию искусственного интеллекта в облачные вычисления.

Вы изучите интеграцию элементов искусственного интеллекта в Visual Studio Code и оптимизацию кода с помощью ChatGPT, и все это в рамках комплексной академической программы.

В преподавательский состав программы входят профессионалы из данного сектора, которые привносят в обучение опыт своей работы, а также признанные специалисты из ведущих сообществ и престижных университетов.

Мультимедийное содержание программы, разработанное с использованием новейших образовательных технологий, позволит специалисту проходить обучение с учетом контекста и ситуации, т.е. в симулированной среде, обеспечивающей иммерсивный учебный процесс, запрограммированный на обучение в реальных ситуациях.

Структура этой программы основана на проблемно-ориентированном обучении, с помощью которого специалист должен попытаться разрешать различные ситуации из профессиональной практики, возникающие в течение учебного курса. В этом специалистам поможет инновационная интерактивная видеосистема, созданная признанными экспертами.

02

Цели

Основная цель этой программы — предоставить профессионалам доступ к самым передовым знаниям в этой области, при этом подход будет способствовать их всесторонней подготовке. Таким образом, студенты получат возможность пройти эксклюзивную академическую программу в режиме онлайн. Студенты будут обладать полезными современными навыками, от разработки программного обеспечения на основе искусственного интеллекта до проектирования и реализации веб-проектов и мобильных приложений с интеллектом и адаптивностью. С помощью этой программы специалист по информатике выйдет за рамки обычного программирования и станет активным участником технологической революции.

Благодаря TECH вы пройдете весь жизненный цикл тестирования, от создания тестовых примеров до обнаружения ошибок"

Общие цели

- Развивать навыки создания и управления эффективными средами разработки, обеспечивая прочную основу для реализации проектов ИИ
- Приобрести навыки планирования, выполнения и автоматизации тестов качества, используя инструменты искусственного интеллекта для обнаружения и исправления ошибок
- Понимать и применять принципы производительности, масштабируемости и ремонтопригодности при проектировании крупномасштабных вычислительных систем
- Познакомиться с наиболее важными паттернами проектирования и эффективно применять их в архитектуре программного обеспечения

Конкретные цели

Модуль 1. Основы искусственного интеллекта

- Анализировать историческую эволюцию искусственного интеллекта, от его зарождения до современного состояния, определить основные вехи и события
- Понимать функционирование нейронных сетей и их применение в моделях обучения в искусственном интеллекте
- Изучить принципы и применение генетических алгоритмов, проанализировать их полезность для решения сложных задач
- Проанализировать важность тезаурусов, словарей и таксономий в структурировании и обработке данных для систем искусственного интеллекта
- Изучить концепцию семантической паутины и ее влияние на организацию и понимание информации в цифровой среде

Модуль 2. Виды и жизненный цикл данных

- Понимать фундаментальные концепции статистики и их применение в анализе данных
- Определять и классифицировать различные типы статистических данных, от количественных до качественных
- Проанализировать жизненный цикл данных, от создания до утилизации, определив основные этапы
- Изучить начальные этапы жизненного цикла данных, подчеркнув важность планирования данных и их структуры
- Изучить процессы сбора данных, включая методологию, инструменты и каналы сбора
- Изучить концепцию *Datawarehouse* (хранилища данных), уделив особое внимание его составным элементам и дизайну
- Анализировать нормативные аспекты, связанные с управлением данными, соблюдением норм конфиденциальности и безопасности, а также передовым опытом

Модуль 3. Данные в искусственном интеллекте

- Освоить основы науки о данных, включая инструменты, типы и источники для анализа информации
- Изучить процесс преобразования данных в информацию с помощью методов интеллектуального анализа данных и визуализации
- Изучить структуру и характеристики *наборов данных*, понять их важность при подготовке и использовании данных для моделей искусственного интеллекта
- Проанализировать контролируемые и неконтролируемые модели, включая методы и классификацию
- Использовать специальные инструменты и передовые методы обработки данных, обеспечивая эффективность и качество при внедрении искусственного интеллекта

Модуль 4. Добыча данных. Отбор, предварительная обработка и преобразование

- Освоить методы статистического вывода, чтобы понимать и применять статистические методы в анализе данных
- Проводить подробный исследовательский анализ наборов данных для выявления соответствующих закономерностей, аномалий и тенденций
- Развивать навыки подготовки данных, включая их очистку, интеграцию и форматирование для использования в анализе данных
- Реализовывать эффективные стратегии обработки отсутствующих значений в наборах данных, применяя методы вменения или исключения в зависимости от контекста
- Выявлять и устранять шумы в данных, используя методы фильтрации и сглаживания для улучшения качества набора данных
- Решать проблему предварительной обработки данных в средах *больших данных*

Модуль 5. Алгоритм и сложность в искусственном интеллекте

- Представить стратегии разработки алгоритмов, обеспечивающие твердое понимание фундаментальных подходов к решению проблем
- Анализировать эффективность и сложность алгоритмов, применяя методы анализа для оценки производительности с точки зрения времени и пространства
- Изучать и применять алгоритмы сортировки, понимать, как они работают, и сравнивать их эффективность в различных контекстах
- Исследовать алгоритмы деревьев, понять их структуру и области применения
- Изучить алгоритмы с *кучами*, проанализировать их реализацию и полезность для эффективного манипулирования данными
- Анализировать алгоритмы на основе графов, изучая их применение для представления и решения задач со сложными отношениями
- Изучить *жадные* алгоритмы, понять их логику и применение в решении оптимизационных задач
- Изучить и применить технику *обратного пути* для систематического решения проблем, проанализировав ее эффективность в различных сценариях

Модуль 6. Интеллектуальные системы

- Изучить теорию агентов, понять фундаментальные концепции их работы и применения в искусственном интеллекте и программной инженерии
- Изучить представление знаний, включая анализ онтологий и их применение для организации структурированной информации
- Проанализировать концепцию семантической паутины и ее влияние на организацию и поиск информации в цифровой среде
- Оценивать и сравнивать различные представления знаний, интегрируя их для повышения эффективности и точности интеллектуальных систем
- Изучать семантические рассуждения, системы, основанные на знаниях, и экспертные системы, понимая их функциональность и применение в интеллектуальном принятии решений

Модуль 7. Машинное обучение и добыча данных

- Ознакомиться с процессами обнаружения знаний и фундаментальными концепциями машинного обучения
- Изучить деревья решений как модели контролируемого обучения, понять их структуру и области применения
- Оценивать классификаторы с помощью специальных методов для определения их производительности и точности при классификации данных
- Изучить нейронные сети, понять их работу и архитектуру для решения сложных задач машинного обучения
- Изучить байесовские методы и их применение в машинном обучении, включая байесовские сети и байесовские классификаторы
- Проанализировать регрессионные модели и модели непрерывного отклика для прогнозирования числовых значений по данным
- Изучить методы *кластеризации* для выявления закономерностей и структур в немаркированных наборах данных
- Изучить методы интеллектуального анализа текста и обработки естественного языка (NLP), чтобы понять, как методы машинного обучения применяются для анализа и понимания текста

Модуль 8. Нейронные сети, основа *глубокого обучения*

- Освоить основы глубокого обучения, понять его важнейшую роль в *глубоком обучении*
- Изучить фундаментальные операции в нейронных сетях и понять их применение для построения моделей
- Проанализировать различные слои, используемые в нейронных сетях, и научиться выбирать их соответствующим образом
- Понимать эффективное соединение слоев и операций для проектирования сложных и эффективных архитектур нейронных сетей
- Использовать тренеры и оптимизаторы для настройки и улучшения работы нейронных сетей
- Исследовать связь между биологическими и искусственными нейронами для более глубокого понимания дизайна моделей
- Выполнять настройку гиперпараметров для *тонкой настройки* нейронных сетей, оптимизируя их работу на конкретных задачах

Модуль 9. Обучение глубоких нейронных сетей

- Решать проблемы, связанные с градиентом, при обучении глубоких нейронных сетей
- Изучать и применять различные оптимизаторы для повышения эффективности и сходимости моделей
- Программировать скорость обучения, чтобы динамически регулировать скорость сходимости модели
- Понимать и устранять перенастройку с помощью специальных стратегий во время обучения
- Применять практические рекомендации для обеспечения эффективного и результативного обучения глубоких нейронных сетей
- Внедрять *трансферное обучение* в качестве продвинутой техники для улучшения работы модели на конкретных задачах
- Изучать и применять методы *дополнения данных* для обогащения наборов данных и улучшения обобщения моделей
- Разрабатывать практические приложения с использованием *трансферного обучения* для решения реальных задач
- Понимать и применять методы регуляризации для улучшения обобщения и предотвращения перегрузки в глубоких нейронных сетях

Модуль 10. Настройка моделей и обучение с помощью *TensorFlow*

- Освоить основы *TensorFlow* и его интеграцию с NumPy для эффективной обработки данных и вычислений
- Настраивать обучающие модели и алгоритмы, используя расширенные возможности *TensorFlow*
- Изучить API tfdata для эффективного управления и манипулирования наборами данных
- Внедрять формат TFRecord для хранения и доступа к большим наборам данных в *TensorFlow*
- Использовать слои предварительной обработки Keras, чтобы облегчить построение пользовательских моделей
- Изучить проект *TensorFlow Datasets*, чтобы получить доступ к заранее определенным наборам данных и повысить эффективность разработки
- Разработать приложение для *глубокого обучения* с помощью *TensorFlow*, используя знания, полученные в этом модуле
- Использовать все полученные знания на практике при построении и обучении пользовательских моделей с помощью *TensorFlow* в реальных ситуациях

Модуль 11. *Глубокое компьютерное зрение* с использованием конволюционных нейронных сетей

- Понимать архитектуру зрительной коры и ее значение для *глубокого компьютерного зрения*
- Исследовать и применять конволюционные слои для извлечения ключевых характеристик из изображений
- Применять слои кластеризации и использовать их в моделях *глубокого компьютерного зрения с помощью Keras*
- Анализировать различные архитектуры конволюционных нейронных сетей (CNN) и их применимость в различных контекстах
- Разрабатывать и внедрять CNN ResNet с помощью библиотеки Keras для повышения эффективности и производительности модели
- Использовать предварительно обученные модели Keras, чтобы использовать трансферное обучение для решения конкретных задач
- Применять методы классификации и локализации в средах *глубокого компьютерного зрения*
- Изучить стратегии обнаружения и отслеживания объектов с помощью конволюционных нейронных сетей
- Реализовывать методы семантической сегментации для детального понимания и классификации объектов на изображениях

Модуль 12. Обработка естественного языка (NLP) с помощью естественных рекуррентных сетей (RNN) и внимания

- Развивать навыки генерации текста с помощью рекуррентных нейронных сетей (RNN)
- Применять RNN в классификации мнений для анализа настроений в текстах
- Понимать и применять механизмы внимания в моделях обработки естественного языка
- Анализировать и использовать модели *трансформеров* в конкретных задачах NLP
- Изучить применение моделей *трансформеров* в контексте обработки изображений и компьютерного зрения
- Познакомиться с библиотекой *трансформеров Hugging Face* для эффективной реализации продвинутых моделей
- Сравнить различные библиотеки *трансформеров*, чтобы оценить их пригодность для решения конкретных задач
- Разработать практическое приложение NLP, объединяющее RNN и механизмы внимания для решения реальных задач

Модуль 13. *Автоэнкодеры, GAN*, и диффузионные модели

- Разрабатывать эффективные представления данных с помощью *автоэнкодеров*, *GAN* и диффузионных моделей
- Выполнять PCA с использованием неполного линейного автоматического кодировщика для оптимизации представления данных
- Внедрять и понимать работу датчиков автоматической укладки
- Изучать и применять конволюционные автоэнкодеры для эффективного представления визуальных данных
- Анализировать и применять эффективность разреженных автоматических кодеров для представления данных
- Генерировать изображения моды из набора данных MNIST с помощью *автоэнкодеров*
- Понять концепцию генеративных адверсарных сетей (*GAN*) и диффузионных моделей
- Реализовать и сравнить производительность диффузионных моделей и *GAN* при генерации данных

Модуль 14. Биоинспирированные вычисления

- Познакомиться с фундаментальными концепциями биоинспирированных алгоритмов
- Исследовать социально адаптивные алгоритмы как ключевой подход к биоинспирированным алгоритмов
- Анализировать стратегии освоения пространства в генетических алгоритмах
- Изучить модели эволюционных вычислений в контексте оптимизации
- Продолжить детальный анализ моделей эволюционных вычислений
- Применять эволюционное программирование для решения конкретных задач обучения
- Решать сложные многоцелевые задачи в рамках биоинспирированных алгоритмов
- Исследовать применение нейронных сетей в области биоинспирированных алгоритмов
- Углубиться во внедрение и использование нейронных сетей в биоинспирированных вычислениях

Модуль 15. Искусственный интеллект: Стратегии и применение

- Разрабатывать стратегии внедрения искусственного интеллекта в финансовые услуги
- Проанализировать последствия применения искусственного интеллекта для оказания медицинских услуг
- Выявить и оценить риски, связанные с использованием ИИ в сфере здравоохранения
- Оценивать потенциальные риски, связанные с использованием ИИ в промышленности
- Применять методы искусственного интеллекта в промышленности для повышения производительности
- Разрабатывать решения на основе искусственного интеллекта для оптимизации процессов в сфере государственного управления
- Оценивать внедрение технологий ИИ в образовательном секторе
- Применять методы искусственного интеллекта в лесном и сельском хозяйстве для повышения производительности
- Оптимизировать процессы управления персоналом за счет стратегического использования искусственного интеллекта

Модуль 16. Повышение производительности при разработке программного обеспечения с помощью ИИ

- Ознакомиться с внедрением необходимых расширений ИИ в Visual Studio Code для повышения производительности и облегчения разработки *программного обеспечения*
- Получить глубокое понимание основных концепций ИИ и его применения в разработке *программного обеспечения*, включая алгоритмы машинного обучения, обработку естественного языка, нейронные сети и т. д.
- Освоить настройку оптимизированных сред разработки, чтобы студенты могли создавать среды, благоприятные для проектов ИИ
- Применять специальные техники используя ChatGPT для автоматического выявления и исправления возможных улучшений кода, поощряя более эффективные методы программирования
- Способствовать сотрудничеству между различными специалистами в области программирования (от программистов до инженеров по обработке данных и дизайнеров пользовательского опыта) для разработки эффективных и этичных *программных* решений в области ИИ

Модуль 17. Архитектура *программного обеспечения* для *QA-тестирования*

- Развивать навыки разработки надежных планов тестирования, охватывающих различные виды *тестирования* и обеспечивающих качество программного обеспечения
- Распознавать и анализировать различные типы *программных* фреймворков, такие как монолитный, микросервисный или сервис-ориентированный
- Получить полное представление о принципах и методах проектирования масштабируемых компьютерных систем, способных обрабатывать большие объемы данных
- Применять передовые знания в реализации структур данных, основанных на искусственном интеллекте, для оптимизации производительности и эффективности *программного* обеспечения
- Разрабатывать безопасные методы разработки, уделяя особое внимание предотвращению уязвимостей для обеспечения безопасности *программного обеспечения* на архитектурном уровне

Модуль 18. Веб-проекты с помощью ИИ

- Развивать комплексные навыки реализации веб-проектов, начиная с *фронтенд-дизайна* и заканчивая оптимизацией *бэкенда*, с включением элементов ИИ
- Оптимизировать процесс развертывания веб-сайта, применяя методы и инструменты для повышения скорости и эффективности
- Интегрировать ИИ в облачные вычисления, что позволит студентам создавать высокомасштабируемые и эффективные веб-проекты
- Приобрести способность определять конкретные проблемы и возможности в веб-проектах, где может быть эффективно применен ИИ, например, обработка текста, персонализация, рекомендация контента и т.д.
- Побудить студентов быть в курсе последних тенденций и разработок в области ИИ для правильного применения в веб-проектах

Модуль 19. Мобильные приложения с помощью ИИ

- Применять передовые концепции *чистой архитектуры*, источников данных и репозиториев для обеспечения надежной и модульной структуры в мобильных приложениях с ИИ
- Развивать навыки разработки интерактивных экранов, иконок и графических активов с использованием ИИ для улучшения пользовательского опыта в мобильных приложениях
- Углубить знания в области настройки фреймворка для мобильных приложений и использовать *Github Copilot* для ускорения процесса разработки

- Оптимизировать мобильные приложения с помощью ИИ для эффективной работы с учетом управления ресурсами и использования данных
- Проводить тестирование качества мобильных приложений с помощью ИИ, позволяя студентам выявлять проблемы и отлаживать ошибки

Модуль 20. ИИ для *QA-тестирования*

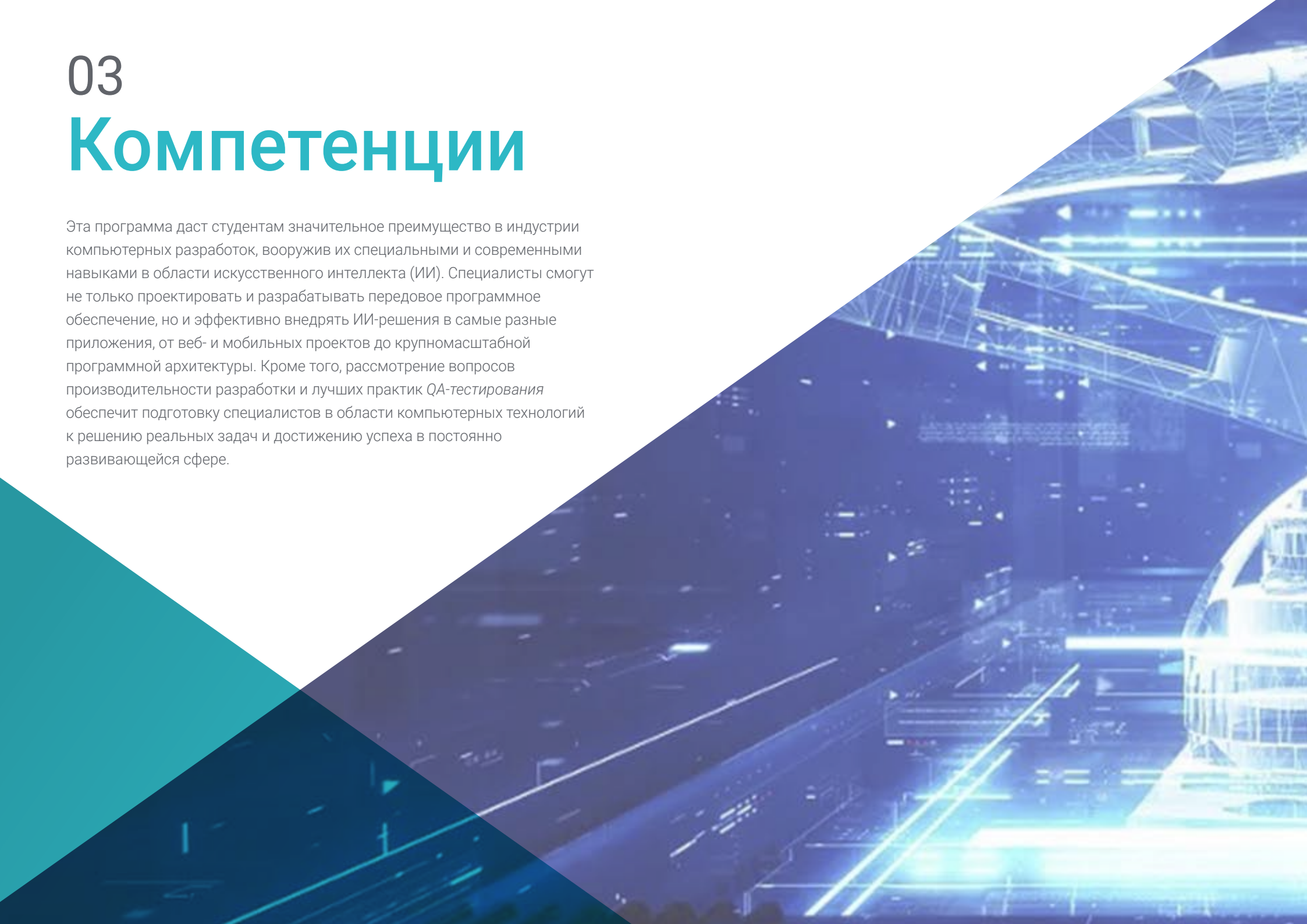
- Освоить принципы и методы проектирования масштабируемых компьютерных систем, способных обрабатывать большие объемы данных
- Применять передовые знания в реализации структур данных, основанных на искусственном интеллекте, для оптимизации производительности и эффективности *программного* обеспечения
- Понимать и применять практики безопасной разработки, уделяя особое внимание предотвращению таких уязвимостей, как инъекции, для обеспечения безопасности *программного* обеспечения на архитектурном уровне
- Создавать автоматизированные тесты, особенно в веб- и мобильных средах, интегрируя инструменты ИИ для повышения эффективности процесса
- Использовать передовые инструменты QA на основе ИИ для более эффективного обнаружения *ошибок* и непрерывного совершенствования *программного обеспечения*

"Вы овладеете технологиями будущего, пройдя обучение по данной эксклюзивной программе университета на 100% в онлайн-формате. Только с TECH!"

03

Компетенции

Эта программа даст студентам значительное преимущество в индустрии компьютерных разработок, вооружив их специальными и современными навыками в области искусственного интеллекта (ИИ). Специалисты смогут не только проектировать и разрабатывать передовое программное обеспечение, но и эффективно внедрять ИИ-решения в самые разные приложения, от веб- и мобильных проектов до крупномасштабной программной архитектуры. Кроме того, рассмотрение вопросов производительности разработки и лучших практик *QA-тестирования* обеспечит подготовку специалистов в области компьютерных технологий к решению реальных задач и достижению успеха в постоянно развивающейся сфере.



Благодаря этой университетской программе вы сможете внедрять алгоритмы искусственного интеллекта в веб-проекты и мобильные приложения"

Общие профессиональные навыки

- Применять расширения искусственного интеллекта в Visual Studio Code и *некодовые* методы проектирования для повышения эффективности разработки программного обеспечения
- Использовать ChatGPT для оптимизации и улучшения качества кода, применяя передовые методы программирования
- Реализовывать веб-проекты, от создания *рабочего пространства* до развертывания, интегрируя искусственный интеллект, как *фронтенд*, так и *бэкенд*
- Разрабатывать мобильные приложения с искусственным интеллектом — от настройки среды до создания расширенных функций и управления графическими активами
- Применять передовые концепции хранения данных и структуры данных на основе искусственного интеллекта для повышения эффективности и масштабируемости систем
- Применять методы безопасной разработки, избегая таких уязвимостей, как инъекции, для обеспечения целостности и безопасности разработанного программного обеспечения

Вы сможете разрабатывать персонализированные и интуитивно понятные пользовательские решения с помощью искусственного интеллекта. Поступайте сейчас!"

Профессиональные навыки

- Применять методы и стратегии искусственного интеллекта для повышения эффективности в сфере *розничной* торговли
- Внедрять методы шумоподавления с помощью автоматических кодировщиков
- Эффективно создавать обучающие наборы данных для задач обработки естественного языка (NLP)
- Выполнять слои кластеризации и их использование в моделях *глубокого компьютерного зрения с помощью Keras*
- Использовать функции и графики *TensorFlow* для оптимизации производительности пользовательских моделей
- Оптимизировать разработку и применение *чат-ботов* и виртуальных помощников, понимая, как они работают и каковы возможности их применения
- Освоить повторное использование предварительно обученных слоев, чтобы оптимизировать и ускорить процесс обучения
- Построить первую нейронную сеть, применяя изученные концепции на практике
- Активировать многослойный перцептрон (MLP) с помощью библиотеки Keras
- Применять методы исследования и предварительной обработки данных, выявляя и подготавливая их для эффективного использования в моделях машинного обучения
- Изучить языки и программное обеспечение для создания онтологий, используя специальные инструменты для разработки семантических моделей
- Разрабатывать методы очистки данных для обеспечения качества и точности информации, используемой в последующем анализе
- Освоить настройку оптимизированных сред разработки, чтобы студенты могли создавать среды, благоприятные для проектов ИИ
- Применять специальные техники, используя ChatGPT для автоматического выявления и исправления возможных улучшений кода, поощряя более эффективные методы программирования
- Создавать автоматизированные тесты, особенно в веб- и мобильных средах, интегрируя инструменты искусственного интеллекта для повышения эффективности процесса
- Использовать передовые инструменты QA на основе искусственного интеллекта для более эффективного обнаружения ошибок и непрерывного совершенствования программного обеспечения
- Интегрировать искусственный интеллект в облачные вычисления, что позволит студентам создавать высокомасштабируемые и эффективные веб-проекты
- Устанавливать фреймворк для мобильных приложений и использовать Github Copilot, чтобы ускорить процесс разработки

04

Руководство курса

Стремясь к элитному преподаванию, TECH тщательно отобрал преподавателей, ответственных за разработку учебного плана этой программы. По этой причине над этой академической программой работает опытный преподавательский состав, обладающий выдающимися знаниями в области применения искусственного интеллекта в программировании. Таким образом, студенты этой Специализированной магистратуры получат доступ к первоклассному образованию с эксклюзивным сочетанием различных аудиовизуальных средств для более эффективной и динамичной интеграции знаний.

Узнайте о последних тенденциях в области искусственного интеллекта, применяемого в программировании, от лучших экспертов в этой области"

Руководство

Д-р Перальта Мартин-Паломино, Артуро

- CEO и CTO Prometeus Global Solutions
- CTO в Korporate Technologies
- CTO в AI Shephers GmbH
- Консультант и советник в области стратегического бизнеса в Alliance Medical
- Руководитель в области дизайна и разработки в компании DocPath
- Руководитель в области компьютерной инженерии в Университете Кастилии-ла-Манча
- Степень доктора в области экономики, бизнеса и финансов Университета Камило Хосе Села
- Степень доктора в области психологии Университета Кастилии-ла-Манча
- Степень магистра Executive MBA Университета Изабель I
- Степень магистра в области управления коммерцией и маркетингом Университета Изабель I
- Степень магистра в области больших данных по программе Hadoop
- Степень магистра в области передовых информационных технологий Университета Кастилии-Ла-Манча
- Член: Исследовательская группа SMILE

Г-н Кастельянос Эррерос, Рикардо

- *Директор по технологиям* в OWQLO
- Специалист по проектированию компьютерных систем и *инженер* по *машинному* обучению
- *Внештатный* технический консультант
- Разработчик мобильных приложений для eDreams, Fnac, Air Europa, Bankia, Cetelem, Banco Santander, Santillana, Groupón и Grupo Planeta
- Разработчик веб-сайтов для Openbank и Banco Santander
- Инженерное профессиональное образование в области компьютерных систем Университета Кастильи-ла-Манчи

05

Структура и содержание

Программа "Искусственный интеллект в программировании" отличается своим целостным подходом, направленным не только на реализацию интеллектуальных алгоритмов, но и на повышение производительности при разработке программного обеспечения и применение ИИ в таких ключевых областях, как *QA-тестирование*, веб-проекты, мобильные приложения и архитектура программного обеспечения. Сочетание технических навыков, передовых инструментов и практического применения ИИ на различных этапах разработки позиционирует эту программу как ведущую, дающую специалистам полное и глубокое понимание применения ИИ в программировании.

Вы изучите практическое применение искусственного интеллекта в веб-проектах, включая как фронтенд, так и бэкенд-разработку"

Модуль 1. Основы искусственного интеллекта

1.1. История искусственного интеллекта
- 1.1.1. Когда мы начали говорить об искусственном интеллекте?
- 1.1.2. Упоминания в кино
- 1.1.3. Важность искусственного интеллекта
- 1.1.4. Технологии, обеспечивающие и поддерживающие искусственный интеллект

1.2. Искусственный интеллект в играх
- 1.2.1. Теория игр
- 1.2.2. *Минимакс* и Альфа-бета-отсечение
- 1.2.3. Моделирование: Монте-Карло

1.3. Нейронные сети
- 1.3.1. Биологические основы
- 1.3.2. Вычислительная модель
- 1.3.3. Контролируемые и неконтролируемые нейронные сети
- 1.3.4. Простой перцептрон
- 1.3.5. Многослойный перцептрон

1.4. Генетические алгоритмы
- 1.4.1. История
- 1.4.2. Биологическая основа
- 1.4.3. Кодирование проблемы
- 1.4.4. Генерация начальной популяции
- 1.4.5. Основной алгоритм и генетические операторы
- 1.4.6. Оценка отдельных лиц: Fitness

1.5. Тезаурусы, словари, таксономии
- 1.5.1. Словари
- 1.5.2. Таксономия
- 1.5.3. Тезаурусы
- 1.5.4. Онтологии
- 1.5.5. Представление знаний: семантическая паутина

1.6. Семантическая паутина
- 1.6.1. Спецификация: RDF, RDFS и OWL
- 1.6.2. Выводы/рассуждения
- 1.6.3. *Linked Data*

1.7. Экспертные системы и DSS
- 1.7.1. Экспертные системы
- 1.7.2. Системы поддержки принятия решений

1.8. *Чатботы* и виртуальные помощники
- 1.8.1. Типы помощников: голосовые и текстовые помощники
- 1.8.2. Основополагающие детали для развития помощника: *Намерения*, структура и диалог
- 1.8.3. Интеграция: web, *Slack*, Whatsapp, Facebook
- 1.8.4. Инструменты разработки помощников: *Dialog Flow*, *Watson Assistant*

1.9. Стратегия и внедрение ИИ

1.10. Будущее искусственного интеллекта
- 1.10.1. Мы понимаем, как определять эмоции с помощью алгоритмов
- 1.10.2. Создание личности: язык, выражения и содержание
- 1.10.3. Тенденции искусственного интеллекта
- 1.10.4. Размышления

Модуль 2. Виды и жизненный цикл данных

2.1. Статистика
- 2.1.1. Статистика: описательная статистика, статистические выводы
- 2.1.2. Население, выборка, индивидуум
- 2.1.3. Переменные: определение, шкалы измерения

2.2. Типы статистических данных
- 2.2.1. По типу
  - 2.2.1.1. Количественные: непрерывные данные и дискретные данные
  - 2.2.1.2. Качественные: биномиальные данные, номинальные данные, порядковые данные
- 2.2.2. По форме
  - 2.2.2.1. Числовые
  - 2.2.2.2. Текст
  - 2.2.2.3. Логические
- 2.2.3. Согласно источнику
  - 2.2.3.1. Первичные
  - 2.2.3.2. Вторичные

2.3. Жизненный цикл данных
- 2.3.1. Этапы цикла
- 2.3.2. Основные этапы цикла
- 2.3.3. Принципы FAIR

2.4. Начальные этапы цикла
- 2.4.1. Определение целей
- 2.4.2. Определение необходимых ресурсов
- 2.4.3. Диаграмма Гантта
- 2.4.4. Структура данных

2.5. Сбор данных
- 2.5.1. Методология сбора
- 2.5.2. Инструменты сбора
- 2.5.3. Каналы сбора

2.6. Очистка данных
- 2.6.1. Этапы очистки данных
- 2.6.2. Качество данных
- 2.6.3. Работа с данными (с помощью R)

2.7. Анализ данных, интерпретация и оценка результатов
- 2.7.1. Статистические меры
- 2.7.2. Индексы отношений
- 2.7.3. Добыча данных

2.8. Хранилище данных (*datawarehouse*)
- 2.8.1. Элементы, входящие в его состав
- 2.8.2. Разработка
- 2.8.3. Аспекты, которые следует учитывать

2.9. Доступность данных
- 2.9.1. Доступ
- 2.9.2. Полезность
- 2.9.3. Безопасность

2.10. Нормативно-правовые аспекты
- 2.10.1. Закон о защите данных
- 2.10.2. Передовая практика
- 2.10.3. Другие нормативные аспекты

Модуль 3. Данные в искусственном интеллекте

3.1. Наука о данных
- 3.1.1. Наука о данных
- 3.1.2. Передовые инструменты для исследователя данных

3.2. Данные, информация и знания
- 3.2.1. Данные, информация и знания
- 3.2.2. Типы данных
- 3.2.3. Источники данных

3.3. От данных к информации
- 3.3.1. Анализ данных
- 3.3.2. Виды анализа
- 3.3.3. Извлечение информации из *набора данных*

3.4. Извлечение информации путем визуализации
- 3.4.1. Визуализация как инструмент анализа
- 3.4.2. Методы визуализации
- 3.4.3. Визуализация набора данных

3.5. Качество данных
- 3.5.1. Данные о качестве
- 3.5.2. Очистка данных
- 3.5.3. Основная предварительная обработка данных

3.6. *Набор данных*
- 3.6.1. Обогащение *набора данных*
- 3.6.2. Проклятие размерности
- 3.6.3. Модификация нашего набора данных

3.7. Выведение из равновесия
- 3.7.1. Дисбаланс классов
- 3.7.2. Методы устранения дисбаланса
- 3.7.3. Сбалансированность *набора данных*

3.8. Модели без контроля
- 3.8.1. Модель без контроля
- 3.8.2. Методы
- 3.8.3. Классификация с помощью моделей без контроля

3.9. Модели под контролем
- 3.9.1. Модель под контролем
- 3.9.2. Методы
- 3.9.3. Классификация с помощью моделей под контролем

3.10. Инструменты и передовой опыт
- 3.10.1. Передовая практика для специалиста по исследованию данных
- 3.10.2. Лучшая модель
- 3.10.3. Полезные инструменты

Модуль 4. Добыча данных. Отбор, предварительная обработка и преобразование

4.1. Статистический вывод
- 4.1.1. Описательная статистика vs. Статистическое заключение
- 4.1.2. Параметрические методы
- 4.1.3. Непараметрические методы

4.2. Исследовательский анализ
- 4.2.1. Описательный анализ
- 4.2.2. Визуализация
- 4.2.3. Подготовка данных

4.3. Подготовка данных
- 4.3.1. Интеграция и очистка данных
- 4.3.2. Нормализация данных
- 4.3.3. Преобразование данных

4.4. Отсутствующие данные
- 4.4.1. Обработка отсутствующих значений
- 4.4.2. Метод максимального правдоподобия
- 4.4.3. Обработка отсутствующих данных в машинном обучении

4.5. Шум в данных
- 4.5.1. Классы и признаки шума
- 4.5.2. Фильтрация шумов
- 4.5.3. Шумовой эффект

4.6. Проклятие размерности
- 4.6.1. *Oversampling*
- 4.6.2. *Undersampling*
- 4.6.3. Редукция многомерных данных

4.7. От непрерывных к дискретным признакам
- 4.7.1. Непрерывные и дискретные данные
- 4.7.2. Процесс дискретизации

4.8. Данные
- 4.8.1. Выбор данных
- 4.8.2. Перспективы и критерии отбора
- 4.8.3. Методы отбора

4.9. Выбор экземпляров
- 4.9.1. Методы выбора экземпляра
- 4.9.2. Выбор прототипов
- 4.9.3. Расширенные методы выбора экземпляра

4.10. Предварительная обработка больших данных

Модуль 5. Алгоритм и сложность в искусственном интеллекте

5.1. Введение в шаблоны разработки алгоритмов
- 5.1.1. Рекурсия
- 5.1.2. "Разделяй и властвуй"
- 5.1.3. Другие стратегии

5.2. Эффективность и анализ работы алгоритмов
- 5.2.1. Меры эффективности
- 5.2.2. Измерение объема данных на входе
- 5.2.3. Измерение времени выполнения
- 5.2.4. Случаи: худший, лучший и средний
- 5.2.5. Асимптотическая нотация
- 5.2.6. Критерии математического анализа нерекурсивных алгоритмов
- 5.2.7. Критерии математического анализа рекурсивных алгоритмов
- 5.2.8. Эмпирический анализ алгоритмов

5.3. Алгоритмы сортировки
- 5.3.1. Концепция сортировки
- 5.3.2. Пузырьковая сортировка
- 5.3.3. Сортировка выбором
- 5.3.4. Сортировка вставками
- 5.3.5. Сортировка слиянием (*Merge_Sort*)
- 5.3.6. Быстрая сортировка (*Quick_Sort*)

5.4. Алгоритмы с применением деревьев
- 5.4.1. Концепция дерева
- 5.4.2. Бинарные деревья
- 5.4.3. Обходы деревьев
- 5.4.4. Представление выражений
- 5.4.5. Упорядоченные бинарные деревья
- 5.4.6. Сбалансированные бинарные деревья

5.5. Алгоритмы с применением *кучей*
- 5.5.1. Что такое *кучи*
- 5.5.2. Алгоритм *сортировки кучей*
- 5.5.3. Очереди с приоритетом

5.6. Алгоритмы на графах
- 5.6.1. Представление
- 5.6.2. Обход в ширину
- 5.6.3. Обход в глубину
- 5.6.4. Топологическая сортировка

5.7. *Жадные* алгоритмы
- 5.7.1. *Жадная* стратегия
- 5.7.2. Элементы *жадной* стратегии
- 5.7.3. Обмен монет
- 5.7.4. Задача коммивояжера
- 5.7.5. Задача о рюкзаке

5.8. Поиск кратчайших путей
- 5.8.1. Задача о кратчайшем пути
- 5.8.2. Отрицательные дуги и циклы
- 5.8.3. Алгоритм Дейкстры

5.9. *Жадные* алгоритмы на графах
- 5.9.1. Минимальное остовное дерево
- 5.9.2. Алгоритм Прима
- 5.9.3. Алгоритм Краскала
- 5.9.4. Анализ сложности

5.10. *Техника Backtracking*
- 5.10.1. Техника *Backtracking*
- 5.10.2. Альтернативные техники

Модуль 6. Интеллектуальные системы

6.1. Теория агентов
- 6.1.1. История концепции
- 6.1.2. Определение агента
- 6.1.3. Агенты в системах искусственного интеллекта
- 6.1.4. Агенты в программной инженерии

6.2. Архитектуры агентов
- 6.2.1. Процесс рассуждения агента
- 6.2.2. Реактивные агенты
- 6.2.3. Дедуктивные агенты
- 6.2.4. Гибридные агенты
- 6.2.5. Сравнение

6.3. Информация и знания
- 6.3.1. Различие между данными, информацией и знаниями
- 6.3.2. Оценка качества данных
- 6.3.3. Методы сбора данных
- 6.3.4. Методы получения информации
- 6.3.5. Методы приобретения знаний

6.4. Представление знаний
- 6.4.1. Важность представления знаний
- 6.4.2. Определение представления знаний через их роли
- 6.4.3. Характеристики представления знаний

6.5. Онтологии
- 6.5.1. Введение в метаданные
- 6.5.2. Философская концепция онтологии
- 6.5.3. Вычислительная концепция онтологии
- 6.5.4. Онтологии доменов и онтологии более высокого уровня
- 6.5.5. Как создать онтологию?

6.6. Онтологические языки и программное обеспечение для создания онтологий
- 6.6.1. Семантическая тройка RDF, *Turtle* и N
- 6.6.2. RDF *Schema*
- 6.6.3. OWL
- 6.6.4. SPARQL
- 6.6.5. Знакомство с различными инструментами для создания онтологий
- 6.6.6. Установка и использование *Protégé*

6.7. Семантическая паутина
- 6.7.1. Текущее состояние и будущее семантической паутины
- 6.7.2. Семантические веб-приложения

6.8. Другие модели представления знаний
- 6.8.1. Словари
- 6.8.2. Обзор
- 6.8.3. Таксономия
- 6.8.4. Тезаурусы
- 6.8.5. Фолксономии
- 6.8.6. Сравнение
- 6.8.7. Карты разума

6.9. Оценка и интеграция представлений знаний
- 6.9.1. Логика нулевого порядка
- 6.9.2. Логика первого порядка
- 6.9.3. Дескрипционная логика
- 6.9.4. Взаимосвязь между различными типами логики
- 6.9.5. *Пролог*: программирование на основе логики первого порядка

6.10. Семантические анализаторы, системы, основанные на знаниях, и экспертные системы
- 6.10.1. Концепция анализатора
- 6.10.2. Применение анализатора
- 6.10.3. Системы, основанные на знаниях
- 6.10.4. MYCIN, история экспертных систем
- 6.10.5. Элементы и архитектура экспертных систем
- 6.10.6. Создание экспертных систем

Модуль 7. Машинное обучение и добыча данных

7.1. Введение в процессы обнаружения знаний и основные концепции машинного обучения
- 7.1.1. Ключевые понятия процесса обнаружения знаний
- 7.1.2. Исторический взгляд процесса обнаружения знаний
- 7.1.3. Этапы процесса обнаружения знаний
- 7.1.4. Методы, используемые в процессах обнаружения знаний
- 7.1.5. Характеристики хороших моделей машинного обучения
- 7.1.6. Типы информации машинного обучения
- 7.1.7. Основные концепции обучения
- 7.1.8. Основные концепции обучения без контроля

7.2. Исследование и предварительная обработка данных
- 7.2.1. Обработка данных
- 7.2.2. Обработка данных в потоке анализа данных
- 7.2.3. Типы данных
- 7.2.4. Преобразование данных
- 7.2.5. Визуализация и исследование непрерывных переменных
- 7.2.6. Визуализация и исследование категориальных переменных
- 7.2.7. Корреляционные меры
- 7.2.8. Наиболее распространенные графические представления
- 7.2.9. Введение в многомерный анализ и снижение размерности

7.3. Деревья решений
- 7.3.1. Алгоритм ID
- 7.3.2. Алгоритм C
- 7.3.3. Перегрузка и обрезка
- 7.3.4. Анализ результатов

7.4. Оценка классификаторов
- 7.4.1. Матрицы путаницы
- 7.4.2. Матрицы численной оценки
- 7.4.3. Карра-статистика
- 7.4.4. ROC-кривая

7.5. Правила классификации
- 7.5.1. Меры по оценке правил
- 7.5.2. Введение в графическое представление
- 7.5.3. Алгоритм последовательного оверлея

7.6. Нейронные сети
- 7.6.1. Основные понятия
- 7.6.2. Простые нейронные сети
- 7.6.3. Алгоритм *Backpropagation*
- 7.6.4. Введение в рекуррентные нейронные сети

7.7. Байесовские методы
- 7.7.1. Основные понятия вероятности
- 7.7.2. Теорема Байеса
- 7.7.3. Наивный Байес
- 7.7.4. Введение в байесовские сети

7.8. Регрессия и модели непрерывного отклика
- 7.8.1. Простая линейная регрессия
- 7.8.2. Множественная линейная регрессия
- 7.8.3. Логистическая регрессия
- 7.8.4. Деревья регрессии
- 7.8.5. Введение в машины опорных векторов (SVM)
- 7.8.6. Меры соответствия

7.9. *Кластеризация*
- 7.9.1. Основные понятия
- 7.9.2. Иерархическая *кластеризация*
- 7.9.3. Вероятностные методы
- 7.9.4. Алгоритм EM
- 7.9.5. Метод *B-Cubed*
- 7.9.6. Неявные методы

7.10. Интеллектуальный анализ текста и обработка естественного языка (NLP)
- 7.10.1. Основные понятия
- 7.10.2. Создание корпуса
- 7.10.3. Описательный анализ
- 7.10.4. Введение в анализ чувств

Модуль 8. Нейронные сети, основа *глубокого обучения*

8.1. Глубокое обучение
- 8.1.1. Виды глубокого обучения
- 8.1.2. Области применения глубокого обучения
- 8.1.3. Преимущества и недостатки глубокого обучения

8.2. Операции
- 8.2.1. Сумма
- 8.2.2. Продукт
- 8.2.3. Перевод

8.3. Слои
- 8.3.1. Входной слой
- 8.3.2. Скрытый слой
- 8.3.3. Выходной слой

8.4. Склеивание слоев и операции
- 8.4.1. Проектирование архитектур
- 8.4.2. Соединение между слоями
- 8.4.3. Распространение вперед

8.5. Построение первой нейронной сети
- 8.5.1. Проектирование сети
- 8.5.2. Определение весов
- 8.5.3. Практика сети

8.6. Тренажер и оптимизатор
- 8.6.1. Выбор оптимизатора
- 8.6.2. Установление функции потерь
- 8.6.3. Установление метрики

8.7. Применение принципов нейронных сетей
- 8.7.1. Функции активации
- 8.7.2. Обратное распространение
- 8.7.3. Установка параметров

8.8. От биологических нейронов к искусственным
- 8.8.1. Функционирование биологического нейрона
- 8.8.2. Передача знаний искусственным нейронам
- 8.8.3. Установление взаимоотношений между ними

8.9. Реализация MLP (многослойного перцептрона) с помощью Keras
- 8.9.1. Определение структуры сети
- 8.9.2. Составление модели
- 8.9.3. Обучение модели

8.10. *Тонкая настройка* гиперпараметров нейронных сетей
- 8.10.1. Выбор функции активации
- 8.10.2. Установка *скорости обучения*
- 8.10.3. Установка весов

Модуль 9. Обучение глубоких нейронных сетей

9.1. Градиентные задачи
- 9.1.1. Методы оптимизации градиента
- 9.1.2. Стохастические градиенты
- 9.1.3. Методы инициализации весов

9.2. Повторное использование предварительно обученных слоев
- 9.2.1. Перенос результатов обучения
- 9.2.2. Извлечение признаков
- 9.2.3. Глубокое обучение

9.3. Оптимизаторы
- 9.3.1. Стохастические оптимизаторы градиентного спуска
- 9.3.2. Оптимизаторы Adam и *RMSprop*
- 9.3.3. Современные оптимизаторы

9.4. Программирование скорости обучения
- 9.4.1. Автоматическое управление скоростью обучения
- 9.4.2. Циклы обучения
- 9.4.3. Условия сглаживания

9.5. Переоценка
- 9.5.1. Перекрестная валидация
- 9.5.2. Регуляризация
- 9.5.3. Метрики оценки

9.6. Практические рекомендации
- 9.6.1. Конструкция модели
- 9.6.2. Выбор метрик и параметров оценки
- 9.6.3. Проверка гипотез

9.7. *Трансферное обучение*
- 9.7.1. Перенос результатов обучения
- 9.7.2. Извлечение признаков
- 9.7.3. Глубокое обучение

9.8. *Расширение данных*
- 9.8.1. Преобразования изображений
- 9.8.2. Формирование синтетических данных
- 9.8.3. Преобразование текста

9.9. Практическое применение *трансферного обучения*
- 9.9.1. Перенос результатов обучения
- 9.9.2. Извлечение признаков
- 9.9.3. Глубокое обучение

9.10. Регуляризация
- 9.10.1. L и L
- 9.10.2. Регуляризация по принципу максимальной энтропии
- 9.10.3. *Dropout*

Модуль 10. Настройка моделей и обучение с помощью *TensorFlow*

10.1. *TensorFlow*
- 10.1.1. Использование библиотеки *TensorFlow*
- 10.1.2. Обучение модели с помощью *TensorFlow*
- 10.1.3. Операции с графиками в *TensorFlow*

10.2. *TensorFlow* и NumPy
- 10.2.1. Вычислительная среда NumPy для *TensorFlow*
- 10.2.2. Использование массивов NumPy в *TensorFlow*
- 10.2.3. Операции NumPy для графиков *TensorFlow*

10.3. Настройка моделей и алгоритмов обучения
- 10.3.1. Построение пользовательских моделей с помощью *TensorFlow*
- 10.3.2. Управление параметрами обучения
- 10.3.3. Использование методов оптимизации для обучения

10.4. Функции и графики *TensorFlow*
- 10.4.1. Функции в *TensorFlow*
- 10.4.2. Использование графиков для обучения модели
- 10.4.3. Оптимизация графов с помощью операций *TensorFlow*

10.5. Загрузка и предварительная обработка данных с помощью *TensorFlow*
- 10.5.1. Загрузка наборов данных с помощью *TensorFlow*
- 10.5.2. Предварительная обработка данных с помощью *TensorFlow*
- 10.5.3. Использование инструментов *TensorFlow* для манипулирования данными

10.6. API *tfdata*
- 10.6.1. Использование API *tfdata* для обработки данных
- 10.6.2. Построение потоков данных с помощью *tfdata*
- 10.6.3. Использование API *tfdata* для обучения моделей

10.7. Формат *TFRecord*
- 10.7.1. Использование API *TFRecord* для сериализации данных
- 10.7.2. Загрузка файлов *TFRecord* с помощью *TensorFlow*
- 10.7.3. Использование файлов *TFRecord* для обучения моделей

10.8. Слои предварительной обработки в Keras
- 10.8.1. Использование API предварительной обработки в Keras
- 10.8.2. Построение *pipelined* предварительной обработки с помощью Keras
- 10.8.3. Использование API предварительной обработки в Keras для обучения моделей

10.9. Проект *TensorFlow Datasets*
- 10.9.1. Использование *TensorFlow Datasets* для загрузки данных
- 10.9.2. Предварительная обработка данных с помощью *TensorFlow Datasets*
- 10.9.3. Использование *TensorFlow Datasets* для обучения моделей

10.10. Построение приложения глубокого *обучения* с помощью *TensorFlow*
- 10.10.1. Практическое применение
- 10.10.2. Построение приложения глубокого *обучения* с помощью *TensorFlow*
- 10.10.3. Обучение модели с помощью *TensorFlow*
- 10.10.4. Использование приложения для прогнозирования результатов

Модуль 11. *Глубокое компьютерное зрение* с использованием конволюционных нейронных сетей

11.1. Архитектура *Visual Cortex*
- 11.1.1. Функции зрительной коры
- 11.1.2. Теории вычислительного зрения
- 11.1.3. Модели обработки изображений

11.2. Конволюционные слои
- 11.2.1. Повторное использование весов в свертке
- 11.2.2. Конволюция D
- 11.2.3. Функции активации

11.3. Слои кластеризации и реализация слоев кластеризации с помощью Keras
- 11.3.1. *Pooling* и *Striding*
- 11.3.2. *Flattening*
- 11.3.3. Виды *Pooling*

11.4. Архитектуры CNN
- 11.4.1. Архитектура VGG
- 11.4.2. Архитектура *AlexNet*
- 11.4.3. Архитектура *ResNet*

11.5. Реализация CNN *ResNet* - с использованием Keras
- 11.5.1. Инициализация весов
- 11.5.2. Определение входного слоя
- 11.5.3. Определение выходного слоя

11.6. Использование предварительно обученных моделей Keras
- 11.6.1. Характеристики предварительно обученных моделей
- 11.6.2. Использование предварительно обученных моделей
- 11.6.3. Преимущества предварительно обученных моделей

11.7. Предварительно обученные модели для трансферного обучения
- 11.7.1. Трансферное обучение
- 11.7.2. Процесс трансферного обучения
- 11.7.3. Преимущества трансферного обучения

11.8. Классификация и локализация в *глубоком компьютерном зрении*
- 11.8.1. Классификация изображений
- 11.8.2. Определение местоположения объектов на изображениях
- 11.8.3. Обнаружение объектов

11.9. Обнаружение объектов и их отслеживание
- 11.9.1. Методы обнаружения объектов
- 11.9.2. Алгоритмы отслеживания объектов
- 11.9.3. Методы отслеживания и трассировки

11.10. Семантическая сегментация
- 11.10.1. Глубокое обучение для семантической сегментации
- 11.10.2. Обнаружение краев
- 11.10.3. Методы сегментации, основанные на правилах

Модуль 12. Обработка естественного языка (NLP) с помощью естественных рекуррентных сетей (RNN) и внимания

12.1. Генерация текста с использованием RNN
- 12.1.1. Обучение RNN для генерации текста
- 12.1.2. Генерация естественного языка с помощью RNN
- 12.1.3. Приложения для генерации текста с помощью RNN

12.2. Создание обучающего набора данных
- 12.2.1. Подготовка данных для обучения RNN
- 12.2.2. Хранение обучающего набора данных
- 12.2.3. Очистка и преобразование данных
- 12.2.4. Анализ настроений

12.3. Ранжирование мнений с помощью RNN
- 12.3.1. Выявление тем в комментариях
- 12.3.2. Анализ настроений с помощью алгоритмов глубокого обучения

12.4. Сеть кодирования-декодирования для нейронного машинного перевода
- 12.4.1. Обучение RNN для машинного перевода
- 12.4.2. Использование *кодирующей-декодирующей* сети для машинного перевода
- 12.4.3. Повышение точности машинного перевода с помощью RNN

12.5. Механизмы внимания
- 12.5.1. Реализация механизмов внимания в RNN
- 12.5.2. Использование механизмов внимания для повышения точности модели
- 12.5.3. Преимущества механизмов внимания в нейронных сетях

12.6. Модели *трансформеров*
- 12.6.1. Использование моделей *трансформеров* для обработки естественного языка
- 12.6.2. Применение моделей *трансформеров* для зрения
- 12.6.3. Преимущества моделей *трансформеров*

12.7. *Трансформеры* для зрения
- 12.7.1. Применение моделей *трансформеров* для зрения
- 12.7.2. Предварительная обработка данных изображений
- 12.7.3. Обучение модели *трансформеров* для зрения

12.8. Библиотека *трансформеров Hugging Face*
- 12.8.1. Использование библиотеки *трансформеров Hugging Face*
- 12.8.2. Применение библиотеки *трансформеров Hugging Face*
- 12.8.3. Преимущества библиотеки *трансформеров Hugging Face*

12.9. Другие библиотеки *трансформеров*. Сравнение
- 12.9.1. Сравнение различных библиотек *трансформеров*
- 12.9.2. Использование других библиотек *трансформеров*
- 12.9.3. Преимущества других библиотек *трансформеров*

12.10. Разработка NLP-приложения с использованием RNN и внимания. Практическое применение
- 12.10.1. Разработка приложения для обработки естественного языка с использованием RNN и внимания
- 12.10.2. Использование RNN, механизмов ухода и моделей *трансформеров* при внедрении
- 12.10.3. Оценка практического применения

Модуль 13. *Автоэнкодеры, GAN* и диффузионные модели

13.1. Эффективные представления данных
- 13.1.1. Снижение размерности
- 13.1.2. Глубокое обучение
- 13.1.3. Компактные представления

13.2. Реализация PCA с неполным линейным автоматическим кодировщиком
- 13.2.1. Процесс обучения
- 13.2.2. Внедрение Python
- 13.2.3. Использование тестовых данных

13.3. Стековые автоматические кодировщики
- 13.3.1. Глубокие нейронные сети
- 13.3.2. Построение архитектур кодирования
- 13.3.3. Использование инструментов

13.4. Конволюционные автокодировщики
- 13.4.1. Конструкция конволюционной модели
- 13.4.2. Обучение конволюционной модели
- 13.4.3. Оценка результатов

13.5. Шумоподавление автоматических энкодеров
- 13.5.1. Применение фильтров
- 13.5.2. Проектирование моделей кодирования
- 13.5.3. Использование методов регуляризации

13.6. Автоматические разреженные автоматические энкодеры
- 13.6.1. Повышение эффективности кодирования
- 13.6.2. Минимизация числа параметров
- 13.6.3. Применение методов регуляризации

13.7. Автоматические вариационные энкодеры
- 13.7.1. Использование вариационной оптимизации
- 13.7.2. Глубокое обучение без контроля
- 13.7.3. Глубокие латентные представления

13.8. Генерация модных изображений MNIST
- 13.8.1. Распознание паттернов
- 13.8.2. Генерация изображений
- 13.8.3. Обучение глубоких нейронных сетей

13.9. Генеративные адверсарные сети и диффузионные модели
- 13.9.1. Формирование контента из изображений
- 13.9.2. Моделирование распределений данных
- 13.9.3. Использование состязательных сетей

13.10. Реализация моделей
- 13.10.1. Практическое применение
- 13.10.2. Реализация моделей
- 13.10.3. Использование реальных данных
- 13.10.4. Оценка результатов

Модуль 14. Биоинспирированные вычисления

14.1. Введение в биоинспирированные вычисления
- 14.1.1. Введение в биоинспирированные вычисления

14.2. Алгоритмы социальной адаптации
- 14.2.1. Биоинспирированные алгоритмы, основанные на муравьиных колониях
- 14.2.2. Разновидности алгоритмов муравьиных колоний
- 14.2.3. Алгоритмы, основанные на облаках с частицами

14.3. Генетические алгоритмы
- 14.3.1. Общая структура
- 14.3.2. Внедрение основных операторов

14.4. Стратегии освоения и использования пространства для генетических алгоритмов
- 14.4.1. Алгоритм CHC
- 14.4.2. Мультимодальные задачи

14.5. Модели эволюционных вычислений (I)
- 14.5.1. Эволюционные стратегии
- 14.5.2. Эволюционное программирование
- 14.5.3. Алгоритмы, основанные на дифференциальной эволюции

14.6. Модели эволюционных вычислений (II)
- 14.6.1. Модели эволюции, основанные на оценке алгоритмов распределения (EDA)
- 14.6.2. Генетическое программирование

14.7. Применение эволюционного программирования при нарушениях обучаемости
- 14.7.1. Обучение на основе правил
- 14.7.2. Эволюционные методы в задачах выбора экземпляра

14.8. Многоцелевые задачи
- 14.8.1. Концепция доминирования
- 14.8.2. Применение эволюционных алгоритмов для решения многоцелевых задач

14.9. Нейронные сети (I)
- 14.9.1. Введение в нейронные сети
- 14.9.2. Практический пример с нейронными сетями

14.10. Нейронные сети (II)
- 14.10.1. Примеры использования нейронных сетей в медицинских исследованиях
- 14.10.2. Примеры использования нейронных сетей в экономике
- 14.10.3. Примеры использования нейронных сетей в искусственном зрении

Модуль 15. Искусственный интеллект: Стратегии и применение

15.1. Финансовые услуги
- 15.1.1. Последствия применения искусственного интеллекта (ИИ) в сфере финансовых услуг. Возможности и проблемы
- 15.1.2. Примеры использования
- 15.1.3. Потенциальные риски, связанные с использованием ИИ
- 15.1.4. Потенциальные будущие разработки/использования ИИ

15.2. Последствия применения искусственного интеллекта в здравоохранении
- 15.2.1. Последствия ИИ в секторе здравоохранения. Возможности и проблемы
- 15.2.2. Примеры использования

15.3. Риски, связанные с использованием ИИ в здравоохранении
- 15.3.1. Потенциальные риски, связанные с использованием ИИ
- 15.3.2. Потенциальные будущие разработки/использования ИИ

15.4. *Розничная торговля*
- 15.4.1. Последствия ИИ в *розничной торговле*. Возможности и проблемы
- 15.4.2. Примеры использования
- 15.4.3. Потенциальные риски, связанные с использованием ИИ
- 15.4.4. Потенциальные будущие разработки/использования ИИ

15.5. Промышленность
- 15.5.1. Последствия ИИ для промышленности. Возможности и проблемы
- 15.5.2. Примеры использования

15.6. Потенциальные риски, связанные с использованием ИИ в промышленности
- 15.6.1. Примеры использования
- 15.6.2. Потенциальные риски, связанные с использованием ИИ
- 15.6.3. Потенциальные будущие разработки/использования ИИ

15.7. Государственное управление
- 15.7.1. Последствия использования искусственного интеллекта в государственном управлении. Возможности и проблемы
- 15.7.2. Примеры использования
- 15.7.3. Потенциальные риски, связанные с использованием ИИ
- 15.7.4. Потенциальные будущие разработки/использования ИИ

15.8. Образовательная сфера
- 15.8.1. Последствия использования искусственного интеллекта в образовании. Возможности и проблемы
- 15.8.2. Примеры использования
- 15.8.3. Потенциальные риски, связанные с использованием ИИ
- 15.8.4. Потенциальные будущие разработки/использования ИИ

15.9. Лесное и сельское хозяйство
- 15.9.1. Последствия ИИ для лесного и сельского хозяйства. Возможности и проблемы
- 15.9.2. Примеры использования
- 15.9.3. Потенциальные риски, связанные с использованием ИИ
- 15.9.4. Потенциальные будущие разработки/использования ИИ

15.10 Человеческие ресурсы
- 15.10.1. Последствия ИИ для кадровых ресурсов. Возможности и проблемы
- 15.10.2. Примеры использования
- 15.10.3. Потенциальные риски, связанные с использованием ИИ
- 15.10.4. Потенциальные будущие разработки/использования ИИ

Модуль 16. Повышение производительности при разработке программного обеспечения с помощью ИИ

16.1. Подготовить подходящую среду разработки
- 16.1.1. Выбор необходимых инструментов для разработки ИИ
- 16.1.2. Конфигурация выбранных инструментов
- 16.1.3. Внедрение конвейеров CI/CD, адаптированных к проектам с использованием ИИ
- 16.1.4. Эффективное управление зависимостями и версиями в средах разработки

16.2. Обязательные расширения ИИ для Visual Studio Code
- 16.2.1. Изучение и выбор расширений ИИ для Visual Studio Code
- 16.2.2. Интеграция инструментов статического и динамического анализа в IDE
- 16.2.3. Автоматизация повторяющихся задач с помощью специальных расширений
- 16.2.4. Настройка среды разработки для повышения эффективности

16.3. Бескодовое проектирование пользовательских интерфейсов с элементами ИИ
- 16.3.1. Принципы *бескодового* проектирования и их применение в пользовательских интерфейсах
- 16.3.2. Включение элементов ИИ в визуальный дизайн интерфейса
- 16.3.3. Инструменты и платформы для создания интеллектуальных интерфейсов *без кода*
- 16.3.4. Оценка и постоянное совершенствование интерфейсов *без кода* с помощью ИИ

16.4. Оптимизация кода с помощью ChatGPT
- 16.4.1. Выявлять дублирующийся код
- 16.4.2. Рефакторить
- 16.4.3. Создавать читабельный код
- 16.4.4. Понимать, что делает код
- 16.4.5. Улучшение имен переменных и функций
- 16.4.6. Автоматическое создание документации

16.5. Управление репозиториями с помощью ИИ с использованием ChagGPT
- 16.5.1. Автоматизация процессов управления версиями с помощью методов ИИ
- 16.5.2. Обнаружение и автоматическое разрешение конфликтов в средах совместной работы
- 16.5.3. Прогнозируемый анализ изменений и тенденций в репозиториях кода
- 16.5.4. Улучшенная организация и категоризация репозиториев с помощью ИИ

16.6. Интеграция ИИ в управление базами данных с помощью AskYourDatabase
- 16.6.1. Оптимизация запросов и производительности с помощью методов ИИ
- 16.6.2. Предиктивный анализ шаблонов доступа к базам данных
- 16.6.3. Внедрение рекомендательных систем для оптимизации структуры баз данных
- 16.6.4. Проактивный мониторинг и обнаружение потенциальных проблем с базой данных

16.7. Поиск неисправностей и создание модульных тестов на основе ИИ с помощью ChatGPT
- 16.7.1. Автоматическая генерация тестовых примеров с использованием методов ИИ
- 16.7.2. Раннее обнаружение уязвимостей и ошибок с помощью статического анализа с помощью ИИ
- 16.7.3. Улучшение тестового покрытия путем выявления критических областей с помощью ИИ

16.8. *Парное программирование* с GitHub Copilot
- 16.8.1. Интеграция и эффективное использование GitHub Copilot в сессиях *парного программирования*
- 16.8.2. Интеграция и улучшение коммуникации и сотрудничества между разработчиками с помощью GitHub Copilot
- 16.8.3. Интеграция и стратегии использования подсказок по коду, генерируемых GitHub Copilot
- 16.8.4. Интеграция и примеры и лучшие практики *парного программирования* с помощью ИИ

16.9. Автоматический перевод между языками программирования с использованием ChatGPT
- 16.9.1. Инструменты и сервисы машинного перевода для конкретных языков программирования
- 16.9.2. Адаптация алгоритмов машинного перевода к контекстам разработки
- 16.9.3. Улучшение совместимости между различными языками с помощью машинного перевода
- 16.9.4. Оценка и устранение потенциальных проблем и ограничений в машинном переводе

16.10. Рекомендуемые инструменты ИИ для повышения производительности
- 16.10.1. Сравнительный анализ инструментов ИИ для разработки программного обеспечения
- 16.10.2. Интеграция инструментов ИИ в рабочие процессы
- 16.10.3. Автоматизация рутинных задач с помощью инструментов ИИ
- 16.10.4. Оценка и выбор инструментов в зависимости от контекста проекта и требований

Модуль 17. Архитектура программного обеспечения с помощью ИИ

17.1. Оптимизация и управление производительностью в инструментах с помощью ИИ с использованием ChatGPT
- 17.1.1. Анализ производительности и профилирование в инструментах искусственного интеллекта
- 17.1.2. Стратегии оптимизации алгоритмов и моделей ИИ
- 17.1.3. Внедрение *кэширования* и распараллеливания для повышения производительности
- 17.1.4. Инструменты и методологии для непрерывного мониторинга производительности в реальном времени

17.2. Масштабируемость в приложениях ИИ с использованием ChatGPT
- 17.2.1. Проектирование масштабируемых архитектур для приложений ИИ
- 17.2.2. Реализация методов разбиения на разделы и распределения нагрузки
- 17.2.3. Управление рабочими процессами и рабочими нагрузками в масштабируемых системах
- 17.2.4. Стратегии горизонтального и вертикального расширения в условиях переменного спроса

17.3. Обслуживаемость приложений ИИ с использованием ChatGPT
- 17.3.1. Принципы проектирования для обеспечения обслуживаемости в проектах ИИ
- 17.3.2. Стратегии документирования, характерные для моделей и алгоритмов ИИ
- 17.3.3. Реализация модульных и интеграционных тестов для облегчения обслуживаемости
- 17.3.4. Методы рефакторинга и непрерывного совершенствования систем с компонентами ИИ

17.4. Проектирование крупномасштабных систем
- 17.4.1. Архитектурные принципы проектирования крупномасштабных систем
- 17.4.2. Декомпозиция сложных систем на микросервисы
- 17.4.3. Реализация специфических паттернов проектирования распределенных систем
- 17.4.4. Стратегии управления сложностью в крупномасштабных архитектурах с компонентами ИИ

17.5. Крупномасштабные хранилища данных для инструментов ИИ
- 17.5.1. Выбор масштабируемых технологий хранения данных
- 17.5.2. Проектирование схем баз данных для эффективной работы с большими объемами данных
- 17.5.3. Стратегии разделения и репликации в средах хранения массовых данных
- 17.5.4. Внедрение систем управления данными для обеспечения целостности и доступности в проектах ИИ

17.6. Структуры данных ИИ с помощью ChatGPT
- 17.6.1. Адаптация классических структур данных для использования в алгоритмах ИИ
- 17.6.2. Проектирование и оптимизация специфических структур данных с помощью ChatGPT
- 17.6.3. Интеграция эффективных структур данных в системы с интенсивным использованием данных
- 17.6.4. Стратегии манипулирования и хранения данных в реальном времени в структурах данных ИИ

17.7. Алгоритмы программирования для продуктов ИИ
- 17.7.1. Разработка и реализация алгоритмов, специфичных для приложений ИИ
- 17.7.2. Стратегии выбора алгоритмов в зависимости от типа задачи и требований к продукту
- 17.7.3. Адаптация классических алгоритмов для интеграции в системы ИИ
- 17.7.4. Оценка и сравнение производительности различных алгоритмов в контексте разработки ИИ

17.8. Модели проектирования для разработки ИИ
- 17.8.1. Выявление и применение общих паттернов проектирования в проектах с компонентами ИИ
- 17.8.2. Разработка специальных паттернов для интеграции моделей и алгоритмов в существующие системы
- 17.8.3. Стратегии реализации паттернов для улучшения многократного использования и поддерживаемости в проектах ИИ
- 17.8.4. Кейс-стади и лучшие практики применения паттернов проектирования в архитектурах ИИ

17.9. Реализация чистой архитектуры с помощью ChatGPT
- 17.9.1. Основополагающие принципы и концепции *чистой архитектуры*
- 17.9.2. Адаптация *чистой архитектуры* к проектам с компонентами ИИ
- 17.9.3. Внедрение слоев и зависимостей в системах с чистой архитектурой
- 17.9.4. Преимущества и проблемы внедрения *чистой архитектуры* при разработке программного обеспечения для ИИ

17.10. Безопасная разработка программного обеспечения в веб-приложениях с помощью DeepCode
- 17.10.1. Принципы обеспечения безопасности при разработке программного обеспечения с использованием компонентов ИИ
- 17.10.2. Выявление и устранение потенциальных уязвимостей в моделях и алгоритмах ИИ
- 17.10.3. Внедрение практик безопасной разработки в веб-приложениях с функциями искусственного интеллекта
- 17.10.4. Стратегии защиты конфиденциальных данных и предотвращения атак в проектах с использованием ИИ

Модуль 18. Веб-проекты с помощью ИИ

18.1. Подготовка рабочей среды для веб-разработки с помощью ИИ
- 18.1.1. Настройка среды веб-разработки для проектов ИИ
- 18.1.2. Выбор и подготовка необходимых инструментов для веб-разработки ИИ
- 18.1.3. Интеграция специальных библиотек и *фреймворков* для веб-проектов ИИ
- 18.1.4. Внедрение лучших практик конфигурирования сред совместной разработки

18.2. Создание рабочего пространства для проектов ИИ с помощью GitHub Copilot
- 18.2.1. Эффективное проектирование и организация *рабочих мест*
- 18.2.2. Использование инструментов управления проектами и контроля версий в *рабочем пространстве*
- 18.2.3. Стратегии эффективного сотрудничества и коммуникации в команде разработчиков
- 18.2.4. Адаптация *рабочего пространства* к специфическим потребностям веб-проектов с помощью ИИ

18.3. Паттерны проектирования продуктов с помощью GitHub Copilot
- 18.3.1. Выявление и применение общих паттернов дизайна в пользовательских интерфейсах с элементами ИИ
- 18.3.2. Разработка специфических паттернов для улучшения пользовательского опыта в веб-проектах с помощью ИИ
- 18.3.3. Интеграция паттернов дизайна в общую архитектуру веб-проектов с помощью ИИ
- 18.3.4. Оценка и выбор подходящих паттернов проектирования в соответствии с контекстом проекта

18.4. Разработка фронтенда с помощью GitHub Copilot
- 18.4.1. Интеграция моделей ИИ в презентационный слой веб-проектов
- 18.4.2. Разработка адаптивных пользовательских интерфейсов с элементами ИИ
- 18.4.3. Реализация функциональных возможностей обработки естественного языка (NLP) во фронтенде
- 18.4.4. Стратегии оптимизации производительности при разработке фронтендов с поддержкой ИИ

18.5. Создание базы данных с помощью GitHub Copilot
- 18.5.1. Выбор технологий баз данных для веб-проектов с искусственным интеллектом
- 18.5.2. Разработка схем баз данных для хранения и управления данными, связанными с ИИ
- 18.5.3. Реализация эффективных систем хранения больших объемов данных, генерируемых моделями ИИ
- 18.5.4. Стратегии обеспечения безопасности и защиты конфиденциальных данных в базах данных веб-проектов ИИ

18.6. Разработка бэкенда с помощью GitHub Copilot
- 18.6.1. Интеграция сервисов и моделей ИИ в бизнес-логику бэкенда
- 18.6.2. Разработка специальных API и конечных точек для взаимодействия между фронтендом и компонентами ИИ
- 18.6.3. Реализация логики обработки данных и принятия решений в бэкенде с помощью ИИ
- 18.6.4. Стратегии масштабируемости и производительности при разработке Backend веб-проектов с помощью ИИ

18.7. Оптимизировать процесс развертывания вашего сайта
- 18.7.1. Автоматизация процесса создания и развертывания веб-проектов с помощью ChatGPT
- 18.7.2. Реализация конвейеров CI/CD, адаптированных к веб-приложениям с помощью Github Copilot
- 18.7.3. Стратегии эффективного управления релизами и обновлениями в непрерывных развертываниях
- 18.7.4. Мониторинг и анализ после развертывания для непрерывного улучшения процессов

18.8. ИИ в облачных вычислениях
- 18.8.1. Интеграция сервисов ИИ в платформы облачных вычислений
- 18.8.2. Разработка масштабируемых и распределенных решений с использованием облачных сервисов с поддержкой ИИ
- 18.8.3. Стратегии эффективного управления ресурсами и затратами в облачных средах с помощью веб-приложений с поддержкой ИИ
- 18.8.4. Оценка и сравнение поставщиков облачных услуг для веб-проектов с помощью искусственного интеллекта

18.9. Создание проекта с ИИ для сред LAMP с помощью ChatGPT
- 18.9.1. Адаптация веб-проектов на базе стека LAMP для включения компонентов искусственного интеллекта
- 18.9.2. Интеграция специфических для ИИ библиотек и *фреймворков* в среды LAMP
- 18.9.3. Разработка функциональных возможностей ИИ, дополняющих традиционную архитектуру LAMP
- 18.9.4. Стратегии оптимизации и сопровождения веб-проектов с ИИ в средах LAMP

18.10. Создание проекта с ИИ для MEVN-сред с использованием ChatGPT
- 18.10.1. Интеграция технологий и инструментов из стека MEVN с компонентами искусственного интеллекта
- 18.10.2. Разработка современных, масштабируемых веб-приложений в среде MEVN с возможностями ИИ
- 18.10.3. Реализация функций обработки данных и машинного обучения в проектах MEVN
- 18.10.4. Стратегии повышения производительности и безопасности веб-приложений с поддержкой ИИ в средах MEVN

Модуль 19. Мобильные приложения с помощью ИИ

19.1. Подготовка рабочей среды для мобильной разработки ИИ
- 19.1.1. Создание мобильных сред разработки для проектов ИИ
- 19.1.2. Выбор и подготовка специальных инструментов для разработки мобильных приложений с ИИ
- 19.1.3. Интеграция библиотек и *фреймворков* ИИ в мобильные среды разработки
- 19.1.4. Настройка эмуляторов и реальных устройств для тестирования мобильных приложений с компонентами искусственного интеллекта

19.2. Создание *рабочего пространства* с помощью GitHub Copilot
- 19.2.1. Интеграция GitHub Copilot в мобильные среды разработки
- 19.2.2. Эффективное использование GitHub Copilot для генерации кода в проектах ИИ
- 19.2.3. Стратегии совместной работы разработчиков при использовании GitHub Copilot в *рабочем пространстве*
- 19.2.4. Лучшие практики и ограничения использования GitHub Copilot при разработке мобильных приложений с ИИ

19.3. Конфигурация Firebase
- 19.3.1. Первоначальная настройка проекта Firebase для мобильной разработки
- 19.3.2. Интеграция Firebase в мобильные приложения с возможностями искусственного интеллекта
- 19.3.3. Использование сервисов Firebase, таких как база данных, аутентификация и уведомления, в проектах с искусственным интеллектом
- 19.3.4. Стратегии управления данными и событиями в реальном времени в мобильных приложениях с использованием Firebase

19.4. Концепции чистой архитектуры, источников данных, репозиториев
- 19.4.1. Фундаментальные принципы чистой архитектуры в мобильной разработке с использованием ИИ
- 19.4.2. Реализация слоев DataSources и Repositories с помощью GitHub Copilot
- 19.4.3. Проектирование и структурирование компонентов в мобильных проектах с помощью GitHub Copilot
- 19.4.4. Преимущества и проблемы реализации *чистой архитектуры* в мобильных приложениях с ИИ

19.5. Создание экрана аутентификации с помощью GitHub Copilot
- 19.5.1. Проектирование и разработка пользовательских интерфейсов для экранов аутентификации в мобильных приложениях с ИИ
- 19.5.2. Интеграция сервисов аутентификации Firebase в экран авторизации
- 19.5.3. Использование технологий безопасности и защиты данных в экране аутентификации
- 19.5.4. Персонализация и настройка пользовательского опыта на экране аутентификации

19.6. Создание приборной панели и навигация с помощью GitHub Copilot
- 19.6.1. Дизайн и разработка *приборной панели* с элементами искусственного интеллекта
- 19.6.2. Реализация эффективных систем навигации в мобильных приложениях с использованием ИИ
- 19.6.3. Интеграция функций ИИ в *приборную панель* для улучшения пользовательского опыта

19.7. Создание экрана листинга с помощью GitHub Copilot
- 19.7.1. Разработка пользовательских интерфейсов для экранов листинга в мобильных приложениях с ИИ
- 19.7.2. Интеграция алгоритмов рекомендации и фильтрации в экран листинга
- 19.7.3. Использование паттернов проектирования для эффективного представления данных в списке
- 19.7.4. Стратегии эффективной загрузки данных в режиме реального времени на экран листинга

19.8. Создание подробного экрана с помощью GitHub Copilot
- 19.8.1. Проектирование и разработка детальных пользовательских интерфейсов для представления конкретной информации
- 19.8.2. Интеграция функциональных возможностей искусственного интеллекта для обогащения экрана детализации
- 19.8.3. Реализация взаимодействий и анимации на экране детализации
- 19.8.4. Стратегии оптимизации производительности при отображении и загрузке деталей мобильного приложения с поддержкой ИИ

19.9. Создание экрана настроек с помощью GitHub Copilot
- 19.9.1. Разработка пользовательских интерфейсов для настройки и конфигурирования в мобильных приложениях с использованием ИИ
- 19.9.2. Интеграция пользовательских настроек, связанных с компонентами искусственного интеллекта
- 19.9.3. Реализация опций и предпочтений настройки на экране конфигурации
- 19.9.4. Стратегии удобства и ясности в представлении опций на экране *настроек*

19.10. Создание значков, *всплесков* и графических ресурсов для вашего приложения с помощью ИИ
- 19.10.1. Разработка и создание привлекательных иконок для представления мобильного приложения с помощью ИИ
- 19.10.2. Разработка заставки (*splash*) с впечатляющими визуальными эффектами
- 19.10.3. Выбор и адаптация графических ресурсов, улучшающих эстетику мобильного приложения
- 19.10.4. Стратегии согласованности и визуального брендинга в графических элементах приложений с помощью ИИ

Модуль 20. ИИ для *QA-тестирования*

20.1. Жизненный цикл *тестирования*
- 20.1.1. Описание и понимание жизненного цикла *тестирования* при разработке программного обеспечения
- 20.1.2. Фазы жизненного цикла *тестирования* и их значение для обеспечения качества
- 20.1.3. Интеграция искусственного интеллекта в различные этапы жизненного цикла *тестирования*
- 20.1.4. Стратегии непрерывного улучшения жизненного цикла *тестирования* с помощью искусственного интеллекта

20.2. Тестовые случаи и обнаружение багов с использованием ChatGPT
- 20.2.1. Эффективное проектирование и написание тестовых примеров в контексте *QA-тестирования*
- 20.2.2. Выявление ошибок и погрешностей во время выполнения тестовых примеров
- 20.2.3. Применение методов раннего обнаружения ошибок с помощью статического анализа
- 20.2.4. Использование инструментов искусственного интеллекта для автоматического выявления ошибок в тестовых примерах

20.3. Виды *тестирования*
- 20.3.1. Изучение различных видов *тестирования* в области QA
- 20.3.2. Модульное, интеграционное, функциональное и приемочное тестирование: характеристики и применение
- 20.3.3. Стратегии выбора и соответствующего сочетания типов тестирования в проектах с использованием ChatGPT
- 20.3.4. Адаптация традиционных типов тестирования к проектам с использованием ChatGPT

20.4. Создание плана тестирования с помощью ChatGPT
- 20.4.1. Разработка и структурирование комплексного плана тестирования
- 20.4.2. Определение требований и тестовых сценариев в проектах ИИ
- 20.4.3. Стратегии планирования ручного и автоматизированного тестирования
- 20.4.4. Оценка и постоянная корректировка плана тестирования в соответствии с развитием проекта

20.5. Обнаружение *ошибок* и составление отчетов с помощью ИИ
- 20.5.1. Реализация методов автоматического обнаружения ошибок с помощью алгоритмов машинного обучения
- 20.5.2. Использование ChatGPT для динамического анализа кода в поисках потенциальных ошибок
- 20.5.3. Стратегии автоматической генерации подробных отчетов о найденных ошибках с использованием ChatGPT
- 20.5.4. Эффективное взаимодействие между командами разработки и QA при работе с ошибками, выявленными искусственным интеллектом

20.6. Создание автоматизированного тестирования с помощью ИИ
- 20.6.1. Разработка сценариев автоматизированного тестирования для проектов с использованием ChatGPT
- 20.6.2. Интеграция средств автоматизации тестирования на основе ИИ
- 20.6.3. Использование алгоритмов ChatGPT для динамической генерации случаев автоматизированных тестов
- 20.6.4. Стратегии эффективного выполнения и сопровождения автоматизированных тестов в проектах ИИ

20.7. *Тестирование* API
- 20.7.1. Фундаментальные концепции *тестирования* API и его важность в QA
- 20.7.2. Разработка тестов для проверки API в средах с использованием ChatGPT
- 20.7.3. Стратегии валидации данных и результатов при тестировании API с использованием ChatGPT
- 20.7.4. Использование специальных инструментов для *тестирования* API в проектах с искусственным интеллектом

20.8. Инструменты ИИ для *веб-тестирования*
- 20.8.1. Изучение инструментов искусственного интеллекта для автоматизации тестирования в веб-средах
- 20.8.2. Интеграция технологий распознавания элементов и визуальной аналитики в *веб-тестирование*
- 20.8.3. Стратегии автоматического обнаружения изменений и проблем с производительностью в веб-приложениях с использованием ChatGPT
- 20.8.4. Оценка конкретных инструментов для повышения эффективности веб-тестирования с помощью ИИ

20.9. *Мобильное тестирование* с помощью ИИ
- 20.9.1. Разработка стратегий *тестирования* мобильных приложений с компонентами искусственного интеллекта
- 20.9.2. Интеграция специальных инструментов *тестирования* на основе ИИ для мобильных платформ
- 20.9.3. Использование ChatGPT для обнаружения проблем с производительностью в мобильных приложениях
- 20.9.4. Стратегии валидации специфических интерфейсов и функций мобильных приложений с использованием ИИ

20.10. Инструменты контроля качества с помощью ИИ
- 20.10.1. Изучение инструментов и платформ QA, включающих функциональность ИИ
- 20.10.2. Оценка инструментов для эффективного управления тестированием и выполнения тестов в проектах ИИ
- 20.10.3. Использование ChatGPT для создания и оптимизации тестовых заданий
- 20.10.4. Стратегии эффективного выбора и внедрения QA-инструментов с поддержкой ИИ

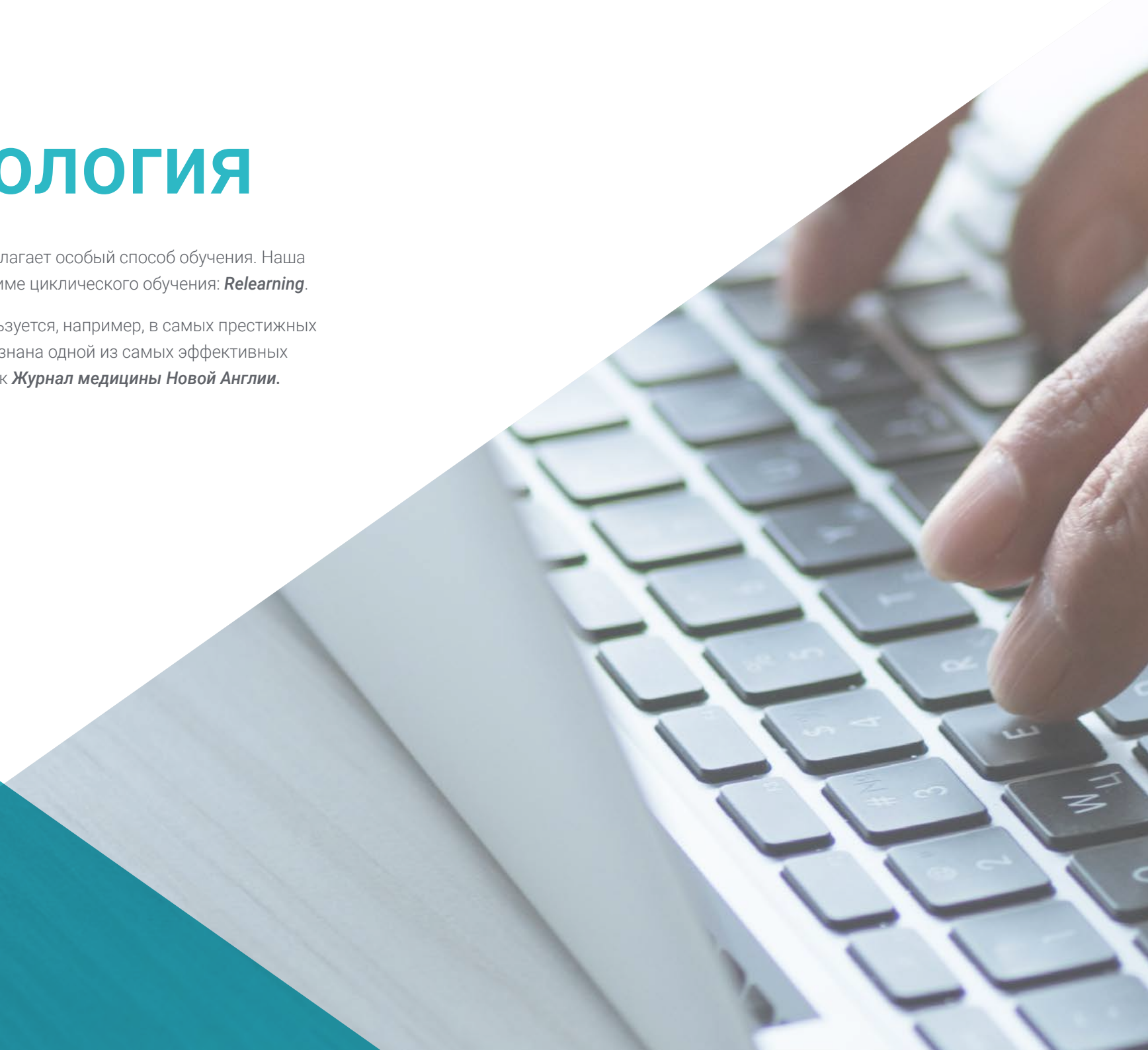
Позиционируйте себя на рынке труда с помощью 100% онлайн-программы, которая адаптируется к вашим потребностям и позволяет погрузиться в процесс обучения"

06

Методология

Данная учебная программа предлагает особый способ обучения. Наша методология разработана в режиме циклического обучения: ***Relearning***.

Данная система обучения используется, например, в самых престижных медицинских школах мира и признана одной из самых эффективных ведущими изданиями, такими как ***Журнал медицины Новой Англии.***



Откройте для себя методику Relearning, которая отвергает традиционное линейное обучение, чтобы показать вам циклические системы обучения: способ, который доказал свою огромную эффективность, особенно в предметах, требующих запоминания"

Исследование кейсов для контекстуализации всего содержания

Наша программа предлагает революционный метод развития навыков и знаний. Наша цель - укрепить компетенции в условиях меняющейся среды, конкуренции и высоких требований.

С TECH вы сможете познакомиться со способом обучения, который опровергает основы традиционных методов образования в университетах по всему миру"

Вы получите доступ к системе обучения, основанной на повторении, с естественным и прогрессивным обучением по всему учебному плану.

В ходе совместной деятельности и рассмотрения реальных кейсов студент научится разрешать сложные ситуации в реальной бизнес-среде.

Инновационный и отличный от других метод обучения

Эта программа TECH - интенсивная программа обучения, созданная с нуля, которая предлагает самые сложные задачи и решения в этой области на международном уровне. Благодаря этой методологии ускоряется личностный и профессиональный рост, делая решающий шаг на пути к успеху. Метод кейсов, составляющий основу данного содержания, обеспечивает следование самым современным экономическим, социальным и профессиональным реалиям.

> *Наша программа готовит вас к решению новых задач в условиях неопределенности и достижению успеха в карьере"*

Кейс-метод является наиболее широко используемой системой обучения лучшими преподавателями в мире. Разработанный в 1912 году для того, чтобы студенты-юристы могли изучать право не только на основе теоретического содержания, метод кейсов заключается в том, что им представляются реальные сложные ситуации для принятия обоснованных решений и ценностных суждений о том, как их разрешить. В 1924 году он был установлен в качестве стандартного метода обучения в Гарвардском университете.

Что должен делать профессионал в определенной ситуации? Именно с этим вопросом мы сталкиваемся при использовании кейс-метода - метода обучения, ориентированного на действие. На протяжении всей курса студенты будут сталкиваться с многочисленными реальными случаями из жизни. Им придется интегрировать все свои знания, исследовать, аргументировать и защищать свои идеи и решения.

Методология *Relearning*

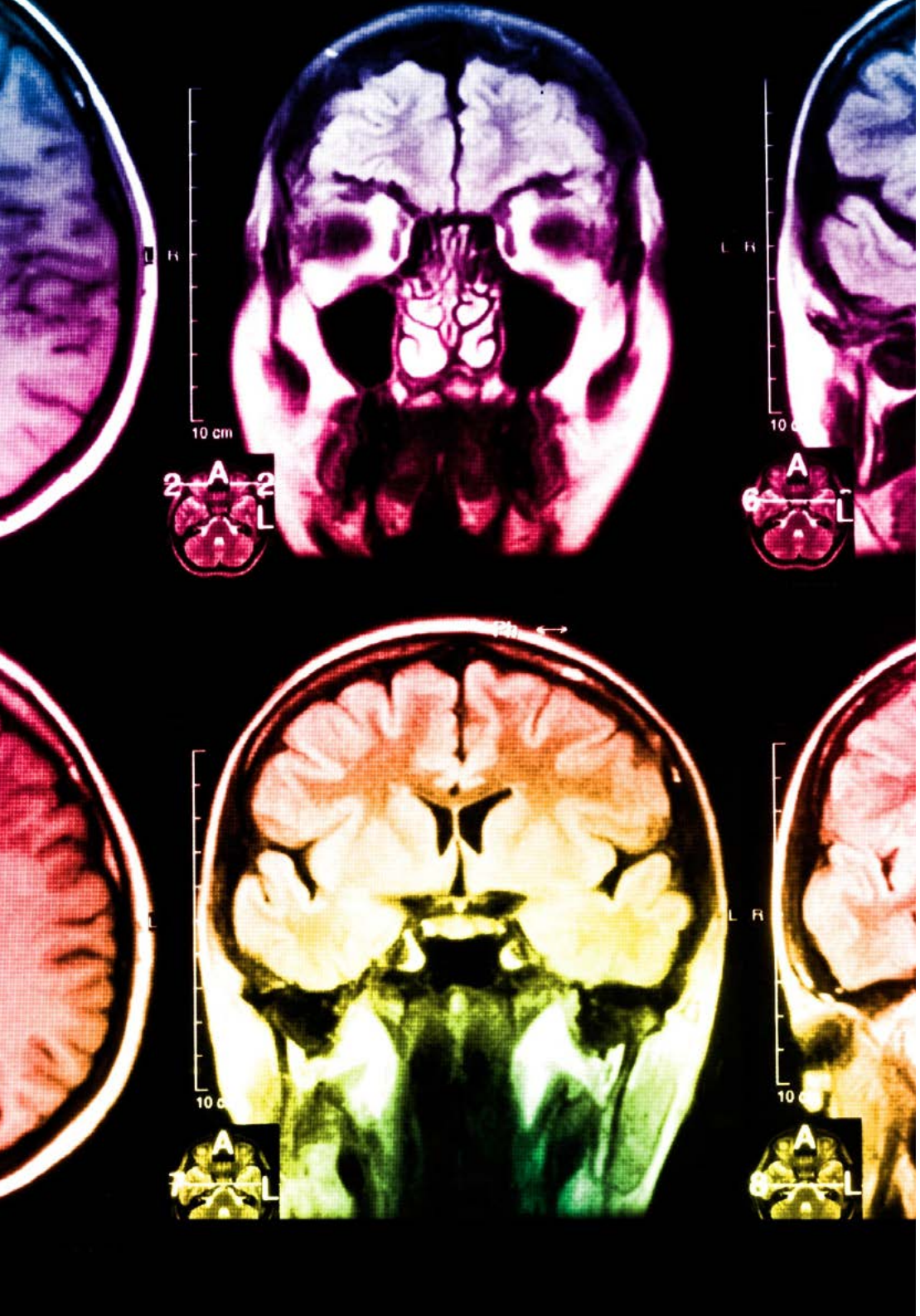
TECH эффективно объединяет метод кейсов с системой 100% онлайн-обучения, основанной на повторении, которая сочетает различные дидактические элементы в каждом уроке.

Мы улучшаем метод кейсов с помощью лучшего метода 100% онлайн-обучения: *Relearning*.

В 2019 году мы достигли лучших результатов обучения среди всех онлайн-университетов в мире.

В TECH вы будете учиться по передовой методике, разработанной для подготовки руководителей будущего. Этот метод, играющий ведущую роль в мировой педагогике, называется *Relearning*.

Наш университет - единственный вуз, имеющий лицензию на использование этого успешного метода. В 2019 году нам удалось повысить общий уровень удовлетворенности наших студентов (качество преподавания, качество материалов, структура курса, цели...) по отношению к показателям лучшего онлайн-университета.



В нашей программе обучение не является линейным процессом, а происходит по спирали (мы учимся, разучиваемся, забываем и заново учимся). Поэтому мы дополняем каждый из этих элементов по концентрическому принципу. Благодаря этой методике более 650 000 выпускников университетов добились беспрецедентного успеха в таких разных областях, как биохимия, генетика, хирургия, международное право, управленческие навыки, спортивная наука, философия, право, инженерное дело, журналистика, история, финансовые рынки и инструменты. Наша методология преподавания разработана в среде с высокими требованиями к уровню подготовки, с университетским контингентом студентов с высоким социально-экономическим уровнем и средним возрастом 43,5 года.

Методика Relearning позволит вам учиться с меньшими усилиями и большей эффективностью, все больше вовлекая вас в процесс обучения, развивая критическое мышление, отстаивая аргументы и противопоставляя мнения, что непосредственно приведет к успеху.

Согласно последним научным данным в области нейронауки, мы не только знаем, как организовать информацию, идеи, образы и воспоминания, но и знаем, что место и контекст, в котором мы что-то узнали, имеют фундаментальное значение для нашей способности запомнить это и сохранить в гиппокампе, чтобы удержать в долгосрочной памяти.

Таким образом, в рамках так называемого нейрокогнитивного контекстно-зависимого электронного обучения, различные элементы нашей программы связаны с контекстом, в котором участник развивает свою профессиональную практику.

В рамках этой программы вы получаете доступ к лучшим учебным материалам, подготовленным специально для вас:

Учебный материал

Все дидактические материалы создаются преподавателями специально для студентов этого курса, чтобы они были действительно четко сформулированными и полезными.

Затем вся информация переводится в аудиовизуальный формат, создавая дистанционный рабочий метод TECH. Все это осуществляется с применением новейших технологий, обеспечивающих высокое качество каждого из представленных материалов.

Мастер-классы

Существуют научные данные о пользе экспертного наблюдения третьей стороны.

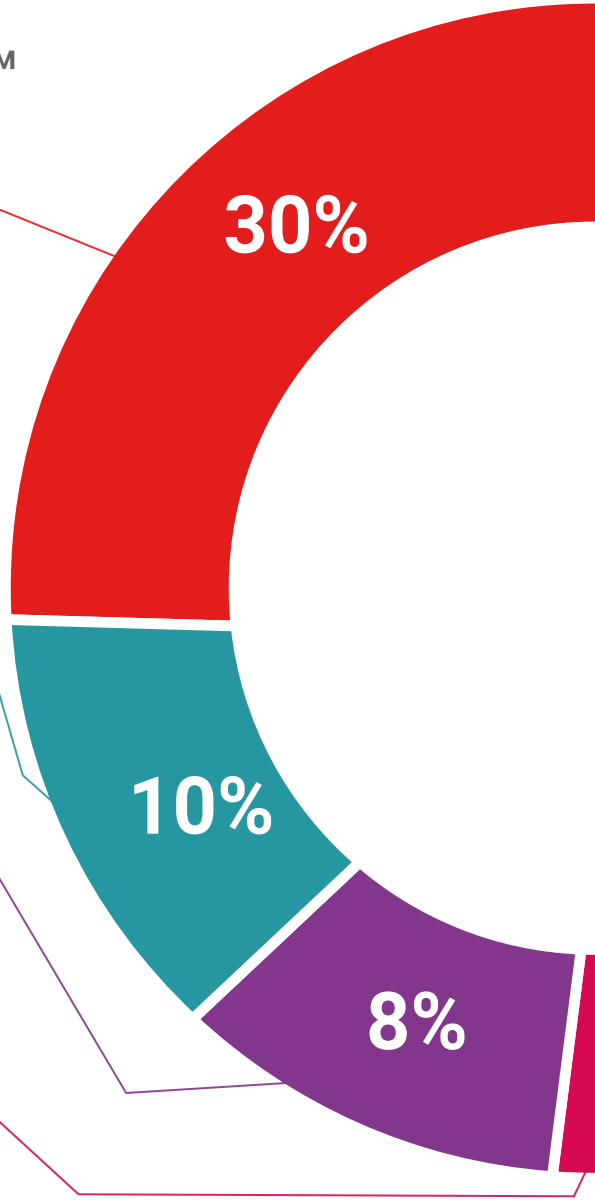
Так называемый метод обучения у эксперта укрепляет знания и память, а также формирует уверенность в наших будущих сложных решениях.

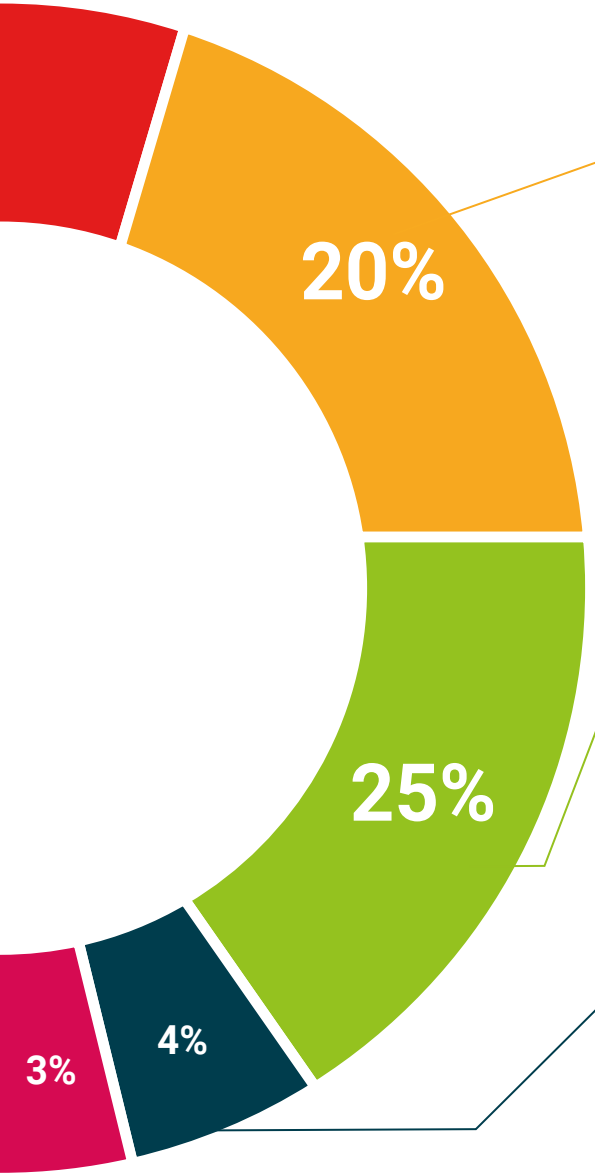
Практика навыков и компетенций

Студенты будут осуществлять деятельность по развитию конкретных компетенций и навыков в каждой предметной области. Практика и динамика приобретения и развития навыков и способностей, необходимых специалисту в рамках глобализации, в которой мы живем.

Дополнительная литература

Новейшие статьи, консенсусные документы и международные руководства включены в список литературы курса. В виртуальной библиотеке TECH студент будет иметь доступ ко всем материалам, необходимым для завершения обучения.





Метод кейсов

Метод дополнится подборкой лучших кейсов, выбранных специально для этой квалификации. Кейсы представляются, анализируются и преподаются лучшими специалистами на международной арене.

Интерактивные конспекты

Мы представляем содержание в привлекательной и динамичной мультимедийной форме, которая включает аудио, видео, изображения, диаграммы и концептуальные карты для закрепления знаний.

Эта уникальная обучающая система для представления мультимедийного содержания была отмечена компанией Microsoft как "Европейская история успеха".

Тестирование и повторное тестирование

На протяжении всей программы мы периодически оцениваем и переоцениваем ваши знания с помощью оценочных и самооценочных упражнений: так вы сможете убедиться, что достигаете поставленных целей.

07

Квалификация

Специализированная магистратура в области искусственного интеллекта в программировании гарантирует, помимо самого строгого и современного обучения, получение диплома об окончании Специализированной магистратуры, выдаваемого TECH Технологическим университетом.

Успешно пройдите эту программу и получите университетский диплом без хлопот, связанных с поездками и бумажной волокитой”

Данная **Специализированная магистратура в области искусственного интеллекта в программировании** содержит самую полную и современную программу на рынке.

После прохождения аттестации студент получит по почте* с подтверждением получения соответствующий диплом **Специализированной магистратуры,** выданный **TECH Технологическим университетом.**

Диплом, выданный **TECH Технологическим университетом**, подтверждает квалификацию, полученную в Специализированной магистратуре, и соответствует требованиям, обычно предъявляемым биржами труда, конкурсными экзаменами и комитетами по оценке карьеры.

Диплом: **Специализированная магистратура в области искусственного интеллекта в программировании**

Формат: **онлайн**

Продолжительность: **12 месяцев**

tech технологический университет

Настоящий

ДИПЛОМ

свидетельствует о том, что

Г-н/Г-жа _____ с документом, удостоверяющий личность_____

Освоил(а) и успешно прошел(ла) аккредитацию программы

СПЕЦИАЛИЗИРОВАННАЯ МАГИСТРАТУРА

в области

Искусственного интеллекта в программировании

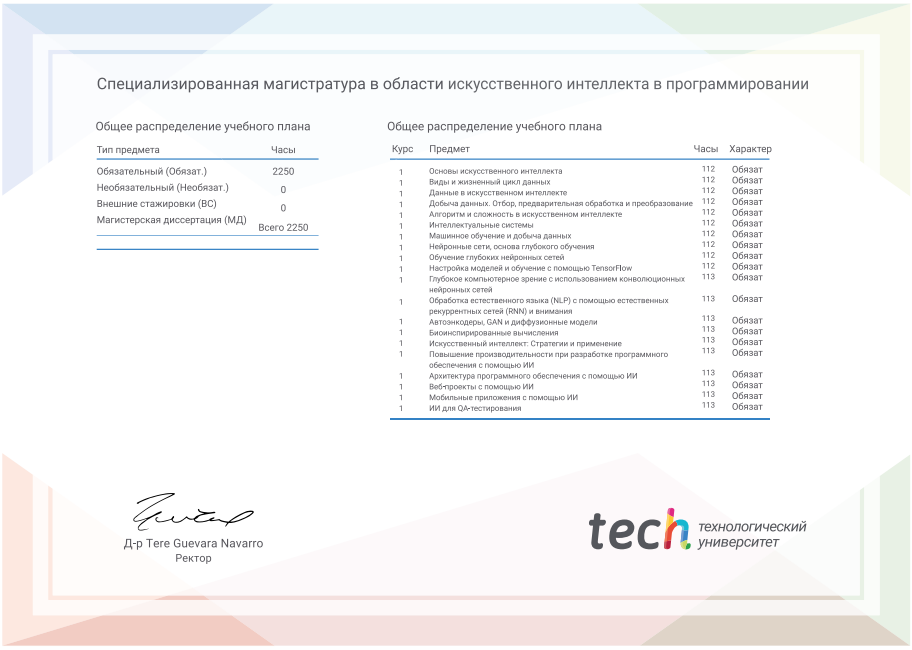
Данный диплом специализированной программы, присуждаемый Университетом, соответствует 2250 учебным часам, с датой начала дд/мм/гггг и датой окончания дд/мм/гггг.

TECH является частным высшим учебным заведением, признанным Министерством народного образования Мексики с 28 июня 2018 года.

17 июня 2020 г.

Д-р Tere Guevara Navarro
Ректор

unique TECH code: AFWOR235 techtitute.com/certificates



Специализированная магистратура в области искусственного интеллекта в программировании

Общее распределение учебного плана

Тип предмета	Часы
Обязательный (Обязат.)	2250
Необязательный (Необязат.)	0
Внешние стажировки (ВС)	0
Магистерская диссертация (МД)	Всего 2250

Общее распределение учебного плана

Курс	Предмет	Часы	Характер
1	Основы искусственного интеллекта	112	Обязат
1	Виды и жизненный цикл данных	112	Обязат
1	Данные в искусственном интеллекте	112	Обязат
1	Добыча данных. Отбор, предварительная обработка и преобразование	112	Обязат
1	Алгоритм и сложность в искусственном интеллекте	112	Обязат
1	Интеллектуальные системы	112	Обязат
1	Машинное обучение и добыча данных	112	Обязат
1	Нейронные сети, основа глубокого обучения	112	Обязат
1	Обучение глубоких нейронных сетей	112	Обязат
1	Настройка моделей и обучение с помощью TensorFlow	112	Обязат
1	Глубокое компьютерное зрение с использованием конволюционных нейронных сетей	113	Обязат
1	Обработка естественного языка (NLP) с помощью естественных рекуррентных сетей (RNN) и внимания	113	Обязат
1	Автоэнкодеры, GAN и диффузионные модели	113	Обязат
1	Биоинспирированные вычисления	113	Обязат
1	Искусственный интеллект: Стратегии и применение	113	Обязат
1	Повышение производительности при разработке программного обеспечения с помощью ИИ	113	Обязат
1	Архитектура программного обеспечения с помощью ИИ	113	Обязат
1	Веб-проекты с помощью ИИ	113	Обязат
1	Мобильные приложения с помощью ИИ	113	Обязат
1	ИИ для QA-тестирования	113	Обязат

Д-р Tere Guevara Navarro
Ректор

tech технологический университет

*Гаагский апостиль. В случае, если студент потребует, чтобы на его диплом в бумажном формате был проставлен Гаагский апостиль, TECH EDUCATION предпримет необходимые шаги для его получения за дополнительную плату.

tech технологический университет

Специализированная магистратура
Искусственный интеллект в программировании

- Формат: **онлайн**
- Продолжительность: **12 месяцев**
- Учебное заведение: **TECH Технологический университет**
- Расписание: **по своему усмотрению**
- Экзамены: **онлайн**

Специализированная магистратура

Искусственный интеллект в программировании

tech технологический университет