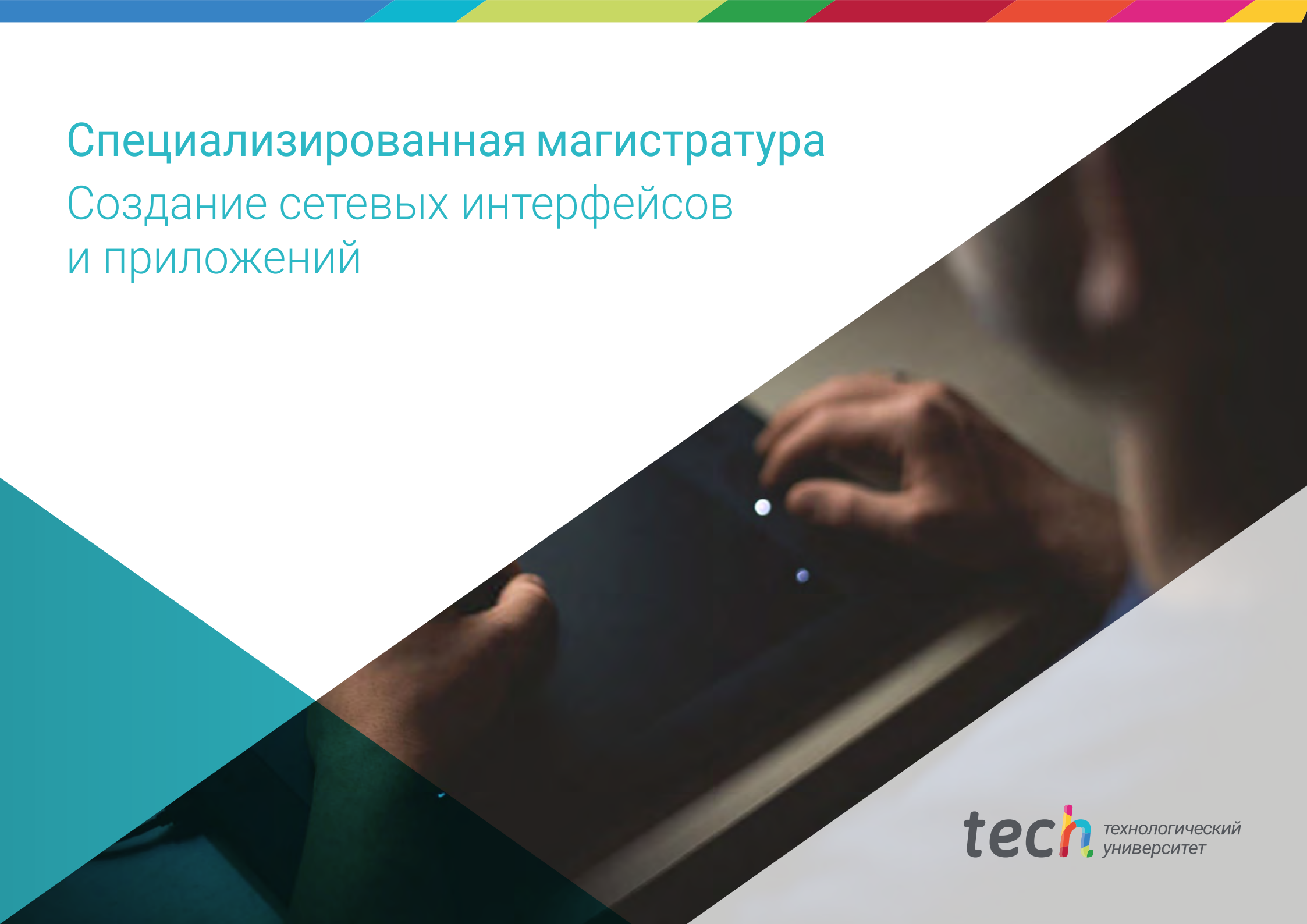
Специализированная магистратура

Создание сетевых интерфейсов и приложений



tech технологический университет

tech технологический университет

Специализированная магистратура

Создание сетевых интерфейсов и приложений

- Формат: **онлайн**
- Продолжительность: **12 месяцев**
- Учебное заведение: **TECH Технологический университет**
- Режим обучения: **16ч./неделя**
- Расписание: **по своему усмотрению**
- Экзамены: **онлайн**

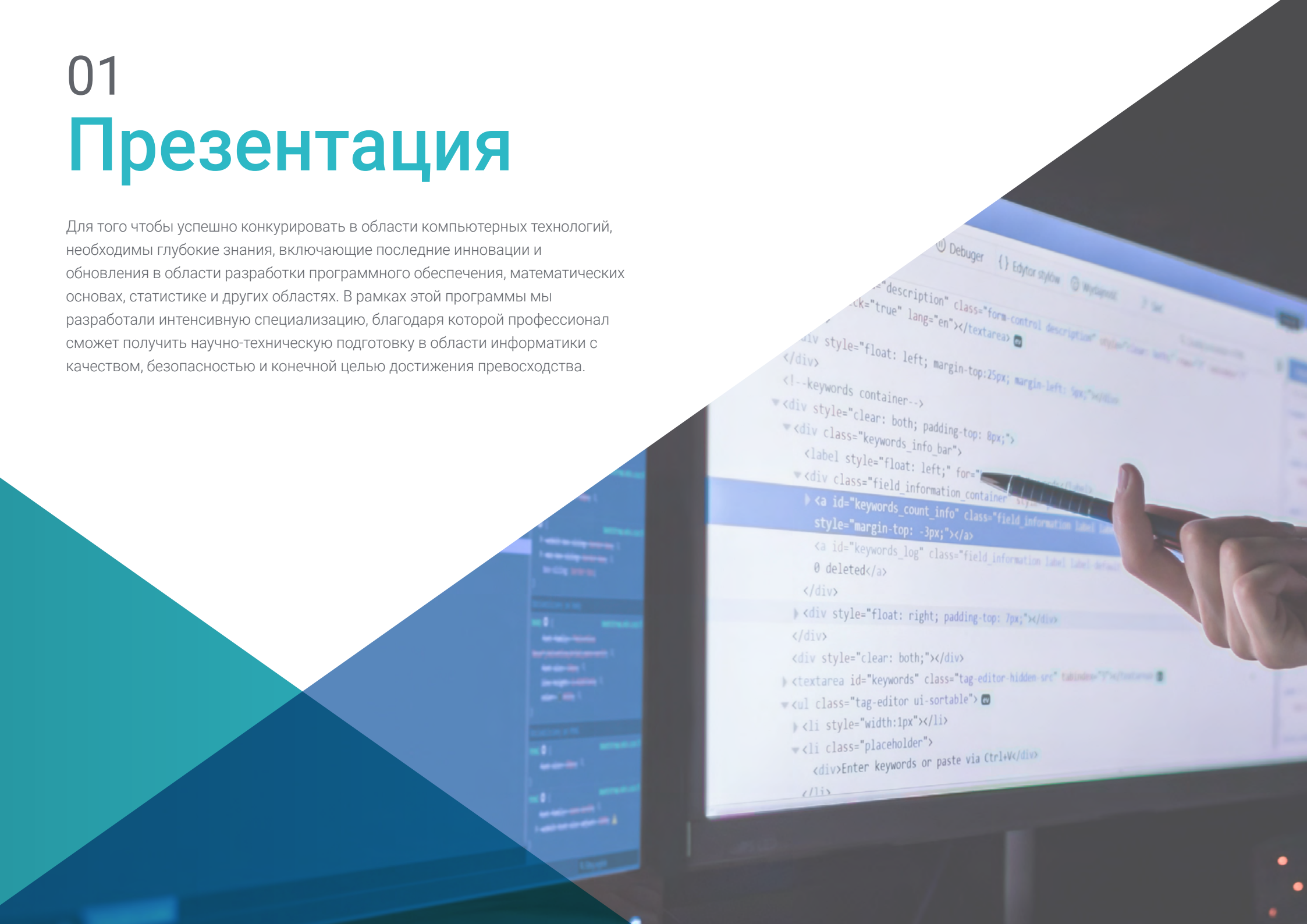
Веб-доступ: www.techtitute.com/ru/information-technology/professional-master-degree/master-creation-interfaces-network-applications

Оглавление

01

Презентация

Для того чтобы успешно конкурировать в области компьютерных технологий, необходимы глубокие знания, включающие последние инновации и обновления в области разработки программного обеспечения, математических основах, статистике и других областях. В рамках этой программы мы разработали интенсивную специализацию, благодаря которой профессионал сможет получить научно-техническую подготовку в области информатики с качеством, безопасностью и конечной целью достижения превосходства.



Наша инновационная концепция телепрактики даст вам возможность учиться в режиме погружения, что обеспечит более быструю интеграцию и гораздо более реалистичное представление о содержании: "Обучение у эксперта"

Данное обучение развивает необходимые концепции для работы в области создания интерфейсов, обеспечивая специалиста полным овладением всех пересекающихся областей знаний по этой теме. В рамках программы предлагаются инновационные дидактические подходы, чтобы углубленно ознакомиться с архитектурой распределенного приложения. Будут рассмотрены особенности клиент-серверной архитектуры, а также основы и необходимые разработки наиболее распространенных языков программирования, их дифференциация, и многие другие фундаментальные вопросы для профессионала.

Эти неотъемлемые знания являются первым шагом к развитию способности к разработке данного типа технологии.

В рамках этого обучения предлагается реальное представление о работе, чтобы оценить целесообразность ее применения в проекте, оценить реальные указания, методы разработки и ожидаемые результаты.

Через опыт достигается понимание, как развивать необходимые знания для продвижения в этой рабочей сфере. Это обучение, которое обязательно требует опыта, в данной специализации объединяет дистанционное обучение и практическое обучение, предлагая уникальную возможность придать вашему резюме тот толчок, который вы ищете.

Станьте одним из самых востребованных профессионалов на сегодняшний день: обучайтесь в области информатики с помощью самой полной и актуальной Специализированной магистратуры в области создания сетевых интерфейсов и приложений”

Данная **Специализированная магистратура в области создания сетевых интерфейсов и приложений** содержит самую полную и современную образовательную программу на рынке. Основными особенностями обучения являются:

- Новейшие технологии в области программного обеспечения для электронного обучения
- Абсолютно наглядная система обучения, подкрепленная графическим и схематическим содержанием, которое легко усвоить и понять
- Разработка практических кейсов, представленных практикующими экспертами
- Современные интерактивные видеосистемы
- Дистанционное преподавание
- Постоянное обновление и переработка знаний
- Саморегулируемое обучение: абсолютная совместимость с другими обязанностями
- Практические упражнения для самооценки и проверки знаний
- Группы поддержки и образовательная совместная деятельность: вопросы эксперту, дискуссии и форумы знаний
- Общение с преподавателем и индивидуальная работа по ассимиляции полученных знаний
- Учебные материалы курса доступны с любого стационарного или мобильного устройства с выходом в интернет
- Постоянный доступ к дополнительным материалам во время и после окончания программы

Благодаря методологии преподавания, основанной на проверенных методиках, данная инновационная Специализированная магистратура в области создания сетевых интерфейсов и приложений проведет вас через различные подходы к обучению, что придаст учебному процессу динамичность и эффективность”

Наш преподавательский состав состоит из специалистов из различных областей, связанных с этой специальностью. Таким образом, TECH гарантирует, что вы достигнете той цели обновления знаний, к которой стремитесь. Одним из отличительных качеств этой программы является многопрофильная команда профессионалов, с образованием и опытом работы в различных сферах, которые преподают теоретические знания, основываясь на собственном опыте.

Все эти знания дополнены эффективной методологией преподавания. Программа разработана многопрофильной командой экспертов в области *электронного обучения* и объединяет в себе последние достижения в области образовательных технологий. Таким образом, вы сможете учиться с помощью ряда удобных и универсальных мультимедийных инструментов, которые обеспечат вам необходимую оперативность в обучении.

В основе этой программы лежит проблемно-ориентированное обучение: подход, который рассматривает обучение как исключительно практический процесс. Для эффективности дистанционного обучения мы используем телепрактику: с помощью инновационной интерактивной видеосистемы и системы Learning from an Expert вы сможете получить знания в таком же объеме, как если бы вы обучались, непосредственно присутствуя на занятиях. Концепция, которая позволит вам интегрировать и закрепить обучение более реалистичным и постоянным способом.

Специализированная магистратура позволит вам работать во всех областях создания сетевых интерфейсов и приложений с уверенностью профессионала высокого уровня.

С помощью опыта действующих профессионалов, которые дадут вам реальные, непосредственные и конкретные знания в этой области работы.

02

Цели

Целью TECH является предоставление актуальной информации профессионалу в области создания сетевых интерфейсов и приложений. Это позволит системному инженеру приобрести новые навыки для создания более полного программного обеспечения. Эта цель может быть достигнута всего за несколько месяцев благодаря учебному плану, ориентированному на текущие потребности отрасли.

Расширьте свои знания в области компьютерных наук и разработки программного обеспечения и подготовьтесь конкурировать с лучшими в отрасли”

Общие цели

- Получить научно-техническую подготовку для работы в области компьютерной инженерии
- Получить широкие знания в области компьютерных наук
- Получить широкие знания в области структуры компьютеров
- Приобрести необходимые знания в области разработки программного обеспечения
- Ознакомиться с основами математики, статистики и физики, необходимыми для этой области

Данная программа дает возможность обучения и профессионального роста и позволит вам повысить конкурентоспособность на рынке труда"

Конкретные цели

Модуль 1. Взаимодействие человека и компьютера

- Получить прочные знания в области взаимодействия человека и компьютера и создания удобных интерфейсов
- Понять важность удобства использования приложений и почему важно учитывать его при разработке нашего программного обеспечения
- Понимать различные типы человеческого разнообразия, ограничения, которые они влекут за собой, и как адаптировать интерфейсы в соответствии с конкретными потребностями каждого из них
- Изучить процесс проектирования интерфейса, от анализа требований до оценки, через различные промежуточные этапы, необходимые для реализации подходящего интерфейса
- Знать различные руководящие принципы доступности, стандарты, которые их устанавливают, и инструменты, позволяющие их оценить
- Понимать различные методы взаимодействия с компьютером, используя периферийные устройства и приспособления

Модуль 2. Базы данных

- Изучать различные области применения и назначения систем баз данных, а также их функционирование и архитектуру
- Понимать реляционную модель, от ее структуры и операций до расширенной реляционной алгебры
- Глубоко изучать, что такое базы данных SQL, как они работают, как определять данные и как создавать запросы от самых простых до самых продвинутых и сложных
- Научиться проектировать базы данных с использованием модели "сущность-отношение", создавать диаграммы и узнавать о характеристиках расширенной модели E-R
- Углубиться в проектирование реляционных баз данных, анализируя различные нормальные формы и алгоритмы декомпозиции
- Заложить основы для понимания функционирования баз данных NoSQL, а также познакомить с базой данных Mongo DB

Модуль 3. Разработка сетевых приложений

- Знать особенности языка разметки HTML и его использование при создании веб-сайтов вместе с таблицами стилей CSS
- Знать, как использовать браузерно-ориентированный язык программирования JavaScript и некоторые его основные возможности
- Понимать концепции компонентно-ориентированного программирования и компонентной архитектуры
- Знать, как использовать *Framework* для *Front-End* Bootstrap для создания веб-сайтов
- Понимать структуру модели представления контроллера при разработке динамических веб-сайтов
- Знать сервисно-ориентированную архитектуру и основы протокола HTTP

Модуль 4. Свободное программное обеспечение и открытые знания

- Изучать концепции свободного программного обеспечения и открытых знаний, а также различные типы соответствующих лицензий
- Знать об основных бесплатных инструментах, доступных в различных областях, таких как операционные системы, управление бизнесом, менеджеры контента и создание мультимедийного контента и т.д.
- Понимать важность и преимущества свободного программного обеспечения в деловом мире, как с точки зрения его характеристик, так и стоимости
- Получить углубленные знания об операционной системе GNU/Linux, а также о различных существующих дистрибутивах и о том, как их можно настроить
- Знать о функционировании и развитии WordPress, учитывая, что на эту CMS приходится более 35% активных веб-сайтов в мире, и более 60% в конкретном случае с CMS
- Понимать, как работает операционная система Android для мобильных устройств, а также основы разработки мобильных приложений как нативных, так и с помощью кроссплатформенных *фреймворков*

Модуль 5. Расширенные базы данных

- Представить различные системы баз данных, доступные в настоящее время на рынке
- Изучить использование XML и баз данных для работы в Интернете
- Понимать работу расширенных баз данных, таких как параллельные и распределенные базы данных
- Понимать важность индексирования и объединения в системах баз данных
- Понимать функционирование систем транзакционной обработки и поиска информации
- Приобрести знания, связанные с нереляционными базами данных и добычей данных

Модуль 6. Программная инженерия

- Ознакомиться с основными принципами программной инженерии и стандартом ISO/IEC 12207
- Понимать особенности объединенного процесса разработки программного обеспечения и планирования в контексте гибкой разработки программного обеспечения
- Изучать различные стили проектирования распределенного программного обеспечения и сервис-ориентированных программных архитектур
- Освоить основные концепции проектирования графического пользовательского интерфейса
- Понимать основы разработки веб-приложений
- Углубиться в изучение стратегий и техник тестирования программного обеспечения, факторов качества программного обеспечения и различных используемых метрик

Модуль 7. Продвинутое программирование

- Получать углубленные знания в области программирования, особенно в отношении объектно-ориентированного программирования и различных видов отношений между существующими классами
- Знать различные шаблоны проектирования для решения объектно-ориентированных задач
- Изучать событийно-управляемое программирование и разработку пользовательского интерфейса с помощью Qt
- Получать необходимые знания о параллельном программировании, процессах и потоках
- Знать, как управлять использованием потоков и синхронизацией, а также как решать общие задачи в параллельном программировании
- Понимать важность документации и тестирования при разработке программного обеспечения

Модуль 8. Повторное использование программного обеспечения

- Понимать общую картину стратегии повторного использования программного обеспечения
- Освоить различные шаблоны, связанные с повторным использованием программного обеспечения, включая шаблоны проектирования, создания, структурные и поведенческие
- Познакомиться с понятием *фреймворка* и изучить основные типы, включая фреймворки для проектирования графических пользовательских интерфейсов, разработки веб-приложений и управления постоянством объектов в базах данных
- Понимать принцип работы широко используемого сейчас паттерна Модель-Представление-Контроллер (Model-View-Controller, MVC)

Модуль 9. Искусственный интеллект и инженерия знаний

- Установить основы искусственного интеллекта и инженерии знаний, сделав краткий обзор истории искусственного интеллекта до наших дней
- Понимать основные концепции поиска в искусственном интеллекте, как информированного, так и неинформированного
- Понимать принцип работы искусственного интеллекта в играх
- Освоить основные концепции нейронных сетей и использование генетических алгоритмов
- Освоить необходимые механизмы для представления знаний, особенно с учетом семантической паутины
- Понимать принцип работы экспертных систем и систем поддержки принятия решения

Модуль 10. Продвинутая программная инженерия

- Изучать в деталях различные методологии гибкой разработки программного обеспечения
- Освоить техники разработки с использованием методик Scrum, экстремального программирования и разработки программного обеспечения на основе повторного использования
- Понимать различные шаблоны архитектуры систем и проектирования программного обеспечения, а также архитектуру облачных приложений
- Освоить методики тестирования программного обеспечения, такие как *разработка программного обеспечения через тестирование*, *разработка через приемочное тестирование, поведенческое тестирование и* использование инструмента *Cucumber*
- Углубиться в улучшение процесса разработки программного обеспечения и качества программного обеспечения с использованием стандартов ISO/IEC
- Ввести понятие DevOps и основные практики этой методологии

03

Компетенции

Данная программа в области создания сетевых интерфейсов и приложений была разработана в качестве мощного инструмента для профессиональной подготовки. Ее интенсивный учебный план внесет значительный вклад в развитие и работу программистов и веб-порталов, используя имеющиеся ресурсы Интернета и свободное программное обеспечение, с уклоном на практическую и полезную ориентацию.

С помощью полученных навыков благодаря данной Специализированной магистратуре вы сможете начать разрабатывать приложения и работать в качестве инженера по разработке программного обеспечения, обладая полной и актуальной специализацией”

Общий профессиональный навык

- Приобрести необходимые навыки для профессиональной практики в области компьютерной инженерии со знанием всех необходимых факторов, чтобы выполнять ее качественно и полноценно

Возможность, созданная для профессионалов, ищущих интенсивную и эффективную программу, чтобы сделать значительный шаг вперед в своей профессии"

Профессиональные навыки

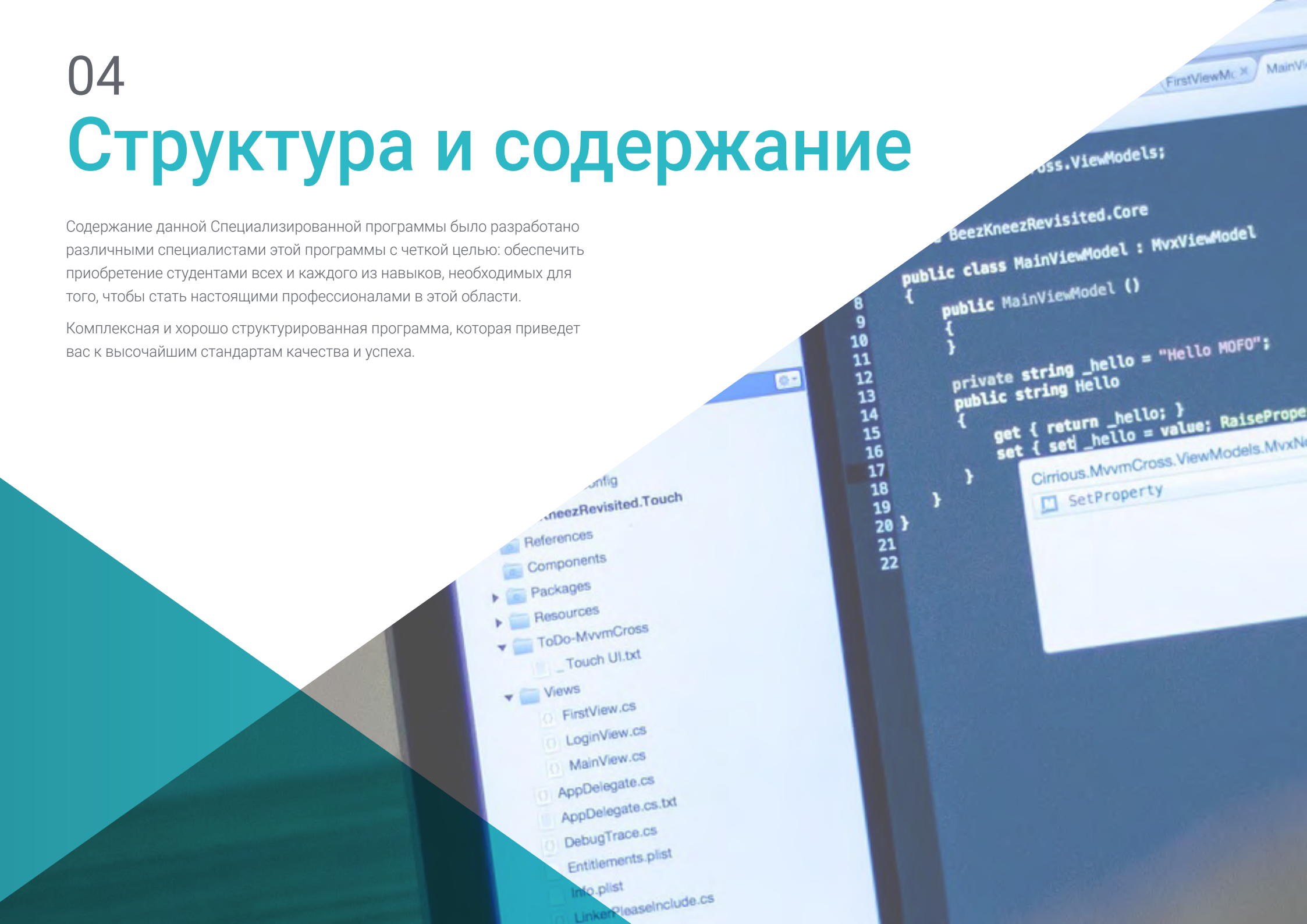
- Получить глубокое понимание всех аспектов взаимодействия человека с компьютером и как они включают в себя разработки информационных технологий
- Быть компетентным в использовании баз данных
- Разрабатывать различные типы приложений в сети
- Описывать и использовать свободное программное обеспечение и открытые знания, существующие в сети
- Работать в качестве инженера программного обеспечения
- Контролировать использование расширенных баз данных
- Выполнять продвинутое программирование
- Знать, как повторно использовать программное обеспечение
- Создавать интерфейсы и приложения в сети
- Овладеть различными методами работы в продвинутой разработке программного обеспечения

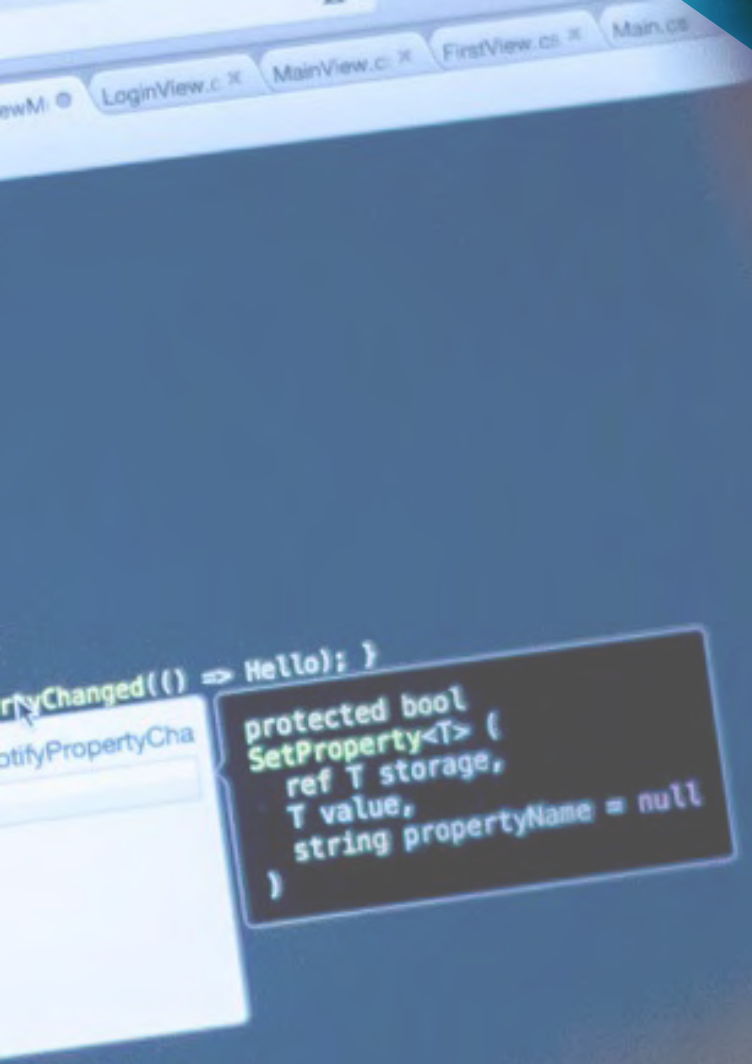
04

Структура и содержание

Содержание данной Специализированной программы было разработано различными специалистами этой программы с четкой целью: обеспечить приобретение студентами всех и каждого из навыков, необходимых для того, чтобы стать настоящими профессионалами в этой области.

Комплексная и хорошо структурированная программа, которая приведет вас к высочайшим стандартам качества и успеха.





Полноценная программа обучения, структурированная в полные дидактические единицы и состоящая из обновленных данных на основе последних достижений в отрасли, ориентированная на обучение, совместимое с вашей личной и профессиональной жизнью"

Модуль 1. Взаимодействие человека и компьютера

1.1. Введение во взаимодействие человека и компьютера
- 1.1.1. Что такое взаимодействие человека и компьютера?
- 1.1.2. Взаимодействия человека и компьютера с другими дисциплинами
- 1.1.3. Пользовательский интерфейс
- 1.1.4. Удобство использования и доступность
- 1.1.5. Опыт пользователей и дизайна, ориентированного на пользователя

1.2. Компьютер и взаимодействие: пользовательский интерфейс и парадигмы взаимодействия
- 1.2.1. Взаимодействие
- 1.2.2. Парадигмы и стили взаимодействия
- 1.2.3. Эволюция пользовательских интерфейсов
- 1.2.4. Классические пользовательские интерфейсы: WIMP/GUI, команды, голос, виртуальная реальность
- 1.2.5. Инновационные пользовательские интерфейсы: мобильные, носимые, совместные, НКИ

1.3. Человеческий фактор: психологические и когнитивные аспекты
- 1.3.1. Важность человеческого фактора на основе взаимодействия
- 1.3.2. Обработка информации человеком
- 1.3.3. Ввод и вывод информации: визуальной, слуховой и тактильной
- 1.3.4. Восприятие и внимание
- 1.3.5. Знания и ментальные модели: представление, организация и приобретение

1.4. Человеческий фактор: сенсорные и физические ограничения
- 1.4.1. Функциональное разнообразие, инвалидность и ограничения жизнедеятельности
- 1.4.2. Визуальное разнообразие
- 1.4.3. Слуховое разнообразие
- 1.4.4. Когнитивное разнообразие
- 1.4.5. Моторно-функциональное разнообразие
- 1.4.6. Случай цифровых мигрантов

1.5. Процесс проектирования (I): анализ требований для проектирования пользовательского интерфейса
  1.5.1. Дизайн, ориентированный на пользователя
  1.5.2. Что такое анализ требований
  1.5.3. Сбор информации
  1.5.4. Анализ и интерпретация информации
  1.5.5. Анализ юзабилити и доступность
1.6. Процесс проектирования (II): создание прототипов и анализ задач
  1.6.1. Концептуальное проектирование
  1.6.2. Создание прототипов
  1.6.3. Иерархический анализ задач
1.7. Процесс проектирования (III): оценка
  1.7.1. Оценка в процессе проектирования: цели и методы
  1.7.2. Методы оценки без участия пользователей
  1.7.3. Методы оценки с участием пользователей
  1.7.4. Стандарты и нормы оценки
1.8. Доступность: определение и рекомендации
  1.8.1. Доступность и универсальный дизайн
  1.8.2. Инициатива WAI и руководство WCAG
  1.8.3. Руководство WCAG 2.0 и 2.1
1.9. Доступность: оценка и функциональное разнообразие
  1.9.1. Инструменты оценки доступности веб-сайтов
  1.9.2. Доступность и функциональное разнообразие
1.10. Компьютер и взаимодействие: периферийные устройства и приспособления
  1.10.1. Традиционные периферийные устройства
  1.10.2. Альтернативные периферийные устройства
  1.10.3. Мобильные телефоны и планшеты
  1.10.4. Функциональное разнообразие, взаимодействие и периферийные устройства

Модуль 2. Базы данных

2.1. Применение и назначение систем баз данных
  2.1.1. Применение различных систем баз данных
  2.1.2. Назначение в различных системах баз данных
  2.1.3. Обзор данных
2.2. База данных и архитектура
  2.2.1. Реляционные базы данных
  2.2.2. Проектирование базы данных
  2.2.3. Объектно-ориентированные и полуструктурированные базы данных
  2.2.4. Хранение и запросы данных
  2.2.5. Управление транзакциями
  2.2.6. Добыча и анализ данных
  2.2.7. Архитектура базы данных
2.3. Реляционная модель: структура, операции и расширенная реляционная алгебра
  2.3.1. Структура реляционных баз данных
  2.3.2. Фундаментальные операции в реляционной алгебре
  2.3.3. Прочие операции реляционной алгебры
  2.3.4. Расширенные операции реляционной алгебры
  2.3.5. Нулевые значения
  2.3.6. Внесение изменений в базу данных
2.4. SQL (I)
  2.4.1. Что такое (SQL)?
  2.4.2. Определение данных
  2.4.3. Базовая структура SQL-запросов
  2.4.4. Операции над множествами
  2.4.5. Агрегатные функции
  2.4.6. Нулевые значения
2.5. SQL (II)
  2.5.1. Вложенные запросы
  2.5.2. Сложные запросы
  2.5.3. Просмотры
  2.5.4. Курсоры
  2.5.5. Сложные запросы
  2.5.6. Триггеры

2.6. Проектирование баз данных и ER модель
- 2.6.1. Обзор процесса проектирования
- 2.6.2. Модель сущности — отношения
- 2.6.3. Ограничения

2.7. Диаграммы сущность — связь
- 2.7.1. Диаграммы сущность — связь
- 2.7.2. Свойства проектирования сущность — отношение
- 2.7.3. Наборы слабых сущностей

2.8. Расширенная модель сущность — отношение
- 2.8.1. Характеристики расширенной ER-модели
- 2.8.2. Проектирование базы данных
- 2.8.3. Сокращение до реляционных схем

2.9. Разработка реляционной базы данных
- 2.9.1. Характеристики хороших реляционных проектов
- 2.9.2. Атомные домены и первая нормальная форма (1НФ)
- 2.9.3. Декомпозиция через функциональные зависимости
- 2.9.4. Теория функциональной зависимости
- 2.9.5. Алгоритмы декомпозиции
- 2.9.6. Декомпозиция через многозначные зависимости
- 2.9.7. Другие нормальные формы
- 2.9.8. Процесс разработки базы данных

2.10. Базы данных NoSQL
- 2.10.1. Что представляют собой базы данных NoSQL?
- 2.10.2. Анализ различных вариантов NoSQL и их особенностей
- 2.10.3. Mongo DB

Модуль 3. Разработка сетевых приложений

3.1. Языки разметки HTML5
- 3.1.1. Основные концепции HTML
- 3.1.2. Новые элементы HTML 5
- 3.1.3. Формы: новые элементы управления

3.2. Введение в таблицы стилей CSS
- 3.2.1. Первые шаги с CSS
- 3.2.2. Введение в CSS3

3.3. Браузерные скрипты: JavaScript
- 3.3.1. Основные концепции JavaScript
- 3.3.2. DOM
- 3.3.3. События
- 3.3.4. JQuery
- 3.3.5. Ajax

3.4. Концепция компонентно-ориентированного программирования
- 3.4.1. Контекст
- 3.4.2. Компоненты и интерфейсы
- 3.4.3. Состояния компонента

3.5. Архитектура компонентов
- 3.5.1. Текущие архитектуры
- 3.5.2. Интеграция и развертывание компонентов

3.6. *Framework Front-End: Bootstrap*
- 3.6.1. Проектирование сетки
- 3.6.2. Формуляры
- 3.6.3. Компоненты

3.7. Контроллер представления модели
- 3.7.1. Методы веб-разработки
- 3.7.2. Шаблон проектирования: MVC

3.8. Информационные *грид*-технологии
- 3.8.1. Увеличение вычислительных ресурсов
- 3.8.2. Концепция *грид*-технологии

3.9. Сервис-ориентированная архитектура
- 3.9.1. SOA и веб-сервисы
- 3.9.2. Топология веб-сервиса
- 3.9.3. Платформы для веб-сервисов

3.10. Протокол HTTP
- 3.10.1. Сообщения
- 3.10.2. Постоянные сеансы
- 3.10.3. Криптографическая система
- 3.10.4. Принцип работы протокола HTTPS

Модуль 4. Свободное программное обеспечение и открытые знания

4.1. Введение в свободное программное обеспечение
- 4.1.1. История свободного программного обеспечения
- 4.1.2. "Свобода" в программном обеспечении
- 4.1.3. Лицензии на использование программных средств
- 4.1.4. Интеллектуальная собственность на программное обеспечение
- 4.1.5. Какова мотивация для использования свободного программного обеспечения?
- 4.1.6. Мифы о свободном программном обеспечении
- 4.1.7. Топ-500

4.2. Открытые знания и лицензии CC
- 4.2.1. Основные понятия
- 4.2.2. Лицензии *Creative Commons*
- 4.2.3. Другие лицензии на контент
- 4.2.4. Википедия и другие открытые проекты знаний

4.3. Основные инструменты свободного программного обеспечения
- 4.3.1. Операционные системы
- 4.3.2. Офисные приложения
- 4.3.3. Приложения для управления бизнесом
- 4.3.4. Менеджеры веб-контента
- 4.3.5. Инструменты для создания мультимедийного контента
- 4.3.6. Другие приложения

4.4. Компания: свободное программное обеспечение и его стоимость
- 4.4.1. Свободное программное обеспечение: да или нет?
- 4.4.2. Правда и ложь о свободном программном обеспечении
- 4.4.3. Бизнес-программы, основанные на свободном программном обеспечении
- 4.4.4. Стоимость программного обеспечения
- 4.4.5. Модели свободного программного обеспечения

4.5. Операционная система GNU/ Linux
- 4.5.1. Архитектура
- 4.5.2. Основная структура каталогов
- 4.5.3. Характеристики и структура файловой системы
- 4.5.4. Внутреннее представление файлов

4.6. Мобильная операционная система Android
- 4.6.1. История
- 4.6.2. Архитектура
- 4.6.3. Форки Android
- 4.6.4. Введение в разработку Android
- 4.6.5. *Фреймворки* для разработки мобильных приложений

4.7. Создание веб-сайтов с помощью WordPress
- 4.7.1. Особенности и структура WordPress
- 4.7.2. Создание сайтов wordpress.com
- 4.7.3. Установка и настройка WodPress на собственном сервере
- 4.7.4. Установка плагинов и расширение WordPress
- 4.7.5. Создание плагинов WordPress
- 4.7.6. Создание тем WordPress

4.8. Тенденции развития свободного программного обеспечения
- 4.8.1. Облачные среды
- 4.8.2. Инструменты мониторинга
- 4.8.3. Операционные системы
- 4.8.4. Большие данные и открытые данные 2.0
- 4.8.5. Квантовые вычисления

4.9. Контроль версий
- 4.9.1. Основные понятия
- 4.9.2. Git
- 4.9.3. Облачные и самостоятельные сервисы Git
- 4.9.4. Другие системы контроля версий

4.10. Пользовательские дистрибутивы GNU/Linux
- 4.10.1. Основные дистрибутивы
- 4.10.2. Дистрибутивы, производные от Debian
- 4.10.3. Создание deb-пакетов
- 4.10.4. Изменение дистрибутива
- 4.10.5. Генерация ISO-образов

Модуль 5. Расширенные базы данных

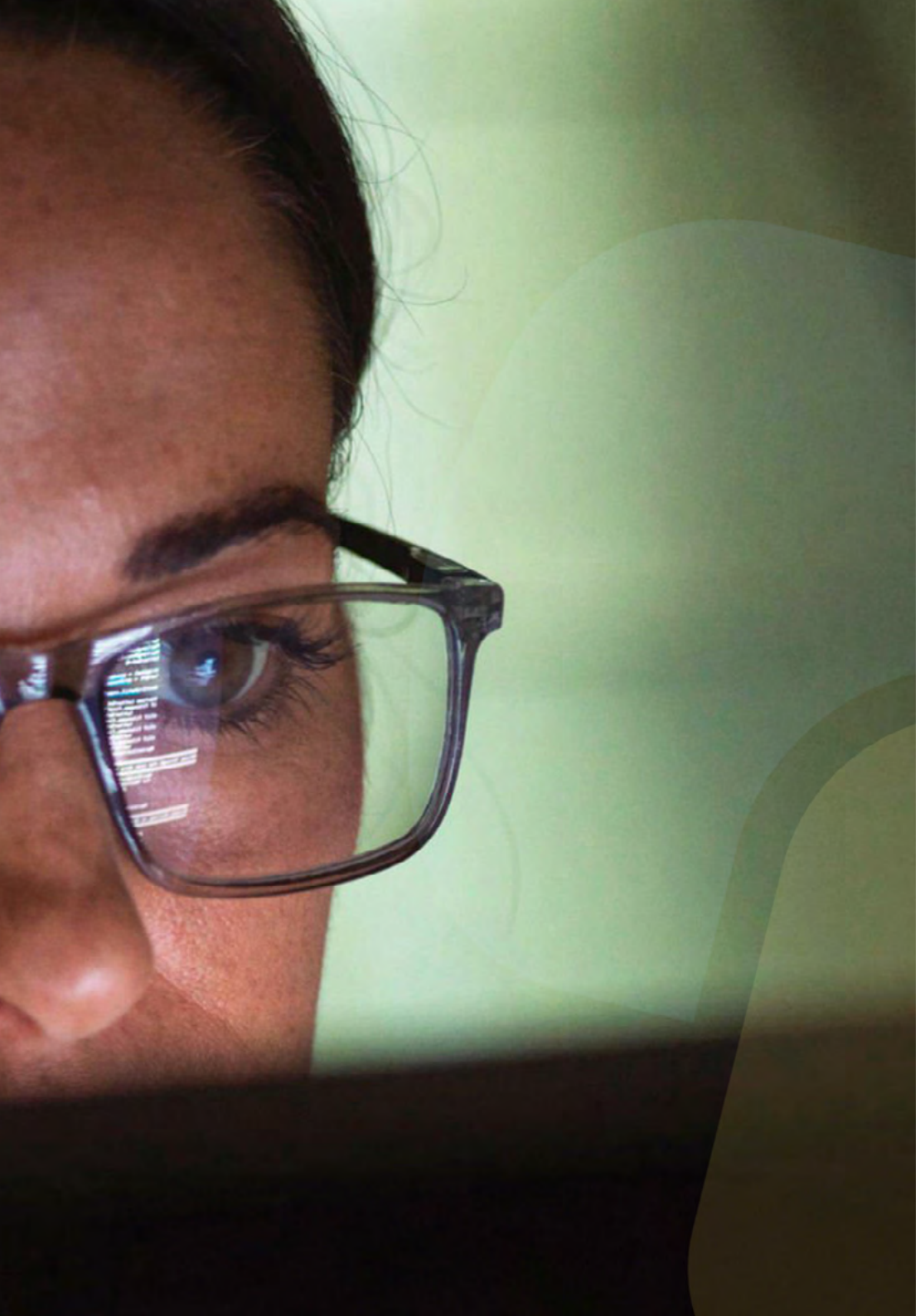
5.1. Введение в различные системы баз данных
- 5.1.1. Исторический обзор
- 5.1.2. Иерархические базы данных
- 5.1.3. Сетевые базы данных
- 5.1.4. Реляционные базы данных
- 5.1.5. Нереляционные базы данных

5.2. XML и базы данных для веб-сайта
- 5.2.1. Валидация XML-файлов
- 5.2.2. Преобразования XML-файлов
- 5.2.3. Хранение данных в XML
- 5.2.4. Реляционные базы данных XML
- 5.2.5. SQL/XML
- 5.2.6. Собственные базы данных XML

5.3. Параллельные базы данных
- 5.3.1. Параллельные системы
- 5.3.2. Параллельные архитектуры баз данных
- 5.3.3. Параллелизм в запросах
- 5.3.4. Параллелизм при запросах
- 5.3.5. Проектирование параллельных систем
- 5.3.6. Параллельная обработка в SQL

5.4. Распределенные базы данных
- 5.4.1. Распределенные системы
- 5.4.2. Распределенное хранение
- 5.4.3. Доступность
- 5.4.4. Распределенная обработка запросов
- 5.4.5. Провайдеры распределенных баз данных

5.5. Индексация и ассоциация
- 5.5.1. Упорядоченные индексы
- 5.5.2. Разреженные и плотные индексы
- 5.5.3. Многоуровневые индексы
- 5.5.4. Обновление индекса
- 5.5.5. Статическая ассоциация
- 5.5.6. Как применять индексы в базах данных



5.6. Введение в транзакционную обработку
- 5.6.1. Стадии транзакции
- 5.6.2. Внедрение атомарности и долговечности
- 5.6.3. Последовательность
- 5.6.4. Восстановление
- 5.6.5. Внедрение изоляции

5.7. Системы восстановления
- 5.7.1. Классификация ошибок
- 5.7.2. Структуры хранения
- 5.7.3. Восстановление и атомарность
- 5.7.4. Восстановление на основе исторических данных
- 5.7.5. Параллельные транзакции и восстановление
- 5.7.6. Высокая доступность в базах данных

5.8. Выполнение и обработка запросов
- 5.8.1. Стоимость консультации
- 5.8.2. Работа по выбору
- 5.8.3. Ординация
- 5.8.4. Введение в оптимизацию запросов
- 5.8.5. Мониторинг эффективности

5.9. Нереляционные базы данных
- 5.9.1. Документо-ориентированные базы данных
- 5.9.2. Графовые базы данных
- 5.9.3. Базы данных с ключами-значениями

5.10. *Хранилище* данных, OLAP и добыча данных
- 5.10.1. Компоненты хранилищ данных
- 5.10.2. Архитектура хранилища данных
- 5.10.3. OLAP
- 5.10.4. Возможности добычи данных
- 5.10.5. Другие виды добычи данных

Модуль 6. Программная инженерия

6.1. Основы программной инженерии
- 6.1.1. Характеристики программного обеспечения
- 6.1.2. Основные процессы в программной инженерии
- 6.1.3. Модели процессов разработки программного обеспечения
- 6.1.4. Стандартная справочная система для процесса разработки программного обеспечения: стандарт ISO/IEC 12207

6.2. Унифицированный процесс разработки программного обеспечения
- 6.2.1. Унифицированный процесс
- 6.2.2. Размеры унифицированного процесса
- 6.2.3. Процесс разработки на основе сценариев использования
- 6.2.4. Фундаментальные рабочие процессы унифицированных процессов

6.3. Планирование в контексте гибкой разработки программного обеспечения
- 6.3.1. Характеристики гибкой разработки программного обеспечения
- 6.3.2. Различные временные горизонты планирования в гибкой разработке
- 6.3.3. Схема гибкой разработки Scrum и временные горизонты планирования
- 6.3.4. Пользовательские истории как единица планирования и оценки
- 6.3.5. Общие методы получения сметы
- 6.3.6. Шкалы для интерпретации оценок
- 6.3.7. *Покер планирования*
- 6.3.8. Общие типы планирования: планирование поставок и планирование итераций

6.4. Стили проектирования распределенного программного обеспечения и сервис-ориентированные программные архитектуры
- 6.4.1. Модели коммуникации в распределенных программных системах
- 6.4.2. Слой промежуточного программного обеспечения
- 6.4.3. Архитектурные паттерны для распределенных систем
- 6.4.4. Общий процесс проектирования программных сервисов
- 6.4.5. Аспекты проектирования программных сервисов
- 6.4.6. Состав услуги
- 6.4.7. Архитектура веб-сервисов
- 6.4.8. Инфраструктура и компоненты SOA

6.5. Введение в разработку программного обеспечения на основе моделей
- 6.5.1. Концепция модели
- 6.5.2. Разработка программного обеспечения, управляемого моделями
- 6.5.3. Система разработки, управляемой моделями MDA
- 6.5.4. Элементы модели трансформации

6.6. Проектирование графического интерфейса пользователя
- 6.6.1. Принципы проектирования пользовательского интерфейса
- 6.6.2. Архитектурные паттерны проектирования интерактивных систем: Модель-Представление-Контроллер (MVC)
- 6.6.3. Пользовательский опыт (UX *User Experience*)
- 6.6.4. Дизайн, ориентированный на пользователя
- 6.6.5. Анализ и процесс проектирования графического пользовательского интерфейса
- 6.6.6. Юзабилити пользовательских интерфейсов
- 6.6.7. Доступность пользовательских интерфейсов

6.7. Дизайн веб-приложений
- 6.7.1. Характеристики веб-приложений
- 6.7.2. Пользовательский интерфейс веб-приложения
- 6.7.3. Проектирование навигации
- 6.7.4. Основной протокол взаимодействия для веб-приложений
- 6.7.5. Архитектурные стили для веб-приложений

6.8. Стратегии и методы тестирования программного обеспечения и факторы качества программного обеспечения
- 6.8.1. Стратегии тестирования
- 6.8.2. Дизайн тестовых кейсов
- 6.8.3. Соотношение цены и качества
- 6.8.4. Модели качества
- 6.8.5. Семейство стандартов ISO/IEC 25000 (SQuaRE)
- 6.8.6. Модель качества продукции (ISO 2501n)
- 6.8.7. Модели качества данных (ISO 2501n)
- 6.8.8. Управление качеством программного обеспечения

6.9. Введение в метрики программной инженерии
- 6.9.1. Основные понятия: измерения, метрики и индикаторы
- 6.9.2. Типы метрик в программной инженерии
- 6.9.3. Процесс измерения
- 6.9.4. ISO 25024. Используемые внешние метрики и метрики качества
- 6.9.5. Объектно-ориентированные метрики

6.10. Сопровождение и реинжиниринг программного обеспечения
- 6.10.1. Процесс сопровождения
- 6.10.2. Стандартная схема процесса сопровождения. ISO/EIEC 14764
- 6.10.3. Модель процесса реинжиниринга программного обеспечения
- 6.10.4. Обратное проектирование

Модуль 7. Продвинутое программирование

7.1. Введение в объектно-ориентированное программирование
- 7.1.1. Введение в объектно-ориентированное программирование
- 7.1.2. Разработка классов
- 7.1.3. Введение в унифицированный язык моделирования (UML) для моделирования задач

7.2. Отношения между классами
- 7.2.1. Абстракция и наследование
- 7.2.2. Расширенные концепции наследования
- 7.2.3. Полиморфизм
- 7.2.4. Состав и агрегация

7.3. Введение в паттерны проектирования для объектно-ориентированных задач
- 7.3.1. Что такое паттерны проектирования?
- 7.3.2. Паттерн *Фабрика*
- 7.3.3. Паттерн Одиночка
- 7.3.4. Паттерн *Наблюдатель*
- 7.3.5. Паттерн *Компоновщик*

7.4. Исключения
- 7.4.1. Что такое исключения?
- 7.4.2. Перехват и обработка исключений
- 7.4.3. Запуск исключений
- 7.4.4. Создание исключений

7.5. Пользовательские интерфейсы
- 7.5.1. Введение во фреймворк Qt
- 7.5.2. Позиционирование
- 7.5.3. Что такое события?
- 7.5.4. События: определение и захват
- 7.5.5. Разработка пользовательского интерфейса

7.6. Введение в параллельное программирование
- 7.6.1. Введение в параллельное программирование
- 7.6.2. Концепция процесса и потока
- 7.6.3. Взаимодействие между процессами или потоками
- 7.6.4. Потоки в C++
- 7.6.5. Преимущества и недостатки параллельного программирования

7.7. Управление потоками и синхронизация
- 7.7.1. Жизненный цикл потока
- 7.7.2. Класс Thread
- 7.7.3. Планирование потоков
- 7.7.4. Группы потоков
- 7.7.5. Демонические потоки
- 7.7.6. Синхронизация
- 7.7.7. Механизмы блокировки
- 7.7.8. Механизмы коммуникации
- 7.7.9. Мониторы

7.8. Распространенные задачи в параллельном программировании
- 7.8.1. Задача "производитель – потребитель"
- 7.8.2. Задача о читателях — писателях
- 7.8.3. Задача об обедающих философах

7.9. Документация и тестирование программного обеспечения
- 7.9.1. Почему важно документировать программное обеспечение?
- 7.9.2. Проектная документация
- 7.9.3. Использование инструментов для документирования

7.10. Тестирование программного обеспечения
- 7.10.1. Введение в тестирование программного обеспечения
- 7.10.2. Виды тестирования
- 7.10.3. Единичное тестирование
- 7.10.4. Интеграционное тестирование
- 7.10.5. Валидационное тестирование
- 7.10.6. Тестирование системы

Модуль 8. Повторное использование программного обеспечения

8.1. Общая картина повторного использования программного обеспечения
- 8.1.1. Что такое повторное использование программного обеспечения?
- 8.1.2. Преимущества и недостатки повторного использования программного обеспечения
- 8.1.3. Основные методы повторного использования программного обеспечения

8.2. Введение в паттерны проектирования
- 8.2.1. Что такое паттерн проектирования?
- 8.2.2. Каталог основных паттернов проектирования
- 8.2.3. Как использовать паттерны для решения проблем проектирования
- 8.2.4. Как выбрать лучший паттерн проектирования

8.3. Порождающие паттерны
- 8.3.1. Порождающие паттерны
- 8.3.2. Паттерн *Абстрактная фабрика*
- 8.3.3. Пример реализации паттерна *Абстрактной фабрики*
- 8.3.4. Паттерн Строитель
- 8.3.5. Пример реализации паттерна Строитель
- 8.3.6. Паттерн Абстрактная фабрика vs. Строитель

8.4. Порождающие паттерны (II)
- 8.4.1. Паттерн Фабричный метод
- 8.4.2. *Фабричный метод* vs. *Абстрактная фабрика*
- 8.4.3. Паттерн Одиночка

8.5. Структурные паттерны
- 8.5.1. Структурные паттерны
- 8.5.2. Паттерн *Адаптер*
- 8.5.3. Паттерн *Мост*

8.6. Структурные паттерны (II)
- 8.6.1. Паттерн *Компоновщик*
- 8.6.2. Паттерн декоратор

8.7. Структурные паттерны (III)
- 8.7.1. Паттерн Фасад
- 8.7.2. Паттерн Заместитель

8.8. Поведенческие паттерны
- 8.8.1. Концепция поведенческих паттернов
- 8.8.2. Поведенческий паттерн: цепочка ответственности
- 8.8.3. Поведенческий паттерн порядка

8.9. Поведенческие паттерны (II)
- 8.9.1. Паттерн *Интерпретатор*
- 8.9.2. Паттерн Итератор
- 8.9.3. Паттерн Наблюдатель
- 8.9.4. Паттерн Стратегия

8.10. *Фреймворки*
- 8.10.1. Концепция *фреймворков*
- 8.10.2. Разработка с использованием *фреймворков*
- 8.10.3. Паттерн *Модель-Представление-Контроллер*
- 8.10.4. *Фреймворк* для проектирования графического пользовательского интерфейса
- 8.10.5. *Фреймворки* для разработки веб-приложений
- 8.10.6. *Фреймворки* для управления персистентностью объектов в базах данных

Модуль 9. Искусственный интеллект и инженерия знаний

9.1. Введение в искусственный интеллект и инженерию знаний
- 9.1.1. Краткая история искусственного интеллекта
- 9.1.2. Искусственный интеллект сегодня
- 9.1.3. Инженерия знаний

9.2. Поиск
- 9.2.1. Общие концепции поиска
- 9.2.2. Неинформированный поиск
- 9.2.3. Информированный поиск

9.3. Выполнимость булевых формул, удовлетворительность ограничений и автоматическое планирование
- 9.3.1. Выполнимость булевых формул
- 9.3.2. Проблемы удовлетворительности ограничений
- 9.3.3. Автоматическое планирование и PDDL
- 9.3.4. Планирование как информированный метод поиска
- 9.3.5. Планирование с помощью SAT

9.4. Искусственный интеллект в играх
- 9.4.1. Теория игр
- 9.4.2. Минимакс и Альфа-бета-отсечение
- 9.4.3. Моделирование: Монте-Карло

9.5. Контролируемое и неконтролируемое обучение
- 9.5.1. Введение в машинное обучение
- 9.5.2. Классификация
- 9.5.3. Регрессия.
- 9.5.4. Проверка результатов
- 9.5.5. *Кластеризация*

9.6. Нейронные сети
- 9.6.1. Биологические основы
- 9.6.2. Вычислительная модель
- 9.6.3. Контролируемые и неконтролируемые нейронные сети
- 9.6.4. Простой перцептрон
- 9.6.5. Многослойный перцептрон

9.7. Генетические алгоритмы
- 9.7.1. История
- 9.7.2. Биологическая основа
- 9.7.3. Кодирование проблемы
- 9.7.4. Генерация начальной популяции
- 9.7.5. Основной алгоритм и генетические операторы
- 9.7.6. Оценка индивидов: пригодность

9.8. Тезаурусы, словари, таксономии
- 9.8.1. Словари
- 9.8.2. Таксономии
- 9.8.3. Тезаурусы
- 9.8.4. Онтологии

9.9. Представление знаний: семантическая паутина
- 9.9.1. Семантическая паутина
- 9.9.2. Спецификация: RDF, RDFS и OWL
- 9.9.3. Выводы/рассуждения
- 9.9.4. *Связанные* данные

9.10. Экспертные системы и DSS
- 9.10.1. Экспертные системы
- 9.10.2. Системы поддержки принятия решений

Модуль 10. Продвинутая программная инженерия

10.1. Введение в agile-методологии
- 10.1.1. Модели процессов и методологии
- 10.1.2. Agile и agile-процессы
- 10.1.3. Agile манифест
- 10.1.4. Некоторые agile-методологии
- 10.1.5. Agile vs. Традиционные методологии

10.2. Scrum
- 10.2.1. Истоки и философия Scrum
- 10.2.2. Ценности Scrum
- 10.2.3. Поток процессов Scrum
- 10.2.4. Роли Scrum
- 10.2.5. Артефакты Scrum
- 10.2.6. События Scrum
- 10.2.7. Пользовательские истории
- 10.2.8. Расширения Scrum
- 10.2.9. Agile-сметы
- 10.2.10. Масштабирование Scrum

10.3. Экстремальное программирование
- 10.3.1. Обоснование и обзор XP
- 10.3.2. Жизненный цикл XP
- 10.3.3. Пять основных ценностей
- 10.3.4. Двенадцать основных практик XP
- 10.3.5. Роли участников
- 10.3.6. Промышленная XP
- 10.3.7. Критическая оценка XP

10.4. Разработка программного обеспечения на основе повторного использования
- 10.4.1. Повторное использование программного обеспечения
- 10.4.2. Уровни повторного использования кода
- 10.4.3. Конкретные методы повторного использования
- 10.4.4. Разработка на основе компонентов
- 10.4.5. Преимущества и проблемы повторного использования
- 10.4.6. Планирование повторного использования

10.5. Системная архитектура и модели проектирования программного обеспечения
- 10.5.1. Архитектурное проектирование
- 10.5.2. Общие архитектурные паттерны
- 10.5.3. Отказоустойчивые архитектуры
- 10.5.4. Архитектуры распределенных систем
- 10.5.5. Паттерны проектирования
- 10.5.6. Гамма-паттерны
- 10.5.7. Паттерны проектирования взаимодействия

10.6. Архитектура облачных приложений
- 10.6.1. Основы *облачных вычислений*
- 10.6.2. Качество облачных приложений
- 10.6.3. Архитектурные стили
- 10.6.4. Паттерны проектирования

10.7. Тестирование программного обеспечения: TDD, ATDD и BDD
- 10.7.1. Верификация и валидация программного обеспечения
- 10.7.2. Тестирование программного обеспечения
- 10.7.3. *Разработка через тестирование* (TDD)
- 10.7.4. *Разработка через приемочное тестирование* (ATDD)
- 10.7.5. *Поведенческая разработка* (BDD)
- 10.7.6. BDD и *Cucumber*

10.8. Совершенствование программных процессов
- 10.8.1. Совершенствование программных процессов
- 10.8.2. Процесс совершенствования процесса
- 10.8.3. Модели зрелости
- 10.8.4. Модель CMMI
- 10.8.5. CMMI V2.0
- 10.8.6. CMMI и Agile

10.9. Качество программного продукта: SQuaRE
- 10.9.1. Качество программного обеспечения
- 10.9.2. Модель качества программного продукта
- 10.9.3. Семейство ISO/IEC 25000
- 10.9.4. ISO/IEC 25010: модель качества и характеристики
- 10.9.5. ISO/IEC 25012: качество данных
- 10.9.6. ISO/IEC 25020: измерение качества программного обеспечения
- 10.9.7. ISO/IEC 25022, 25023 и 25024: метрики качества программного обеспечения и данных
- 10.9.8. ISO/IEC 25040: оценка программного обеспечения
- 10.9.9. Процесс сертификации

10.10. Введение в DevOps
- 10.10.1. Концепция DevOps
- 10.10.2. Основные практики

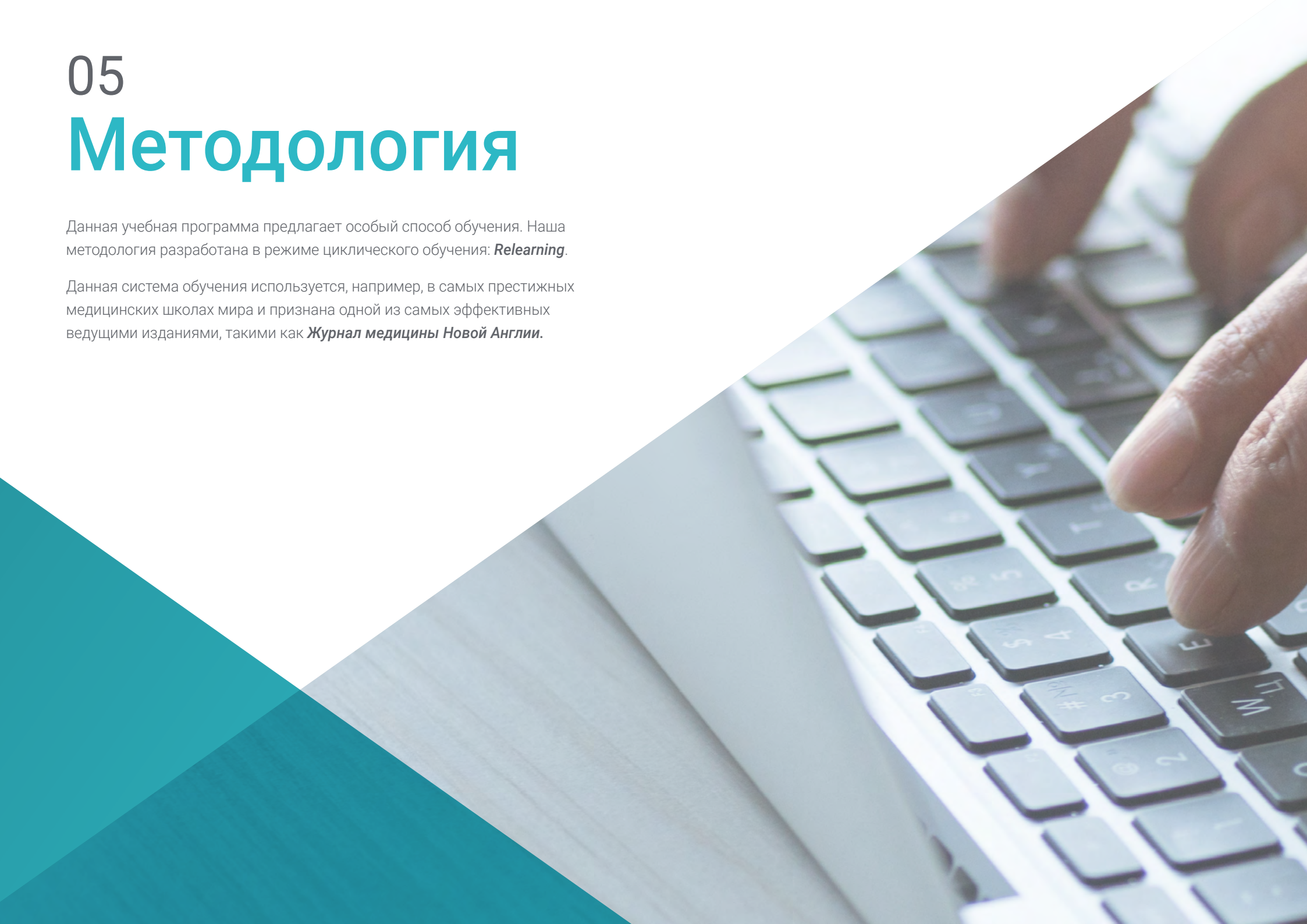
Уникальный, важный и значимый курс обучения для повышения вашей квалификации"

05

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

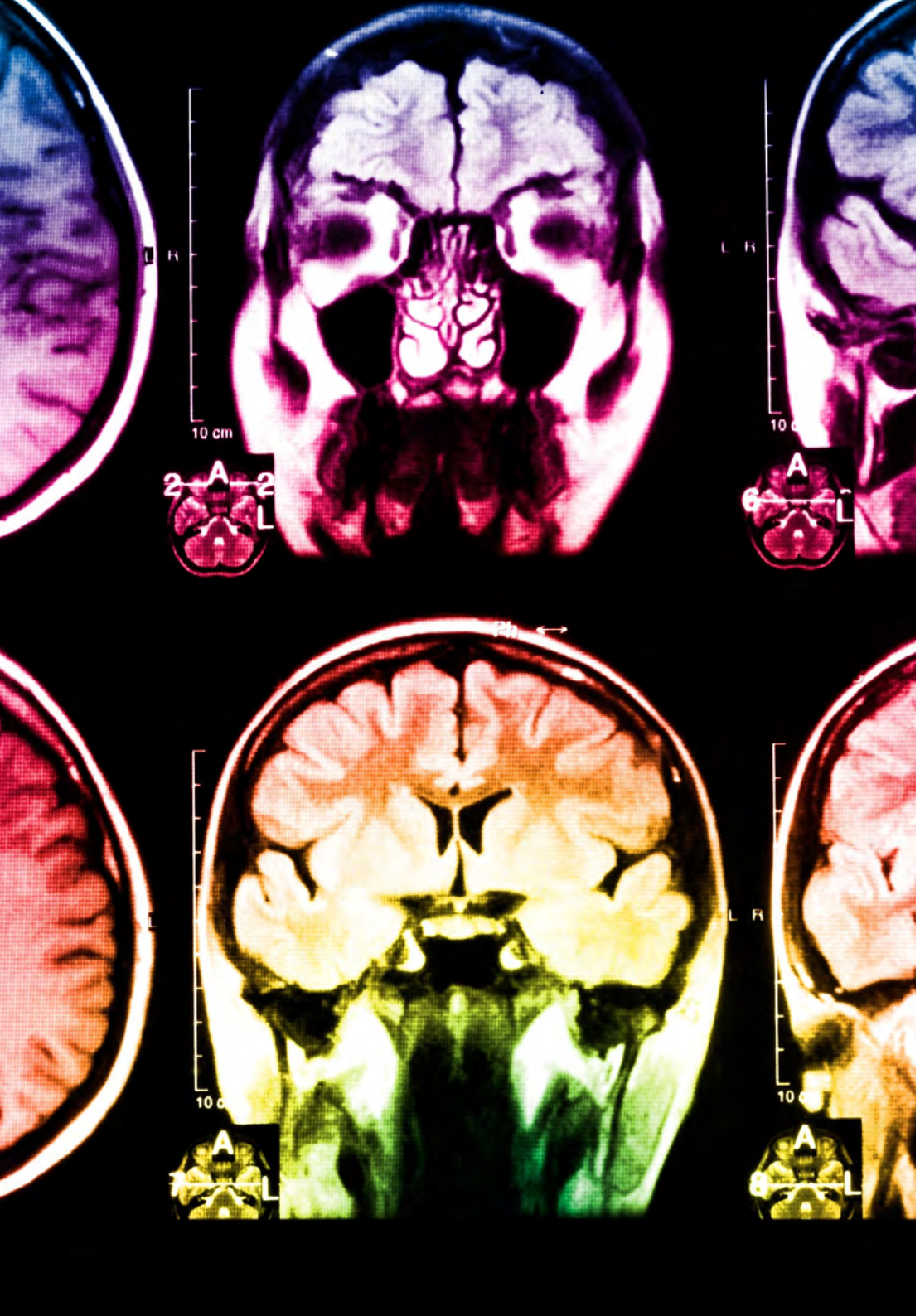
TECH эффективно объединяет метод кейсов с системой 100% онлайн-обучения, основанной на повторении, которая сочетает различные дидактические элементы в каждом уроке.

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.

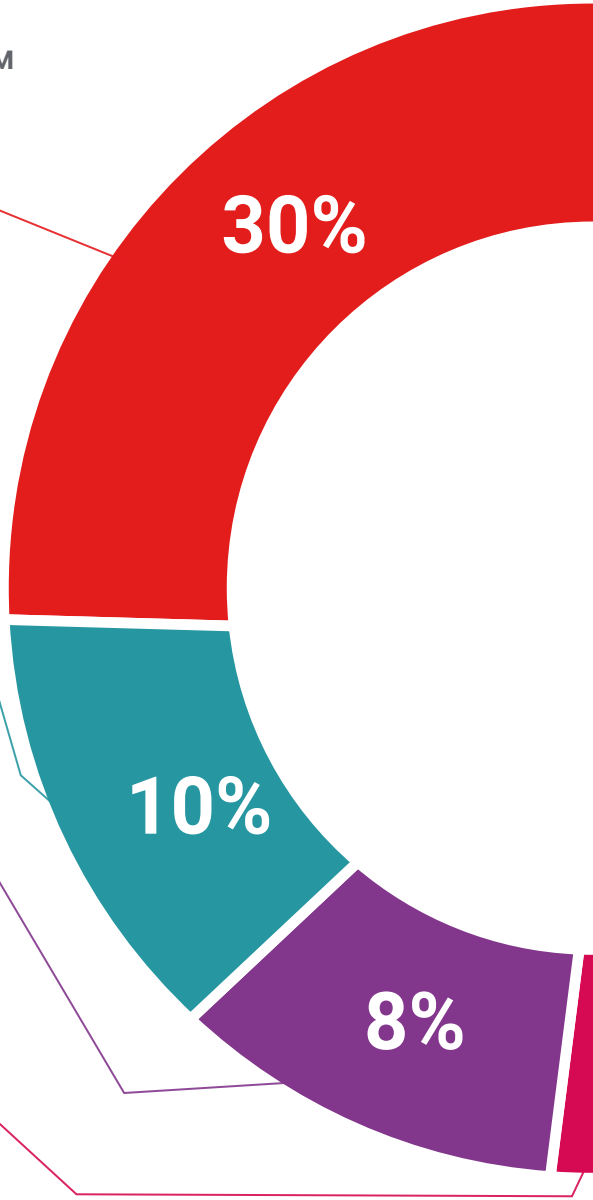
Так называемый метод обучения у эксперта укрепляет знания и память, а также формирует уверенность в наших будущих сложных решениях.

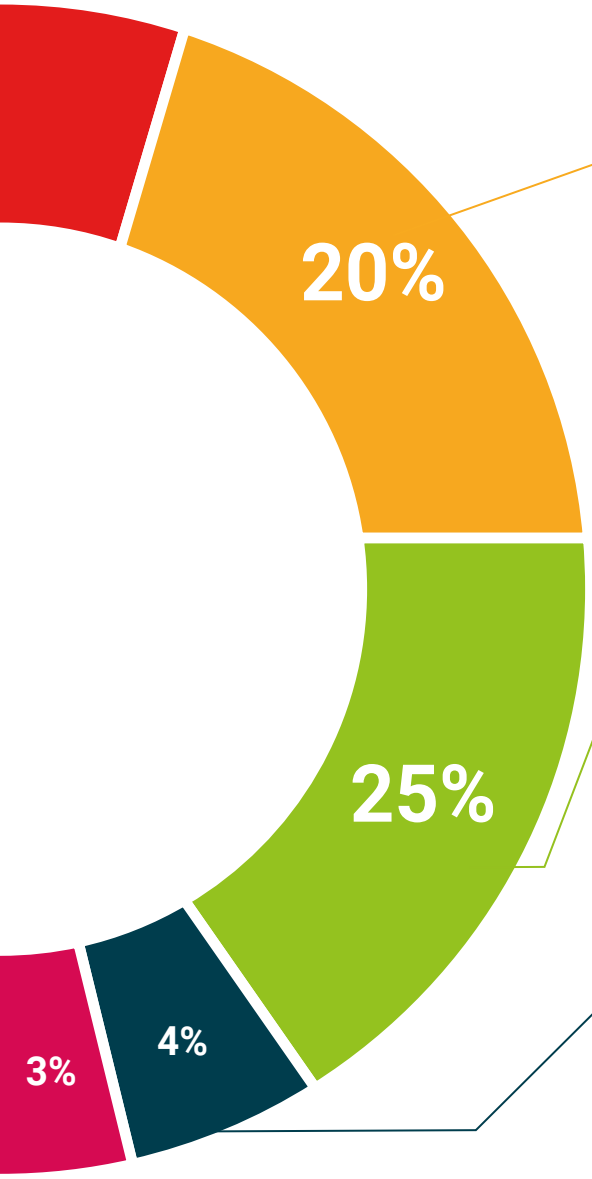
Практика навыков и компетенций

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.

06

Квалификация

Специализированная магистратура в области создания сетевых интерфейсов и приложений 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 оформлением документов"

Данная **Специализированная магистратура в области создания сетевых интерфейсов и приложений** содержит самую полную и современную образовательную программу на рынке.

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

Диплом, выданный **ТЕСН 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 создания сетевых интерфейсов и приложений**

Количество учебных часов: **1500 часов**

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

Создание сетевых интерфейсов и приложений

Данный диплом специализированной программы, присуждаемый Университетом, соответствует 1500 учебным часам, с датой начала дд/мм/гггг и датой окончания дд/мм/гггг.

ТЕСН 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: AFWOR23S techtitute.com/certificates

Специализированная магистратура в области создания сетевых интерфейсов и приложений

Общее распределение учебного плана

Тип предмета	Часы
Обязательный (Обязат.)	1500
Необязательный (Необязат.)	0
Внешние стажировки (ВС)	0
Магистерская диссертация (МД)	0
Всего	1500

Общее распределение учебного плана

Курс	Предмет	Часы	Характер
1º	Взаимодействие человека и компьютера	150	Обязат
1º	Базы данных	150	Обязат
1º	Разработка сетевых приложений	150	Обязат
1º	Свободное программное обеспечение и открытые знания	150	Обязат
1º	Расширенные базы данных	150	Обязат
1º	Программная инженерия	150	Обязат
1º	Продвинутое программирование	150	Обязат
1º	Повторное использование программного обеспечения	150	Обязат
1º	Искусственный интеллект и инженерия знаний	150	Обязат
1º	Продвинутая программная инженерия	150	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

Создание сетевых интерфейсов и приложений

- Формат: **онлайн**
- Продолжительность: **12 месяцев**
- Учебное заведение: **TECH Технологический университет**
- Режим обучения: **16ч./неделя**
- Расписание: **по своему усмотрению**
- Экзамены: **онлайн**

Специализированная магистратура

Создание сетевых интерфейсов и приложений

tech технологический университет