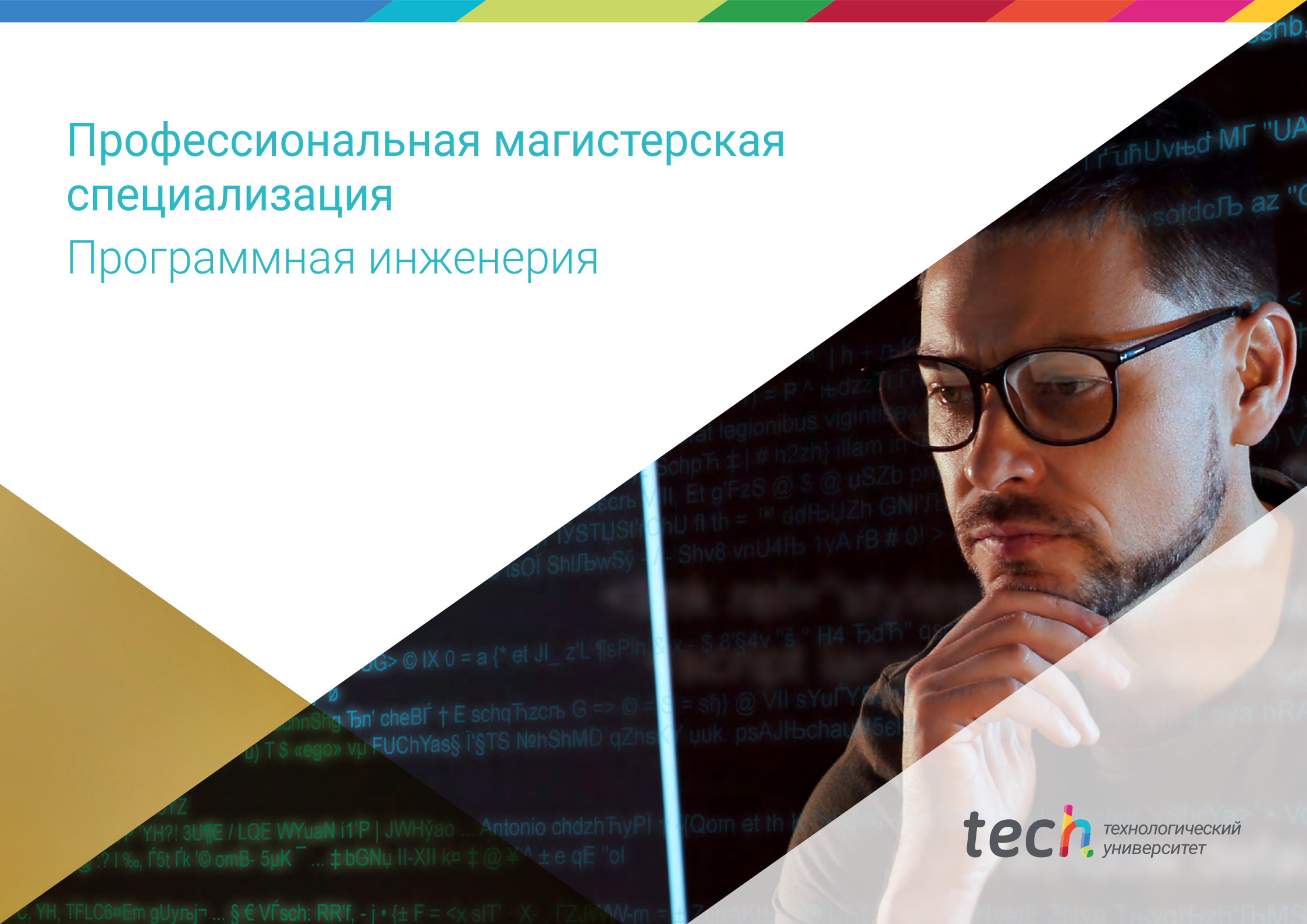
Профессиональная магистерская специализация

Программная инженерия



tech технологический университет

tech технологический университет

Профессиональная магистерская специализация

Программная инженерия

- Формат: **онлайн**
- Продолжительность: **2 года**
- Учебное заведение: **TECH Технологический университет**
- Режим обучения: **16ч./неделя**
- Расписание: **по своему усмотрению**
- Экзамены: **онлайн**

Веб-доступ: www.techtitute.com/ru/information-technology/advanced-master-degree/advanced-master-degree-software-engineering

Оглавление

01

Презентация

Спрос на программное обеспечение в последние годы резко возрос. С появлением новых цифровых платформ, более сложного оборудования и растущей виртуализацией повседневных процессов *программные* инженеры постоянно сталкиваются с новыми проблемами. Поскольку общество все больше и больше привыкает к новым технологиям, а его требования становятся все выше, специалисты по разработке *программного обеспечения* должны адаптироваться к этим требованиям и создавать продукты, соответствующие ожиданиям рынка. Это требует высокого уровня технической экспертизы в нескольких областях знаний информационных технологий.

Вы будете играть ключевую роль в технологическом будущем многих компаний. Получите специализацию в области программной инженерии и начните разрабатывать системы, которые изменят мир к лучшему”

Технологическая индустрия - одна из самых востребованных отраслей на сегодняшний день, поскольку почти каждый человек ежедневно взаимодействует с каким-либо цифровым устройством. В этом контексте инженеры-программисты представляют собой передовую линию всего процесса технологического развития, поскольку именно им постоянно приходится обновлять системы, разрабатывать новые и предлагать разумные решения возникающих проблем. С этой точки зрения специалисты по разработке программного обеспечения должны быть очень решительными людьми, с большими техническими знаниями и выдающейся способностью адаптироваться к любым типам разработки и средам.

Руководствуясь этой целью, компания TECH разработала Профессиональную магистерскую специализацию в области программной инженерии, предлагающую полную и высоко уровневую подготовку для всех разработчиков, которые хотят специализировать свою карьеру и направить ее на создание систем. С одной стороны, в программе рассматриваются различные методологии создания и управления проектом разработки программного обеспечения, а также все аспекты, которые необходимо учитывать в отношении вычислений, требований и платформ. С другой стороны, речь идет о безопасности как самого программного обеспечения, так и информационных систем и рабочей среды, используемых в процессе работы. По завершении обучения студент овладеет всеми необходимыми знаниями, чтобы стать эффективным и высоко компетентным специалистом в области программной инженерии.

Кроме того, одним из главных преимуществ этой программы является ее 100% онлайн-формат. Это означает, что студенту не нужно подстраиваться под фиксированное расписание и не обязательно посещать определенный физический центр. Таким образом, студент имеет возможность свободно управлять изучением выбранного предмета, в своем собственном темпе и с учетом своих обязательств, планируя свое расписание так, как он считает нужным.

Эта **Профессиональная магистерская специализация в области программной инженерии** содержит самую полную и современную программу на рынке. Основными особенностями обучения являются:

- Разработка практических кейсов, представленных практикующими экспертами
- Наглядное, схематичное и исключительно практичное содержание курса предоставляет научную и практическую информацию по тем дисциплинам, которые необходимы для осуществления профессиональной деятельности
- Практические упражнения для оценки самого себя, самоконтроля и улучшения успеваемости
- Особое внимание уделяется инновационным методологиям в области программной инженерии
- 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

Можете ли вы представить, что принимали участие в разработке Netflix? Пора прекратить фантазировать и сосредоточить вашу карьеру на лучших программных проектах"

Ваш опыт и знания могут сыграть решающую роль в крупных проектах, включающих множество требований. Не упустите возможность отличиться в своей карьере и поступите сейчас же на Профессиональную магистерскую специализацию в области программной инженерии”

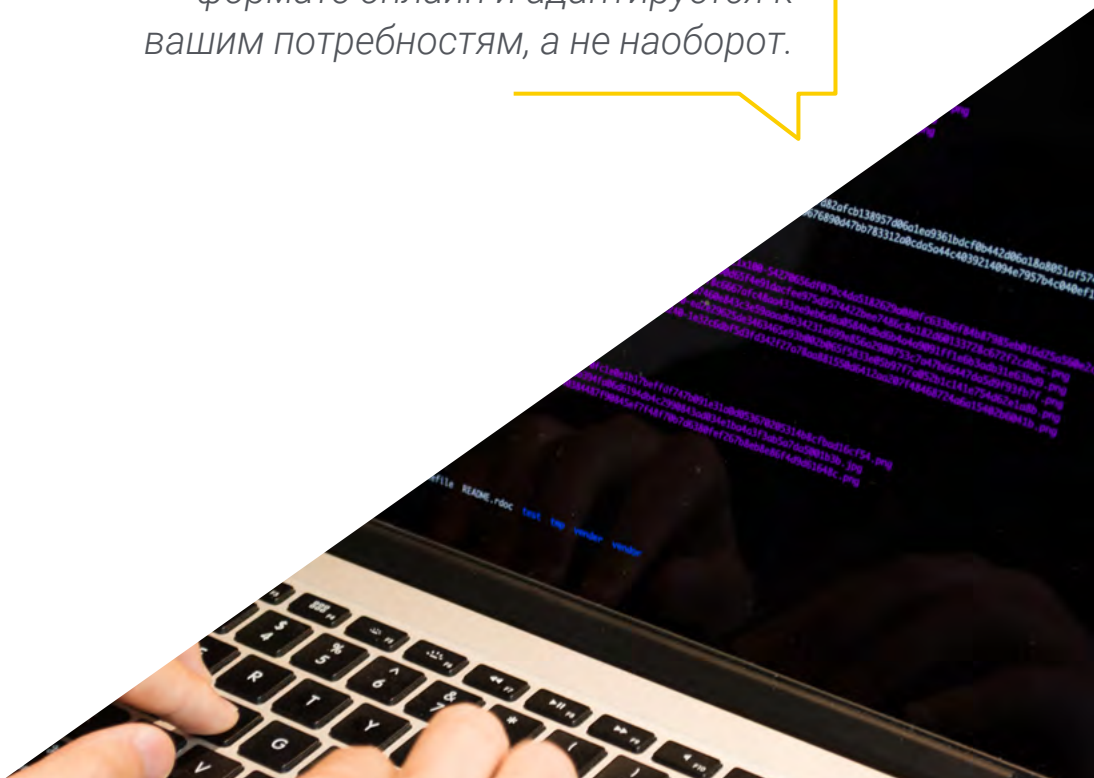
Цель TECH - сделать из вас отличного программиста. Вам гарантирован доступ к самому лучшему материалу и обучению.

Изучайте его когда, где и как хотите. Программа на 100% работает в формате онлайн и адаптируется к вашим потребностям, а не наоборот.

В преподавательский состав входят профессионалы в области программной инженерии, которые привносят в эту программу свой опыт работы, а также признанные специалисты из ведущих сообществ и престижных университетов.

Мультимедийное содержание, разработанное с использованием новейших образовательных технологий, позволит профессионалам проходить обучение в симулированной среде, обеспечивающей иммерсивный учебный процесс, основанный на обучении в реальных ситуациях.

Разработка данной программы основана на проблемно-ориентированном обучении, с помощью которого студент должен попытаться решить различные ситуации профессиональной практики, возникающие на протяжении всей программы. Для этого практикующему будет помогать инновационная интерактивная видеосистема, созданная известными и опытными специалистами.



02

Цели

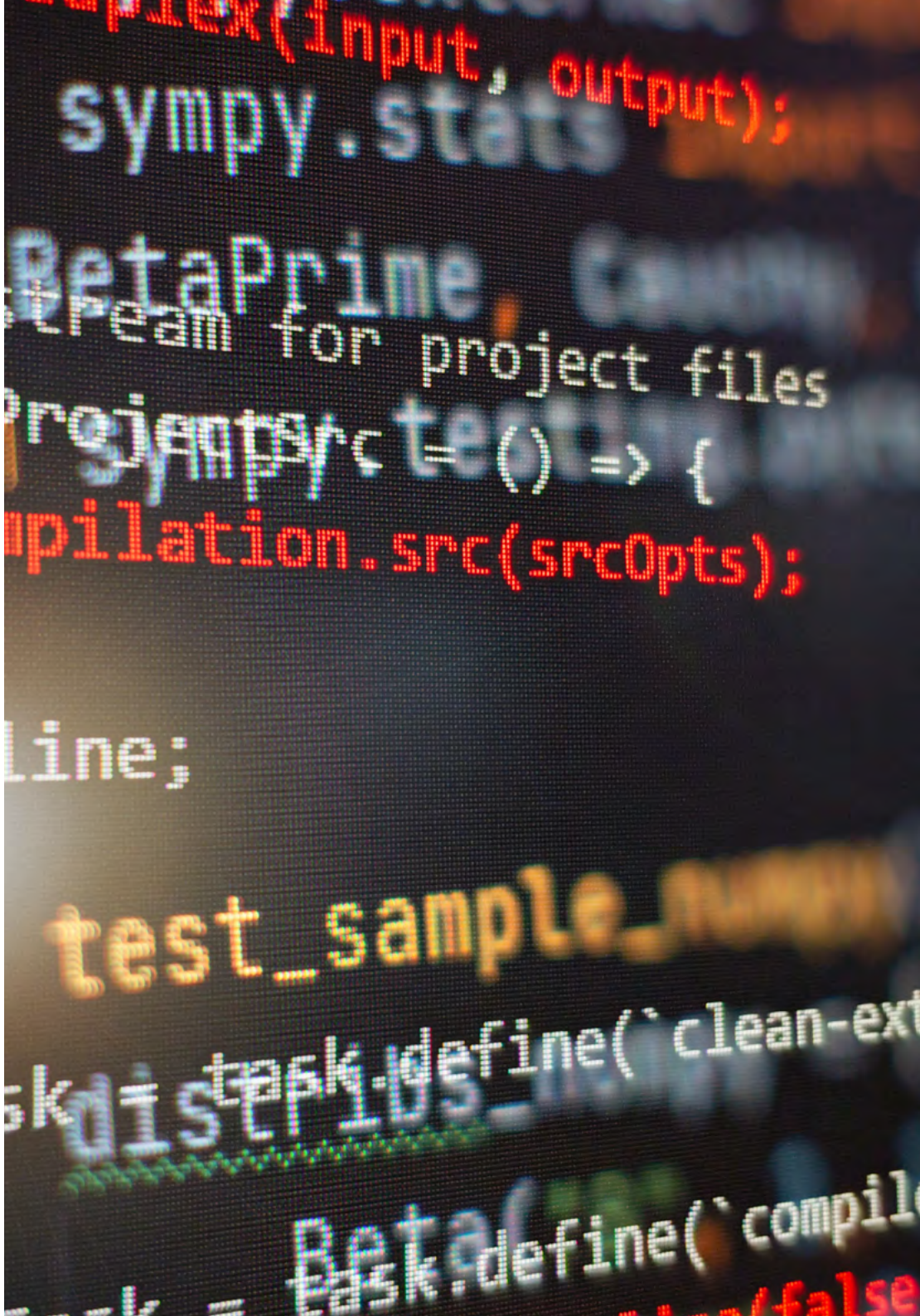
Эта Профессиональная магистерская специализация в области программной инженерии была разработана с целью предложить всем профессионалам в области ИТ высшее образование, необходимое для того, чтобы сосредоточить свою карьеру на разработке современного программного обеспечения, адаптированного к новым колеблющимся реалиям рынка. Благодаря высоко техническим знаниям, которые преподаются на протяжении всего курса, студенты значительно повысят свои шансы на профессиональный рост и доступ к работе в крупных компаниях отрасли.

Профессиональная магистерская специализация, которая станет самым значительным положительным толчком, который вы можете дать своей карьере на пути к профессиональному успеху"

Общие цели

- Приобрести новые компетенции, необходимые и востребованные с точки зрения новых технологий и последних разработок программного обеспечения
- Дополнить полученные знания навыками в области вычислительной техники и компьютерной структуры, включая математические, статистические и физические основы, необходимые в инженерном деле
- Расширить знания в области программной инженерии и компьютерных систем, используя последние разработки и самые инновационные методологии
- Работать над сложными программными проектами и средами, умея находить разумные решения для различных проблем

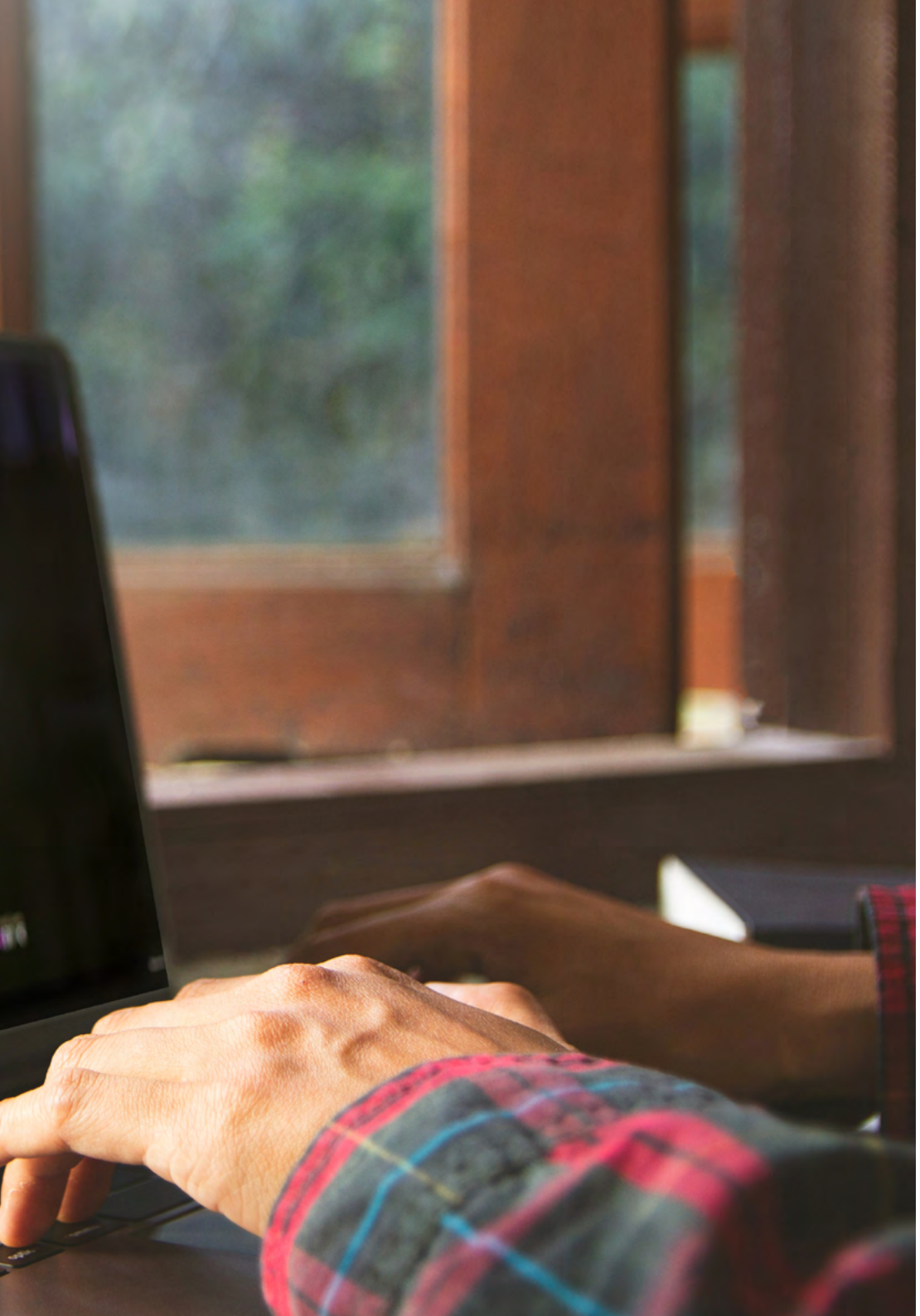
Специализация, которая поможет вам освоить разработку программного обеспечения с уникальным набором навыков, востребованных каждой ведущей компанией в этом секторе"



Конкретные цели

- Знать основы программной инженерии, а также свод правил или принципов этики и профессиональной ответственности во время и после разработки
- Понимать процесс разработки программного обеспечения, различных моделей программирования и парадигмы объектно-ориентированного программирования
- Понимать различные типы моделирования приложений и шаблоны проектирования на унифицированном языке моделирования (UML)
- Ознакомиться с фундаментальными концепциями управления проектами и жизненным циклом управления проектами
- Понять, как функционирует управление проектами, включая планирование, обеспечение, контроль, статистические концепции и доступные инструменты
- Приобрести основные знания, связанные с профессиональной ответственностью в области управления проектами
- Понимать различные платформы разработки программного обеспечения
- Приобрести необходимые знания для разработки приложений и графических интерфейсов на языках *Java* и *.NET*
- Изучить среды разработки мобильных приложений для *Android*, процессов отладки и публикации
- Понимать разработку облачных приложений и определять правильные процедуры для их внедрения
- Освоить процесс создания веб-контента с помощью языка разметки HTML
- Приобрести необходимые знания для разработки веб-приложений на стороне веб-клиента
- Разрабатывать приложения со сложной структурой, используя различные процедуры, функции и объекты, входящие в состав *JavaScript*
- Научиться использовать интерфейс программирования DOM для документов HTML и XML, чтобы изменять их структуру, стиль и содержание
- Знать концепцию веб-юзабилити, ее преимущества, принципы, методы и приемы, позволяющие сделать веб-сайт удобным для пользователя
- Понять архитектуру программного обеспечения Model–view–controller (MVC), которая разделяет данные, пользовательский интерфейс и логику управления приложения на три отдельных компонента
- Приобрести навыки использования веб-сервисов, используя XML, SOA и REST
- Понять процесс информационной безопасности, его последствия для конфиденциальности, целостности, доступности и экономических затрат
- Научиться использовать передовые методы обеспечения безопасности при управлении услугами информационных технологий
- Приобрести знания для правильной сертификации процессов безопасности
- Понять механизмы и методы аутентификации для контроля доступа, а также процесс аудита доступа
- Понимать программы управления безопасностью, управление рисками и разработку политики безопасности
- Узнать о планах обеспечения непрерывности бизнеса, их этапах и процессе поддержания
- Знать процедуры правильной защиты компании с помощью сетей DMZ, использования систем обнаружения вторжений и других методологий
- Понимать проблемы безопасности программного обеспечения, уязвимости и способы их классификации
- Проанализировать различные веб-серверы, которые являются модными на сегодняшнем рынке

- Понять процесс статистики использования и балансировки нагрузки на веб-серверах
- Приобрести необходимые знания для установки, администрирования, настройки и обеспечения безопасности
- Понять концепции и процессы проектирования программного обеспечения, изучая также проектирование архитектуры, проектирование на уровне компонентов и на основе паттернов
- Понять различные модели системных архитектур и проектирования программного обеспечения, а также архитектуру облачных приложений
- Углубиться в изучении совершенствования процесса разработки программного обеспечения и качества программного обеспечения с использованием стандартов ISO/IEC
- Понять важность разработки требований в процессе разработки программного обеспечения
- Углубиться в понимании источников требований и методов их выявления, поскольку они являются неотъемлемой частью процесса
- Понимать и применять прототипирование как важную часть процесса разработки
- Заложить основы судебной экспертизы в мире аудита программного обеспечения и информационных технологий
- Ознакомиться с фундаментальными концепциями управления проектами и жизненным циклом управления проектами
- Научиться составлять расписание для управления временем, разработки бюджета и реагирования на риски
- Понять, как функционирует управление проектами, включая планирование, обеспечение, контроль, статистические концепции и доступные инструменты

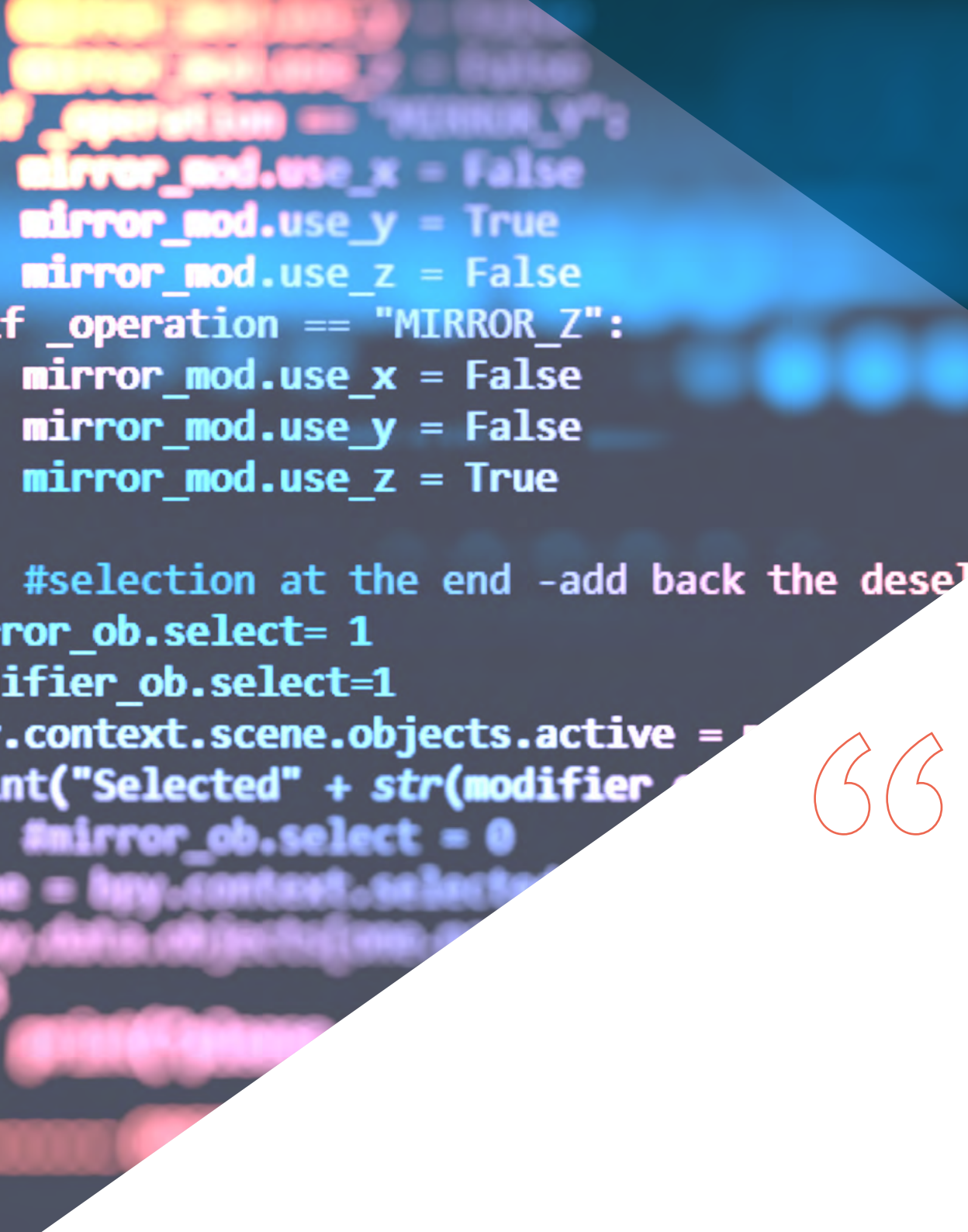


Полное обучение, в ходе которого вы получите знания, необходимые для того, чтобы конкурировать среди лучших”

03

Компетенции

Программные инженеры постоянно обновляют свои знания, поскольку сами инструменты и реалии, в которых они работают, ежедневно модернизируются. Это требует регулярного процесса обучения, для которого необходимы различные общие навыки, как в плане чистой разработки программного обеспечения, так и в других дисциплинах, таких как управление командой. Понимая это обстоятельство, TECH разработал Профессиональную магистерскую специализацию в области программной инженерии с целью предоставить студентам все возможные и необходимые навыки, чтобы сделать их собственный непрерывный процесс обучения более легким и автоматическим.

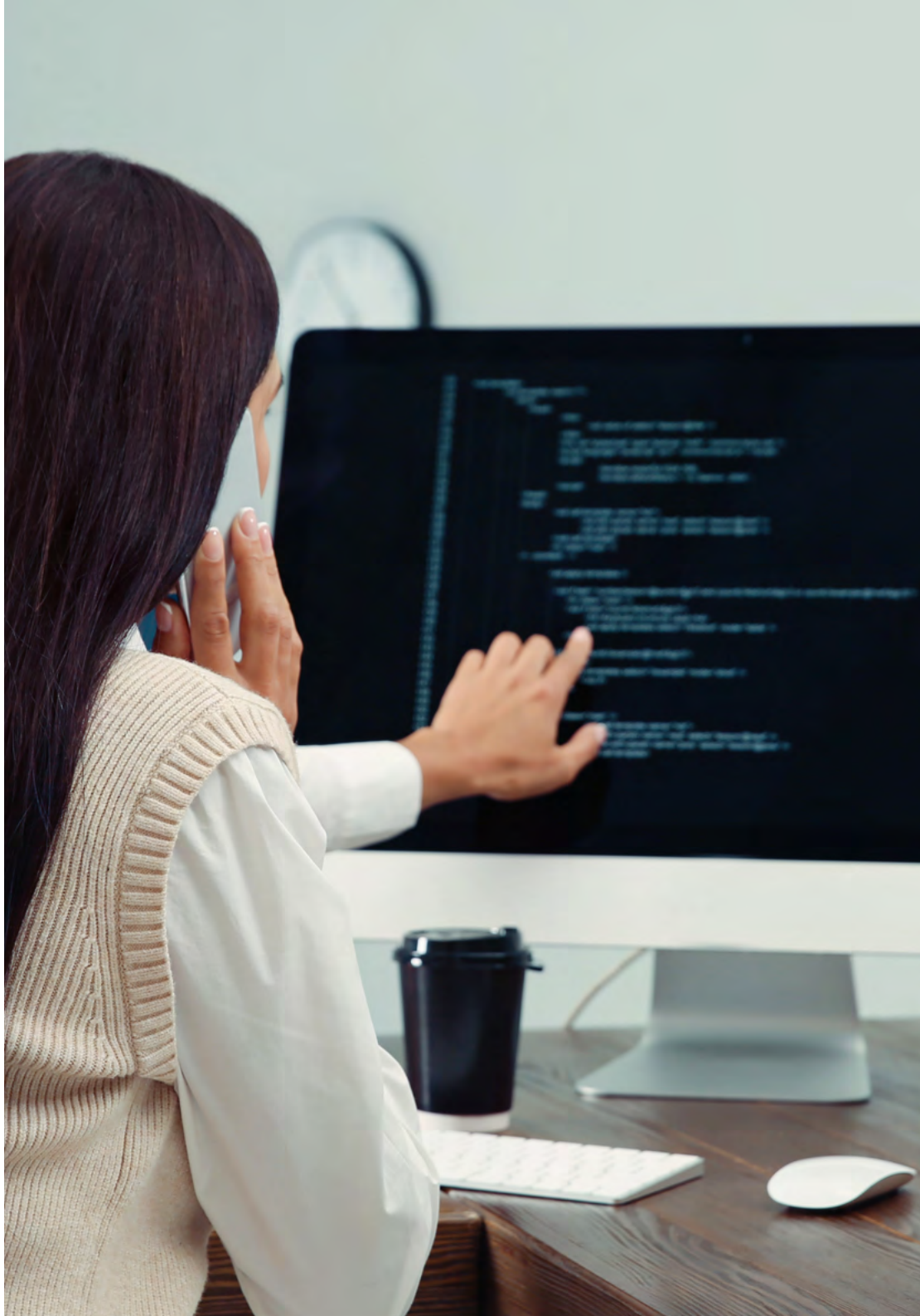


Со всеми навыками, которые вы приобретете, проходя обучение в рамках этой Профессиональной магистерской специализации в области программной инженерии, вы станете лучшим кандидатом на любую должность, на которую претендуете"

Общие профессиональные навыки

- Разработать систему программного обеспечения с учетом всех этапов разработки, платформ безопасности и вопросов безопасности
- Правильно и профессионально обрабатывать все данные, полученные в ходе разработки
- Применять наилучшую рабочую методологию, соответствующую проекту или людям, вовлеченным в проект
- Знать всю реальность программной инженерии и быстро и эффективно предотвращать возможные риски или проблемы

Вы можете сделать шаг навстречу лучшему профессиональному будущему. Сделайте это и поступайте прямо сейчас на эту Профессиональную магистерскую специализацию, которая откроет множество дверей для вашей профессиональной карьеры”



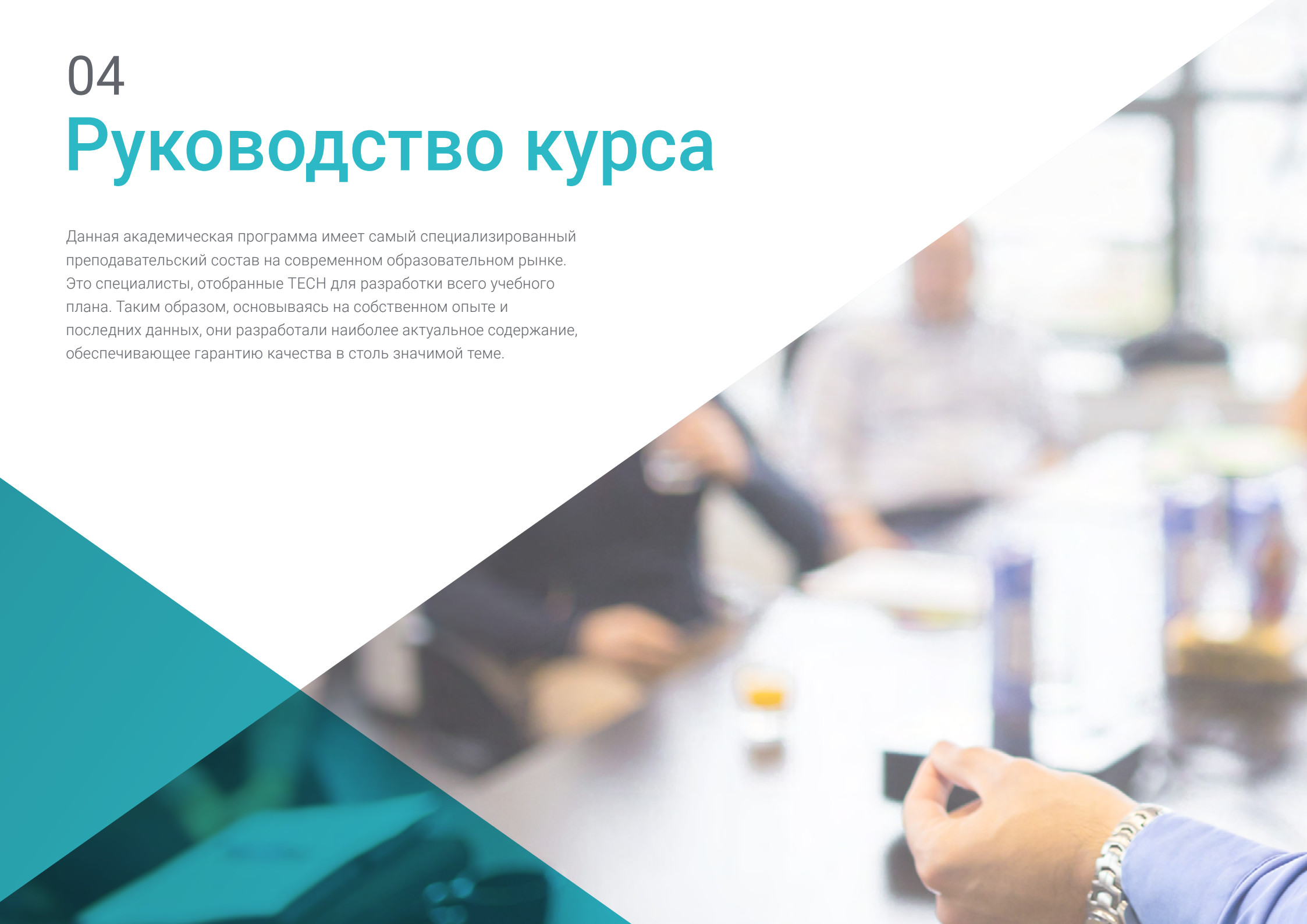
Профессиональные навыки

- Понимать различные типы моделирования приложений и шаблоны проектирования на унифицированном языке моделирования (UML)
- Понять, как функционирует управление проектами, включая планирование, обеспечение, контроль, статистические концепции и доступные инструменты
- Использовать знания, необходимые для разработки приложений и графических интерфейсов на следующих языков: *Java y .NET*
- Освоить процесс создания веб-контента с помощью языка разметки HTML
- Освоить процесс взаимодействия с клиентами с помощью: форм, *Cookies* и управления сессиями
- Понять механизмы и методы аутентификации для контроля доступа, а также процесс аудита доступа
- Понять применение безопасности на различных этапах жизненного цикла программного обеспечения
- Знать концепцию, функционирование, архитектуру, ресурсы и содержание веб-сервера
- Понимать различные вспомогательные инструменты, методологии и последующий анализ в ходе аудита безопасности интернета и мобильных устройств
- Понимать политику и стандарты безопасности, применяемые к онлайн-приложениям
- Уметь составлять, планировать, разрабатывать и подписывать проекты в области компьютерной инженерии, направленные на разработку или эксплуатацию компьютерных систем, услуг и приложений
- Руководить деятельностью ИТ-проектов
- Уметь определять, оценивать и выбирать аппаратные и программные платформы для разработки и внедрения ИТ-систем, услуг и приложений
- Уметь разрабатывать с использованием *Scrum*, экстремального программирования и методов разработки программного обеспечения, основанных на повторном использовании
- Иметь способность проектировать, разрабатывать и поддерживать компьютерные системы, услуги и приложения, используя методы программной инженерии как инструмент обеспечения качества
- Внедрить основы симметричной криптографии и асимметричной криптографии, а также их основные алгоритмы
- Применять основные концепции, связанные с информационными системами в бизнесе, и определять возможности и потребности информационных систем
- Знать, как разработать график с учетом времени, бюджета и управления рисками
- Понять, как функционирует руководство и управление ИКТ, какие стандарты ISO/IEC его регулируют и какие передовые методы следует внедрять
- Планировать управление безопасностью и управлять основными механизмами защиты информационных активов

04

Руководство курса

Данная академическая программа имеет самый специализированный преподавательский состав на современном образовательном рынке. Это специалисты, отобранные TECH для разработки всего учебного плана. Таким образом, основываясь на собственном опыте и последних данных, они разработали наиболее актуальное содержание, обеспечивающее гарантию качества в столь значимой теме.



TECH предлагает вам самый специализированный преподавательский состав в области обучения. Поступайте прямо сейчас и наслаждайтесь качеством, которого вы заслуживаете”

Приглашенный международный руководитель

Даррен Пулшифер - опытный архитектор программного обеспечения, новатор с выдающимся международным послужным списком в области разработки программного обеспечения и микропрограмм. Он обладает высокоразвитыми навыками общения, управления проектами и ведения бизнеса, которые позволили ему возглавить крупные глобальные инициативы.

На протяжении своей карьеры он занимал ответственные посты, такие как главный архитектор решений для государственного сектора в Intel Corporation, где он продвигал современный бизнес, процессы и технологии для клиентов, партнеров и пользователей в государственном секторе. Кроме того, он основал компанию Yoly Inc., где также занимал пост генерального директора, занимаясь разработкой инструмента для агрегации и диагностики социальных сетей на основе программного обеспечения как услуги (SaaS) с использованием технологий Big Data и Web 2.0.

Кроме того, он работал в других компаниях, в том числе старшим директором по инженерным вопросам в Dell Technologies, где возглавлял отдел больших данных в облачном бизнесе, руководил командами в США и Китае по управлению крупными проектами и реструктуризации бизнес-подразделений для успешной интеграции. Он также занимал должность директора по информационным технологиям (Chief Information Officer) в компании XanGo, где руководил такими проектами, как поддержка справочной службы, поддержка производства и разработка решений.

Среди многочисленных специализаций, в которых он является экспертом, - облачные технологии, кибербезопасность, генеративный искусственный интеллект, разработка программного обеспечения, сетевые технологии, облачные нативные разработки и контейнерная экосистема. Своими знаниями он делится в еженедельном подкасте и информационном бюллетене "Embracing Digital Transformation", который он создал и представил, помогая организациям успешно пройти через цифровую трансформацию, используя людей, процессы и технологии.

Г-н Пулшифер , Даррен

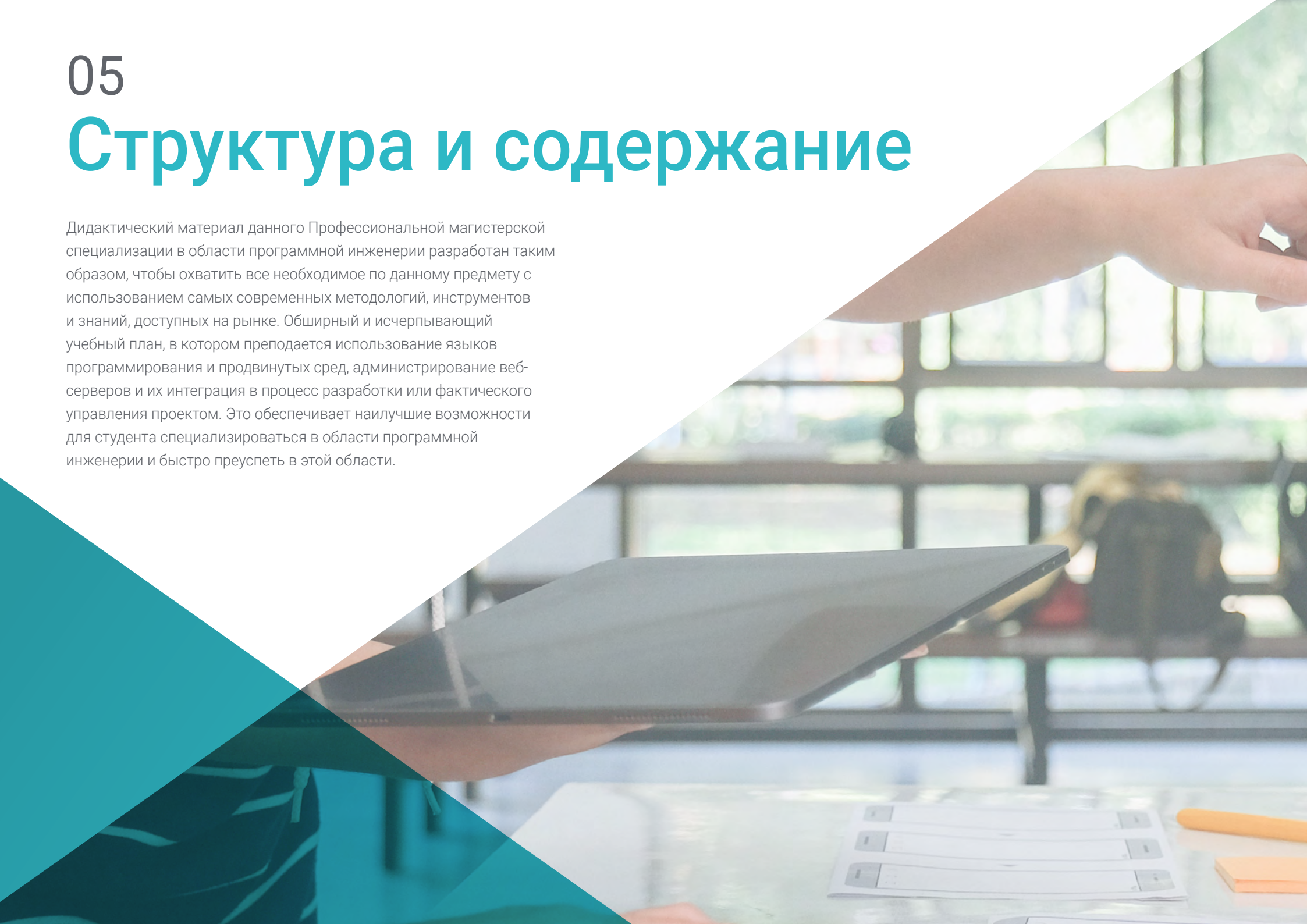
- Главный архитектор решений для государственного сектора в Intel, Калифорния, США
- Ведущий и продюсер программы “Embracing Digital Transformation”, Калифорния
- Основатель и генеральный директор компании Yoly Inc., Арканзас
- Старший директор по инженерным вопросам в Dell Technologies, Арканзас
- Главный информационный директор (Chief Information Officer) компании XanGo, штат Юта
- Старший архитектор в Cadence Design Systems, Калифорния
- Старший менеджер по проектным процессам в Lucent Technologies, Калифорния
- Инженер-программист в компании Cemax-Icon, Калифорния
- Инженер-программист в компании ISG Technologies, Канада
- MBA в области управления технологиями в Университете Феникса, Университет Феникса, Калифорния
- Бакалавр наук по информатике и электротехнике в Университете Бригама Янга

Благодаря TECH вы сможете учиться у лучших мировых профессионалов”

05

Структура и содержание

Дидактический материал данного Профессиональной магистерской специализации в области программной инженерии разработан таким образом, чтобы охватить все необходимое по данному предмету с использованием самых современных методологий, инструментов и знаний, доступных на рынке. Обширный и исчерпывающий учебный план, в котором преподается использование языков программирования и продвинутых сред, администрирование веб-серверов и их интеграция в процесс разработки или фактического управления проектом. Это обеспечивает наилучшие возможности для студента специализироваться в области программной инженерии и быстро преуспеть в этой области.



Покажите, что ваши знания соответствуют вашему стремлению быть лучшим профессионалом, и добавьте большую ценность в свое резюме с этой Профессиональной магистерской специализацией в области программной инженерии”

Модуль 1. Методологии, разработка и качество в программной инженерии

1.1. Введение в программное обеспечение (ПО)
- 1.1.1. Введение
- 1.1.2. Кризис программного обеспечения
- 1.1.3. Различия между программной инженерией и компьютерными науками
- 1.1.4. Этика и профессиональная ответственность в программной инженерии
- 1.1.5. Фабрики разработки программного обеспечения

1.2. Процесс разработки программного обеспечения
- 1.2.1. Определение
- 1.2.2. Модель процесса программного обеспечения
- 1.2.3. Унифицированный процесс разработки ПО

1.3. Развитие объектно-ориентированного ПО
- 1.3.1. Введение
- 1.3.2. Принципы объектной ориентации
- 1.3.3. Определение объекта
- 1.3.4. Определение класса
- 1.3.5. Объектно-ориентированный анализ vs. Объектно-ориентированная разработка

1.4. Разработка программного обеспечения на основе моделей
- 1.4.1. Необходимость моделирования
- 1.4.2. Модель ПО
- 1.4.3. Моделирование объектов
- 1.4.4. UML
- 1.4.5. Инструменты CASE

1.5. Моделирование приложений и паттернов разработки с UML
- 1.5.1. Расширенное моделирование требований
- 1.5.2. Усовершенствованное статическое моделирование
- 1.5.3. Продвинутое динамическое моделирование
- 1.5.4. Моделирование компонентов
- 1.5.5. Введение в шаблоны проектирования с UML
- 1.5.6. *Adapter*
- 1.5.7. *Factory*
- 1.5.8. *Singleton*
- 1.5.9. *Strategy*
- 1.5.10. *Composite*
- 1.5.11. *Facade*
- 1.5.12. *Observer*

1.6. Инженерия, управляемая моделями
- 1.6.1. Введение
- 1.6.2. Метамоделирование систем
- 1.6.3. MDA
- 1.6.4. DSL
- 1.6.5. Уточнения модели с помощью OCL
- 1.6.6. Преобразования моделей

1.7. Онтологии в программной инженерии
- 1.7.1. Введение
- 1.7.2. Инженерия онтологии
- 1.7.3. Применение полученных онтологии в программной инженерии

1.8. Гибкие методологии для разработки программного обеспечения *Scrum*
- 1.8.1. Что такое гибкость программного обеспечения?
- 1.8.2. Манифест agile
- 1.8.3. Дорожная карта гибкого проекта
- 1.8.4. *Product Owner*
- 1.8.5. Пользовательские истории
- 1.8.6. Планирование и оценивание agile
- 1.8.7. Измерения в процессе разработки agile
- 1.8.8. Введение в Scrum
- 1.8.9. Роли
- 1.8.10. *Бэклог* продукта
- 1.8.11. *Sprint*
- 1.8.12. Собрания

1.9. Методология *бережливой* разработки программного обеспечения
- 1.9.1. Введение
- 1.9.2. *Канбан*

1.10. Качество и совершенствование процессов программного обеспечения
- 1.10.1. Введение
- 1.10.2. Измерение программного обеспечения
- 1.10.3. Тестирование программного обеспечения
- 1.10.4. Модель качества программного процесса: CMMI

Модуль 2. Управление проектами прораммного обеспечения

2.1. Фундаментальные концепции управления проектами и жизненного цикла управления проектами
- 2.1.1. Что такое проект?
- 2.1.2. Общая методология
- 2.1.3. Что такое управление проектами?
- 2.1.4. Что такое план проекта?
- 2.1.5. Преимущества
- 2.1.6. Жизненный цикл проектов
- 2.1.7. Группы процессов или жизненный цикл управления проектами
- 2.1.8. Взаимосвязь между группами процессов и областями знаний
- 2.1.9. Взаимосвязи между жизненным циклом продукта и проекта

2.2. Начало и планирование
- 2.2.1. От идеи до реализации проекта
- 2.2.2. Разработка акта проекта
- 2.2.3. Начальное совещание по проекту
- 2.2.4. Задачи, знания и навыки в процессе запуска
- 2.2.5. План проекта
- 2.2.6. Разработка базового плана Шаги
- 2.2.7. Задачи, знания и навыки в процессе планирования

2.3. Управление *заинтересованными сторонами* и информационно разъяснительная работа
- 2.3.1. Определять заинтересованных сторон
- 2.3.2. Разработка плана по управлению заинтересованными сторонами
- 2.3.3. Управление взаимодействием между заинтересованными сторонами
- 2.3.4. Контроль взаимодействия между заинтересованными сторонами
- 2.3.5. Цель проекта
- 2.3.6. Управление масштабом и его план
- 2.3.7. Сбор информации о требованиях
- 2.3.8. Определение сферы применения
- 2.3.9. Создание структуры разбивки работ WBS
- 2.3.10. Утверждение и контроль масштаба

2.4. Разработка графика
- 2.4.1. Управление временем и его планирование
- 2.4.2. Определение деятельности
- 2.4.3. Составление последовательности деятельности
- 2.4.4. Оценка ресурсов деятельности
- 2.4.5. Оценка продолжительности виды деятельности
- 2.4.6. Разработка графика и расчет критического пути
- 2.4.7. Контроль расписания

2.5. Составление бюджета и реагирование на риски
- 2.5.1. Оценка затрат
- 2.5.2. Разработка бюджета и S-образной кривой
- 2.5.3. Контроль затрат и метод затраченной стоимости
- 2.5.4. Понятие риска
- 2.5.5. Как проводить анализ рисков
- 2.5.6. Разработка плана реагирования

2.6. Управление качеством
- 2.6.1. Планирование качества
- 2.6.2. Обеспечение качества
- 2.6.3. Контроль качества
- 2.6.4. Основные статистические концепции
- 2.6.5. Инструменты управления качеством

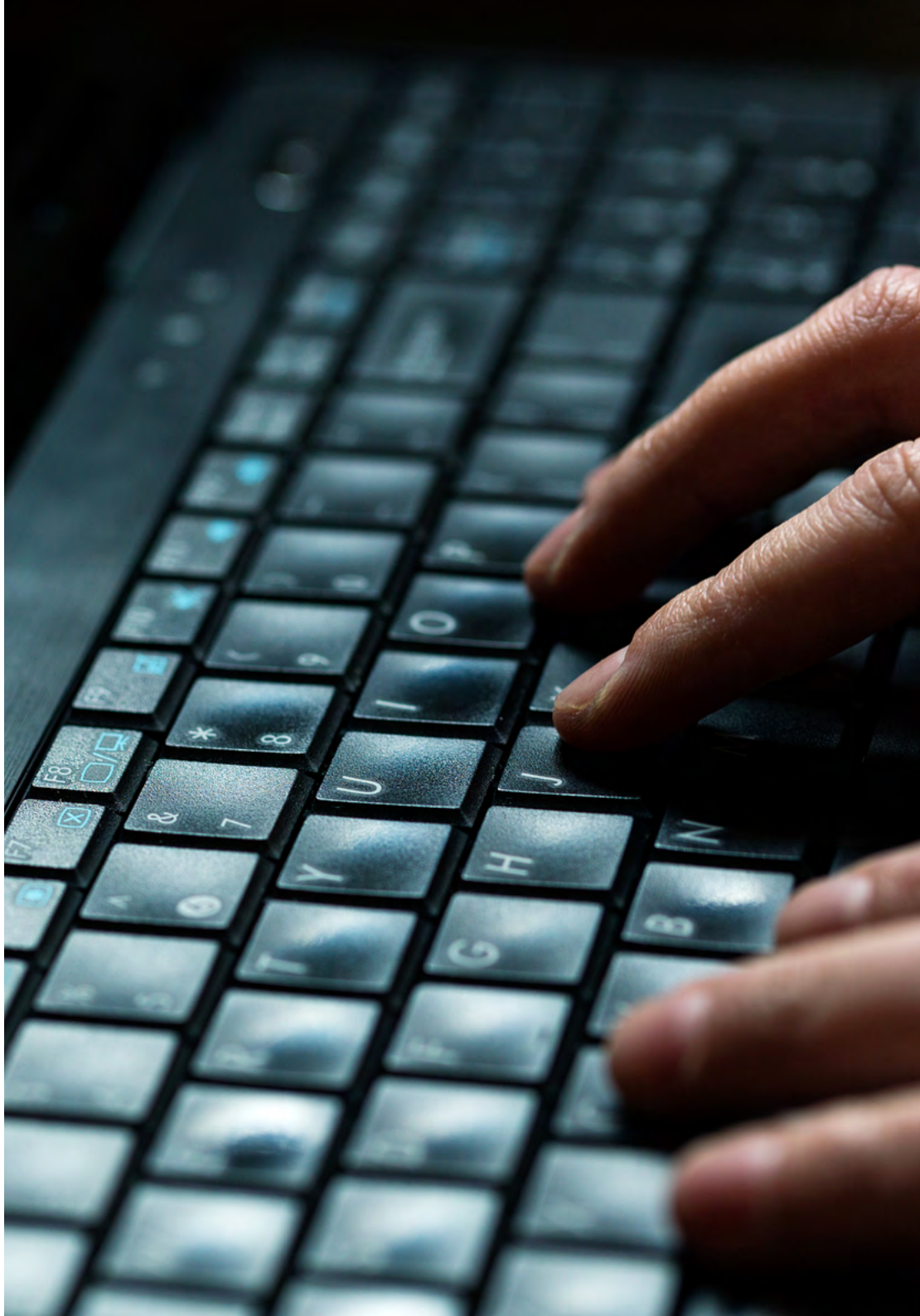
2.7. Коммуникация и человеческие ресурсы
- 2.7.1. Планирование управления коммуникациями
- 2.7.2. Анализ требований к коммуникациям
- 2.7.3. Коммуникационные технологии
- 2.7.4. Модели коммуникации

2.7.5. Методы коммуникации
2.7.6. План управления коммуникациями
2.7.7. Управление коммуникациями
2.7.8. Управление персоналом
2.7.9. Основные участники и их роли в проектах
2.7.10. Типы организаций
2.7.11. Организация проекта
2.7.12. Рабочая команда

2.8. Снабжение
2.8.1. Процесс закупок
2.8.2. Планирование
2.8.3. Поиск поставщиков и запрос на тендеры
2.8.4. Заключение контракта
2.8.5. Администрирование договоров
2.8.6. Контракты
2.8.7. Виды контрактов
2.8.8. Ведение переговоров по контракту

2.9. Исполнение, мониторинг, контроль и закрытие
2.9.1. Группы процессов
2.9.2. Осуществление проекта
2.9.3. Наблюдение и контроль проекта
2.9.4. Закрытие проекта

2.10. Профессиональная ответственность
2.10.1. Профессиональная ответственность
2.10.2. Характеристики социальной и профессиональной ответственности
2.10.3. Кодекс этических норм руководителя проекта
2.10.4. Ответственность vs. PMP®
2.10.5. Примеры ответственности
2.10.6. Преимущества геймификации



Модуль 3. Платформы для разработки программного обеспечения

3.1. Введение в разработку приложений
- 3.1.1. Приложения для настольных ПК
- 3.1.2. Язык программирования
- 3.1.3. Интегрированные среды разработки
- 3.1.4. Веб-приложения
- 3.1.5. Мобильные приложения
- 3.1.6. Облачные приложения

3.2. Разработка приложений и графического интерфейса на *Java*
- 3.2.1. Интегрированные среды разработки для *Java*
- 3.2.2. Основные IDE для *Java*
- 3.2.3. Знакомство с платформой разработки *Eclipse*
- 3.2.4. Знакомство с платформой разработки *NetBeans*
- 3.2.5. Модель View Controller для графических пользовательских интерфейсов
- 3.2.7. Создание графического интерфейса в *Eclipse*
- 3.2.6. Дизайн графического интерфейса в *NetBeans*

3.3. Отладка и тестирование в языке *Java*
- 3.3.1. Тестирование и отладка *Java*-программ
- 3.3.2. Отладка в *Eclipse*
- 3.3.3. Отладка в *NetBeans*

3.4. Разработка приложений и графического интерфейса на .NET
- 3.4.1. *Net Framework*
- 3.4.2. Компоненты платформы разработки .NET
- 3.4.3. Визуальная студия .NET
- 3.4.4. Инструменты графического интерфейса .NET
- 3.4.5. Графический интерфейс с *Windows Presentation Foundation*
- 3.4.6. Отладка и компиляция приложения WPF

3.5. Программирование для сетей .NET
- 3.5.1. Введение в сетевое программирование в .NET
- 3.5.2. Запросы и ответы в .NET
- 3.5.3. Использование прикладных протоколов в .NET
- 3.5.4. Безопасность при программировании сетей .NET

3.6. Среды разработки мобильных приложений
- 3.6.1. Мобильные приложения
- 3.6.2. Мобильные приложения *Android*
- 3.6.3. Шаги для разработки *Android*
- 3.6.4. Интегрированная среда разработки *Android Studio*

3.7. Разработка приложений в среде *Android Studio*
- 3.7.1. Установить и запустить Android Studio
- 3.7.2. Запуск приложения *Android*
- 3.7.3. Разработка графического интерфейса в *Android Studio*
- 3.7.4. Запуск действий в *Android Studio*

3.8. Отладка и публикация *Android*-приложений
- 3.8.1. Отладка приложения в *Android Studio*
- 3.8.2. Запоминание приложений в *Android Studio*
- 3.8.3. Публикация приложения в *Google Play*

3.9. Разработка облачных приложений
- 3.9.1. *Облачные вычисления*
- 3.9.2. Уровень *облачной системы:* SaaS, PaaS, IaaS
- 3.9.3. Основные платформы для разработки облаков
- 3.9.4. Библиографические ссылки

3.10. Введение в *облачные платформы* Google
- 3.10.1. Основы *облачной платформы* Google
- 3.10.2. Услуги *облачной платформы* Google
- 3.10.3. Инструменты *облачной платформы* Google

Модуль 4. Вычисления на веб-клиенте

4.1. Введение в HTML
- 4.1.1. Структура документа
- 4.1.2. Цвет
- 4.1.3. Текст
- 4.1.4. Гипертекстовые ссылки
- 4.1.5. Изображения
- 4.1.6. Списки
- 4.1.7. Таблицы

4.1.8. Рамки (*Frames*)
4.1.9. Формы
4.1.10. Специфические элементы для мобильных технологий
4.1.11. Неиспользуемые предметы
4.2. Таблицы веб-стиля (CSS)
4.2.1. Элементы и структура таблицы стилей
4.2.1.1. Создание таблиц стилей
4.2.1.2. Применение стилей. Селекторы
4.2.1.3. Наследование стилей и каскадирование
4.2.1.4. Форматирование страниц с помощью стилей
4.2.1.5. Структура страницы с использованием стилей. Модель коробки
4.2.2. Стили дизайна для различных устройств
4.2.3. Типы таблиц стилей: статические и динамические. Псевдоклассы
4.2.4. Передовая практика использования таблиц стилей
4.3. Введение и история *JavaScript*
4.3.1. Введение
4.3.2. История *JavaScript*
4.3.3. Среда разработки для использования
4.4. Основные понятия веб-программирования
4.4.1. Основной синтаксис *JavaScript*
4.4.2. Примитивные типы данных и операторов
4.4.3. Переменные и области
4.4.4. Текстовые строки и *шаблонные литералы*
4.4.5. Числа и логические значения
4.4.6 Сравнtybz
4.5. Сложные структуры в *JavaScript*
4.5.1. Векторы или *массивы* и объекты
4.5.2. Объединительные
4.5.3. Карты
4.5.4. Разъединительные
4.5.5. Петли
4.6. Функции и объекты
4.6.1. Определение и использование функций
4.6.2. Аргументы
4.6.3. Функции стрелок
4.6.4. Функции обратной связи или *Callback*
4.6.5. Функции высшего порядка
4.6.6. Буквальные объекты
4.6.7. Объект *This*
4.6.8. Объекты как пространства имен: объект *Math* и объект *Date*
4.7. Объектная модель документа (DOM)
4.7.1. Что такое DOM?
4.7.2. Немного истории
4.7.3. Навигация и получение элементов
4.7.4. Виртуальный DOM с помощью JSDOM
4.7.5. Селекторы запросов или *Query Selectors*
4.7.6. Навигация по свойствам
4.7.7. Распределение атрибутов элементам
4.7.8. Создание и изменение узлов
4.7.9. Обновление стиля элементов DOM
4.8. Современная веб-разработка
4.8.1. Поток, управляемый событиями, и *слушатели*
4.8.2. Современные *веб-инструменты* и системы выравнивания
4.8.3. Строгий режим *JavaScript*
4.8.4. И еще немного о функциях
4.8.5. Асинхронные обещания и функции
4.8.6. *Закрытия*
4.8.7. Функциональное программирование
4.8.8. POO в *JavaScript*
4.9. Веб-использования
4.9.1. Введение в удобство использования
4.9.2. Определение удобства использования
4.9.3. Методология веб-дизайна, ориентированного на пользователя
4.9.4. Различия между доступностью и удобством использования
4.9.5. Преимущества и проблемы сочетания доступности и удобства использования
4.9.6. Преимущества и трудности при внедрении полезных веб-сайтов

4.9.7. Методы использования
4.9.8. Анализ требований пользователей
4.9.9. Принципы концептуального дизайна. Прототипирование, ориентированное на пользователя
4.9.10. Руководство по созданию веб-сайтов
4.9.10.1. Рекомендации юзабилити *Якова Нильсена*
4.9.10.2. Рекомендации юзабилити Брюса Тоназзини
4.9.11. Оценка юзабилити
4.10. Доступность веб-сайтов
4.10.1. Введение
4.10.2. Определение веб-доступности
4.10.3. Виды инвалидности
4.10.3.1. Временная или постоянная инвалидность
4.10.3.2. Нарушение зрения
4.10.3.3. Нарушение слуха
4.10.3.4. Двигательные нарушения
4.10.3.5. Неврологические или когнитивные нарушения
4.10.3.6. Трудности, связанные со старением
4.10.3.7. Ограничения, возникающие в связи с окружающей средой
4.10.3.8. Барьеры, препятствующие доступу в интернет
4.10.4. Технические средства и вспомогательные продукты для преодоления барьеров
4.10.4.1. Помощь для слепых людей
4.10.4.2. Помощь для людей со слабым зрением
4.10.4.3. Помощь для людей с дальтонизмом
4.10.4.4. Помощь для людей с ограниченными возможностями слуха
4.10.4.5. Помощь для людей с ограниченными двигательными возможностями
4.10.4.6. Помощь для людей с когнитивными и неврологическими нарушениями
4.10.5. Преимущества и трудности при внедрении полезных веб-сайтов
4.10.6. Правила и стандарты веб-доступности
4.10.7. Органы регулирования веб-доступности
4.10.8. Сравнение норм и стандартов
4.10.9. Руководство по соблюдению нормативных актов и стандартов
4.10.9.1. Описание основных руководящих принципов (изображения, ссылки, видео и т.д.)
4.10.9.2. Рекомендации для доступной навигации
4.10.9.2.1. Восприимчивость
4.10.9.2.2. Оперативность
4.10.9.2.3. Понятность
4.10.9.2.4. Устойчивость
4.10.10. Описание процесса обеспечения соответствия требованиям веб-доступности
4.10.11. Уровни соответствия
4.10.12. Критерии соответствия
4.10.13. Требования к соответствию
4.10.14. Методология оценки доступности веб-сайтов

Модуль 5. Вычисления на веб-серверах

5.1. Введение в программирование сервера: PHP
5.1.1. Основы серверного программирования
5.1.2. Основной синтаксис PHP
5.1.3. Создание HTML-контента с помощью PHP
5.1.4. Среды разработки и тестирования: XAMPP
5.2. Продвинутый PHP
5.2.1. Структуры управления в PHP
5.2.2. Функции в PHP
5.2.3. Работа с *массивами* в PHP
5.2.4. Работа со строками в PHP
5.2.5. Объектно-ориентированный PHP
5.3. Модели данных
5.3.1. Концепция данных. Жизненный цикл данных
5.3.2. Виды данных
5.3.2.1. Основные
5.3.2.2. Регистрационные
5.3.2.3. Динамические

5.4. Реляционная модель
- 5.4.1. Описание
- 5.4.2. Организации и типы организаций
- 5.4.3. Элементы данных. Атрибуты
- 5.4.4. Отношения: типы, подтипы, кардинальность
- 5.4.5. Ключи. Типы ключей
- 5.4.6. Нормализация. Нормальные формы

5.5. Построение логической модели данных
- 5.5.1. Спецификация таблиц
- 5.5.2. Определение столбцов
- 5.5.3. Спецификация паролей
- 5.5.4. Переход к нормальным формам. Зависимости

5.6. Физическая модель данных. Карточки данных
- 5.6.1. Описание карточек данных
- 5.6.2. Типы файлов
- 5.6.3. Режимы доступа
- 5.6.4. Организация файлов

5.7. Доступ к базе данных из PHP
- 5.7.1. Введение в *MariaDB*
- 5.7.2. Работа с базой данных *MariaDB*: язык SQL
- 5.7.3. Доступ к базе данных *MariaDB* через PHP
- 5.7.4. Введение в MySQL
- 5.7.5. Работа с базой данных MySQL: язык SQL
- 5.7.6. Доступ к базе данных MySQL через PHP

5.8. Взаимодействие с клиентом через PHP
- 5.8.1. Формы PHP
- 5.8.2. *Файлы cookies*
- 5.8.3. Управление сессиями

5.9. Архитектура веб-приложений
- 5.9.1. Схема разделения данных приложения и управляющей логики на три отдельных компонента
- 5.9.2. Контроль
- 5.9.3. Модель
- 5.9.4. Обзор

5.10. Введение в веб-сервисы
- 5.10.1. Введение в XML
- 5.10.2. Сервис-ориентированные архитектуры (SOA): веб-сервисы
- 5.10.3. Создание веб-сервисов SOAP и REST
- 5.10.4. Протокол SOAP
- 5.10.5. Протокол REST

Модуль 6. Управление безопасностью

6.1. Безопасность информации
- 6.1.1. Введение
- 6.1.2. Информационная безопасность подразумевает конфиденциальность, целостность и доступность
- 6.1.3. Безопасность - это экономический вопрос
- 6.1.4. Безопасность - это процесс
- 6.1.5. Классификация информации
- 6.1.6. Информационная безопасность включает в себя управление рисками
- 6.1.7. Безопасность связана с элементами управления безопасностью
- 6.1.8. Безопасность является как физической, так и логической
- 6.1.9. Безопасность включает в себя людей

6.2. Специалист по информационной безопасности
- 6.2.1. Введение
- 6.2.2. Информационная безопасность как профессия
- 6.2.3. Сертификация (ISC) 2
- 6.2.4. Стандарт ISO 27001
- 6.2.5. Эффективные методы обеспечения безопасности при управлении ИТ-услугами
- 6.2.6. Модели зрелости для информационной безопасности
- 6.2.7. Другие сертификаты, стандарты и профессиональные ресурсы

6.3. Контроль доступа
- 6.3.1. Введение
- 6.3.2. Требования по контролю доступа
- 6.3.3. Механизмы аутентификации
- 6.3.4. Методы авторизации
- 6.3.5. Учет доступа и аудит
- 6.3.6. Технологии «тройного А»

6.4. Программы, процессы и политики информационной безопасности
- 6.4.1. Введение
- 6.4.2. Программы управления безопасностью
- 6.4.3. Управление рисками
- 6.4.4. Разработка политики безопасности

6.5. Планы обеспечения непрерывной работы бизнеса
- 6.5.1. Введение в PCN
- 6.5.2. Стадии I и II
- 6.5.3. Стадии III и IV
- 6.5.4. Обслуживание PCN

6.6. Процедуры для надлежащей защиты компании
- 6.6.1. DMZ-сети
- 6.6.2. Системы обнаружения вторжений
- 6.6.3. Требования по контролю доступа
- 6.6.4. Научиться у взломщиков: *Honeypot*

6.7. Архитектура безопасности. Профилактика
- 6.7.1. Общий обзор. Деятельность и многоуровневая модель
- 6.7.2. Защита периметра (*Firewalls, WAFs*, IPS и т.д.)
- 6.7.3. Защита конечных точек (оборудование, серверы и услуги)

6.8. Архитектура безопасности. Выявление
- 6.8.1. Обзор обнаружения и мониторинга
- 6.8.2. Журналы, зашифрованное разбиение трафика, запись и Siems
- 6.8.3. Оповещения и разведка

6.9. Архитектура безопасности. Реакция
- 6.9.1. Реакция. Продукты, услуги и ресурсы
- 6.9.2. Управление инцидентами
- 6.9.3. CERTS и CSIRTs

6.10. Архитектура безопасности. Восстановление
- 6.10.1. Устойчивость, понятия, бизнес-требования и стандарты
- 6.10.2. Устойчивость ИТ-решений
- 6.10.3. Управление и руководство кризисными ситуациями

Модуль 7. Безопасность в информационных системах

7.1. Глобальный взгляд на безопасность, криптографию и классический криптоанализ
- 7.1.1. Информационная безопасность: историческая перспектива
- 7.1.2. Но именно понимается под «безопасностью»?
- 7.1.3. История криптографии
- 7.1.4. Суррогатные шифры
- 7.1.5. Исследование кейса: машина Энигма

7.2. Симметричная криптография
- 7.2.1. Введение и основная терминология
- 7.2.2. Симметричное шифрование
- 7.2.3. Режимы работы
- 7.2.4. DES
- 7.2.5. Новый стандарт AES
- 7.2.6. Шифрование в потоке
- 7.2.7. Криптоанализ

7.3. Асимметричная криптография
- 7.3.1. Происхождение криптографии с открытым ключом
- 7.3.2. Основные концепции и работа
- 7.3.3. Алгоритм RSA
- 7.3.4. Цифровые сертификаты
- 7.3.5. Хранение и управление ключами

7.4. Сетевые атаки
- 7.4.1. Сетевые угрозы и атаки
- 7.4.2. Перечисление
- 7.4.3. Перехват трафика: *Снифферы*
- 7.4.4. Атаки на отказ в обслуживании
- 7.4.5. Атаки отравления ARP

7.5. Архитектуры безопасности
- 7.5.1. Традиционная архитектура безопасности
- 7.5.2. *Secure Socket Layer*: SSL
- 7.5.3. Протокол SSH
- 7.5.4. Частные виртуальные сети (VPNs)
- 7.5.5. Механизмы защиты внешних запоминающих устройств
- 7.5.6. Аппаратные механизмы защиты

7.6. Методы защиты системы и разработка безопасного кода
- 7.6.1. Оперативная безопасность
- 7.6.2. Ресурсы и контроль
- 7.6.3. Мониторинг
- 7.6.4. Системы обнаружения вторжений
- 7.6.5. IDS *хоста*
- 7.6.6. IDS сети
- 7.6.7. IDS, основанные на подписи
- 7.6.8. Системы приманок
- 7.6.9. Основные принципы безопасности при разработке кода
- 7.6.10. Управление неисправностями
- 7.6.11. Общественный враг номер 1: переполнение буфера
- 7.6.12. Криптографические боты

7.7. *Ботнеты и спам*
- 7.7.1. Происхождение проблемы
- 7.7.2. Процесс рассылки спама
- 7.7.3. Отправление спама
- 7.7.4. Уточнение списков рассылки
- 7.7.5. Методы защиты
- 7.7.6. Услуги по борьбе со спамом, предлагаемые третьими лицами
- 7.7.7. Тематические исследования
- 7.7.8. Экзотический спам

7.8. Аудит и веб-атаки
- 7.8.1. Сбор информации
- 7.8.2. Техники атак
- 7.8.3. Инструменты

7.9. Вредоносное ПО и вредоносный код
- 7.9.1. Что такое вредоносное ПО?
- 7.9.2. Типы вредоносных ПО
- 7.9.3. Вирус
- 7.9.4. Криптовирус
- 7.9.5. Черви
- 7.9.6. Adware
- 7.9.7. *Spyware*
- 7.9.8. *Hoaxes*
- 7.9.9. *Pishing*
- 7.9.10. Трояны
- 7.9.11. Экономика вредоносных ПО
- 7.9.12. Возможные решения

7.10. Криминалистический анализ
- 7.10.1. Сбор доказательств
- 7.10.2. Анализ доказательств
- 7.10.3. Антикриминалистические техники
- 7.10.4. Изучение практического кейса

Модуль 8. Безопасность в программном обеспечении

8.1. Проблемы в области безопасности в программном обеспечении
- 8.1.1. Введение в проблемы безопасности программного обеспечения
- 8.1.2. Уязвимости и их классификация
- 8.1.3. Защищенные свойства программного обеспечения
- 8.1.4. Ссылки

8.2. Принципы проектирования безопасности программного обеспечения
- 8.2.1. Введение
- 8.2.2. Принципы проектирования безопасности программного обеспечения
- 8.2.3. Типы S-SDLC
- 8.2.4. Безопасность программного обеспечения на этапах S-SDLC
- 8.2.5. Методологии и стандарты
- 8.2.6. Ссылки

8.3. Безопасность жизненного цикла программного обеспечения на этапах разработки требований и проектирования
- 8.3.1. Введение
- 8.3.2. Модели атак
- 8.3.3. Случаи злоупотребления
- 8.3.4. Разработка требований безопасности
- 8.3.5. Анализ риска. Архитектурный
- 8.3.6. Модели проектирования
- 8.3.7. Ссылки

8.4. Безопасность жизненного цикла программного обеспечения на этапах кодирования, тестирования и эксплуатации
- 8.4.1. Введение
- 8.4.2. Тестирование безопасности с учетом рисков
- 8.4.3. Обзор кода
- 8.4.4. Тест на проникновение
- 8.4.5. Операции по обеспечению безопасности
- 8.4.6. Внешний обзор
- 8.4.7. Ссылки

8.5. Приложения для безопасного кодирования I
- 8.5.1. Введение
- 8.5.2. Практики безопасной кодификации
- 8.5.3. Манипуляция и валидация входов
- 8.5.4. Переполнение памяти
- 8.5.5. Ссылки

8.6. Приложения для безопасного кодирования II
- 8.6.1. Введение
- 8.6.2. *Переполнения целых чисел*, ошибки усечения и проблемы с преобразованием типов между целыми числами
- 8.6.3. Ошибки и исключения
- 8.6.4. Приватность и конфиденциальность
- 8.6.5. Привилегированные программы
- 8.6.6. Ссылки

8.7. Безопасность в разработке и в облаке
- 8.7.1. Безопасность в развитии; методология и практика
- 8.7.2. Модели PaaS, IaaS, CaaS и SaaS
- 8.7.3. Безопасность в облаке и для облачных услуг

8.8. Оркестровка и автоматизация безопасности (SOAR)
- 8.9.1. Сложность ручной обработки; необходимость автоматизации задач
- 8.9.2. Продукты и услуги
- 8.9.3. Архитектура SOAR

8.9. Безопасность при дистанционной работе
- 8.9.1. Потребность и сценарии
- 8.9.2. Продукты и услуги
- 8.9.3. Безопасность при дистанционной работе

Модуль 9. Качество и аудит информационных систем

9.1. Введение в системы управления информационной безопасностью (СУИБ)
- 9.1.1. Основные принципы СУИБ
- 9.1.2. Золотые правила **СУИБ**
- 9.1.3. Роль информационного аудита в СУИБ

9.2. Планирование в управлении безопасностью
- 9.2.1. Концепции, связанные с управлением безопасностью
- 9.2.2. Классификация информации: цели, концепции и роли
- 9.2.3. Внедрение политик безопасности: политики, стандарты и процедуры безопасности
- 9.2.4. Управление рисками: принципы и анализ рисков информационных активов

9.3. Основные механизмы защиты информационных активов (I)
- 9.3.1. Обзор основных криптографических инструментов для защиты триады КЦД (конфиденциальность, целостность, доступность)
- 9.3.2. Учет конфиденциальности, анонимности и соответствующих требований к управлению отслеживаемостью пользователей

9.4. Основные механизмы защиты информационных активов (II)
- 9.4.1. Безопасность коммуникаций: протоколы, устройства и архитектуры безопасности
- 9.4.2. Безопасность операционной системы

9.5. Внутренний контроль СУИБ
- 9.5.1. Таксономия средств контроля СУИБ: административные, логические и физические средства контроля
- 9.5.2. Классификация средств контроля в зависимости от способа устранения угрозы: средства контроля для предотвращения, обнаружения и устранения угрозы
- 9.5.3. Внедрение систем внутреннего контроля в СУИБ

9.6. Виды аудита
- 9.6.1. Разница между аудитом и внутренним контролем
- 9.6.2. Внутренний аудит перед внешним аудитом
- 9.6.3. Классификация аудита в зависимости от цели и типа анализа

9.7. Сценарист и сценарий: субъект и объект, защищенные интеллектуальной собственностью
- 9.7.1. Введение в тестирование на проникновение и криминалистический анализ
- 9.7.2. Определение и актуальность понятий *Fingerprinting* и *Footprinting*

9.8. Сканирование уязвимостей и мониторинг сетевого трафика
- 9.8.1. Инструменты для анализа уязвимостей в системах
- 9.8.2. Основные уязвимости в контексте веб-приложений
- 9.8.3. Анализ коммуникационных протоколов

9.9. Процесс информационного аудита
- 9.9.1. Концепция жизненного цикла в разработке систем
- 9.9.2. Мониторинг деятельности и процессов: сбор и обработка доказательств
- 9.9.3. Методология ИТ-аудита
- 9.9.4. Процесс ИТ-аудита
- 9.9.5. Определение основных правонарушений и проступков в контексте ИТ
- 9.9.6. Расследование преступлений, связанных с компьютерами: введение в криминалистический анализ и его связь с компьютерным аудитом

9.10. Планы обеспечения непрерывности бизнеса и аварийного восстановления
- 9.10.1. Определение плана обеспечения непрерывности бизнеса и концепции прерывания бизнеса
- 9.10.2. Рекомендация NIST по планам обеспечения непрерывности бизнеса
- 9.10.3. План аварийного восстановления
- 9.10.4. Процесс разработки плана аварийного восстановления

Модуль 10. Администрирование веб-серверов

10.1. Введение в веб-серверы
- 10.1.1. Что такое веб-сервер?
- 10.1.2. Архитектура и работа веб-сервера
- 10.1.3. Ресурсы и содержание веб-сервера
- 10.1.4. Серверы приложений
- 10.1.5. Прокси-серверы
- 10.1.6. Основные веб-серверы на рынке
- 10.1.7. Статистика использования веб-сервера
- 10.1.8. Безопасность веб-сервера
- 10.1.9. Балансировка нагрузки на веб-серверах
- 10.1.10. Ссылки

10.2. Работа с протоколом HTTP
- 10.2.1. Функционирование и структура
- 10.2.2. Описание запросов или *Request Methods*
- 10.2.3. Коды состояния
- 10.2.4. Заголовки
- 10.2.5. Кодирование содержимого. Кодовые страницы
- 10.2.6. Выполнение HTTP-запросов в Интернете с использованием прокси, *Livehttpheaders* или аналогичного метода, анализ используемого протокола

10.3. Описание многосерверных архитектур
- 10.3.1. 3-слойная модель
- 10.3.2. Устойчивость к сбоям
- 10.3.3. Распределение нагрузки
- 10.3.4. Хранилища состояния сеанса
- 10.3.5. Хранилища кэша

10.4. *Информационные службы Интернета* (IIS)
- 10.4.1. Что такое IIS?
- 10.4.2. История и эволюция IIS
- 10.4.3. Основные преимущества и характеристики IIS и последующие версии
- 10.4.4. Архитектура IIS и последующих версий

10.5. Установка, администрирование и настройка IIS
- 10.5.1. Преамбула
- 10.5.2. Установка *информационных служб Интернета* (IIS)
- 10.5.3. Инструменты администрирования IIS
- 10.5.4. Создание, настройка и администрирование веб-сайтов
- 10.5.5. Установка и обработка расширений в IIS

10.6. Продвинутая безопасность в IIS
- 10.6.1. Преамбула
- 10.6.2. Аутентификация, авторизация и контроль доступа в IIS
- 10.6.3. Настройка защищенного веб-сайта на IIS с помощью SSL
- 10.6.4. Политики безопасности, применяемые в IIS 8.x

10.7. Введение в Apache
- 10.7.1. Что такое Apache?
- 10.7.2. Основные преимуществ Apache
- 10.7.3. Основные характеристики Apache
- 10.7.4. Архитектура

10.8. Установка и настройка Apache
- 10.8.1. Начальная установка Apache
- 10.8.2. Настройка Apache

10.9. Установка и настройка различных модулей в Apache
- 10.9.1. Установка модулей Apache
- 10.9.2. Типы модулей
- 10.9.3. Безопасная настройка Apache

10.10. Продвинутая безопасность
- 10.10.1. Аутентификация, авторизация и контроль доступа
- 10.10.2. Методы аутентификации
- 10.10.3. Безопасная конфигурация Apache с помощью SSL

Модуль 11. Безопасность в онлайн-приложениях

11.1. Уязвимости и проблемы безопасности в онлайн-приложениях
- 11.1.1. Введение в безопасность в онлайн-приложениях
- 11.1.2. Уязвимости безопасности при разработке веб-приложений
- 11.1.3. Уязвимости безопасности пути внедрения веб-приложений
- 11.1.4. Уязвимости безопасности при развертывании веб-приложений
- 11.1.5. Официальные списки уязвимостей безопасности

11.2. Политики и стандарты для обеспечения безопасности онлайн-приложений
- 11.2.1. Основные принципы обеспечения безопасности онлайн-приложений
- 11.2.2. Политика безопасности
- 11.2.3. Система управления информационной безопасностью
- 11.2.4. Жизненный цикл разработки безопасного ПО
- 11.2.5. Стандарты безопасности приложений

11.3. Безопасность при разработке веб-приложений
- 11.3.1. Введение в безопасность веб-приложений
- 11.3.2. Безопасность при разработке веб-приложений

11.4. Тестирование онлайн безопасности веб-приложений
- 11.4.1. Анализ и тестирование безопасности веб-приложений
- 11.4.2. Безопасность при развертывании и производстве веб-приложений

11.5. Безопасность веб-сервисов
- 11.5.1. Введение в безопасность веб-сервисов
- 11.5.2. Функции и технологии обеспечения безопасности веб-сервисов

11.6. Тестирование онлайн безопасности веб-приложений
- 11.6.1. Оценка безопасности веб-сервисов
- 11.6.2. Онлайн-защита. *Брандмауэры* и *шлюзы* XML

11.7. Этический *взлом*, вредоносное ПО и *криминалистика*
- 11.7.1. Этический *взлом*
- 11.7.2. Анализ вредоносного ПО
- 11.7.3. Криминалистический анализ

11.8. Разрешение инцидентов для веб-сервисов
- 11.8.1. Мониторинг
- 11.8.2. Инструменты измерения эффективности
- 11.8.3. Меры сдерживания
- 11.8.4. Анализ коренных причин
- 11.8.5. Проактивное управление проблемами

11.9. Эффективные практики обеспечения безопасности приложений
- 11.9.1. Руководство эффективной практики разработки онлайн-приложений
- 11.9.2. Руководство по передовой практике внедрения онлайн-приложений

11.10. Распространенные ошибки, подрывающие безопасность приложений
- 11.10.1. Распространенные ошибки при разработке
- 11.10.2. Распространенные ошибки в хостинге
- 11.10.3. Распространенные ошибки в производстве

Модуль 12. Программная инженерия

12.1. Введение в программную инженерию и моделирование
- 12.1.1. Природа ПО
- 12.1.2. Уникальная природа *веб-приложений*
- 12.1.3. Программная инженерия
- 12.1.4. Процесс ПО
- 12.1.5. Практика программной инженерии
- 12.1.6. Мифы о ПО
- 12.1.7. Как все началось?
- 12.1.8. Объекто-ориентированные понятия
- 12.1.9. Введение в UML

12.2. Процесс ПО
- 12.2.1. Общая модель процесса
- 12.2.2. Предписывающие модели процессов
- 12.2.3. Специализированные модели процессов
- 12.2.4. Единый процесс

12.2.5. Модели личных и командных процессов
12.2.6. Что такое гибкость?
12.2.7. Что такое гибкая методология разработки?
12.2.8. *Scrum*
12.2.9. Инструментарий гибкой методологии разработки
12.3. Принципы, определяющие практику программной инженерии
12.3.1. Принципы, которыми руководствуется процесс
12.3.2. Принципы, которыми руководствуется практика
12.3.3. Принципы коммуникации
12.3.4. Принципы планирования
12.3.5. Принципы моделирования
12.3.6. Принципы строительства
12.3.7. Принципы развертывания
12.4. Понимание требований
12.4.1. Разработка требований
12.4.2. Закладка фундамента
12.4.3. Выяснение требований
12.4.4. Разработка сценариев использования
12.4.5. Разработка модели требований
12.4.6. Согласование требований
12.4.7. Валидация требований
12.5. Моделирование требований: сценарии, информация и классы анализа
12.5.1. Анализ требований
12.5.2. Моделирование на основе сценариев
12.5.3. UML-модели, предоставляющие сценарий использования
12.5.4. Концепции моделирования данных
12.5.5. Моделирование на основе классов
12.5.6. Диаграммы классов
12.6. Моделирование требований: поток, поведение и модели
12.6.1. Стратегии моделирования требований
12.6.2. Моделирование, ориентированное на поток
12.6.3. Диаграммы состояний
12.6.4. Создание поведенческой модели
12.6.5. Диаграммы последовательности
12.6.6. Диаграммы связи
12.6.7. Образцы для моделирования требований
12.7. Концепции разработки
12.7.1. Разработка в контексте программной инженерии
12.7.2. Процесс разработки
12.7.3. Концепции разработки
12.7.4. Концепции объектно-ориентированной разработки
12.7.5. Модель разработки
12.8. Разработка архитектуры
12.8.1. Архитектура ПО
12.8.2. Архитектурные жанры
12.8.3. Архитектурные стили
12.8.4. Архитектурная разработка
12.8.5. Эволюция альтернативных проектов для архитектуры
12.8.6. Составление схемы архитектуры с использованием потока данных
12.9. Разработка на компонентном уровне и на основе паттернов
12.9.1. Что такое компонент?
12.9.2. Разработка компонентов на основе классов
12.9.3. Реализация разработки на уровне компонентов
12.9.4. Разработка традиционных компонентов
12.9.5. Разработка на основе компонентов
12.9.6. Модели разработки
12.9.7. Разработка ПО, основанного на паттернах
12.9.8. Архитектурные паттерны
12.9.9. Паттерны разработки на уровне компонентов
12.9.10. Паттерны разработки пользовательского интерфейса
12.10. Качество ПО и администрирование проектов
12.10.1. Качество
12.10.2. Качество ПО
12.10.3. Дилемма качества ПО
12.10.4. Достигнуть качественного ПО

12.10.5. Обеспечение качества программного обеспечения
12.10.6. Административный спектр
12.10.7. Персонал
12.10.8. Продукт
12.10.9. Процесс
12.10.10. Проект
12.10.11. Принципы и практики

Модуль 13. Продвинутая программная инженерия

13.1. Введение в гибкую методологию разработки
13.1.1. Модели процессов и методологии
13.1.2. Гибкая методология разработки и гибкие процессы
13.1.3. Манифест гибкой методологии разработки
13.1.4. Некоторые гибкие методологии
13.1.5. Гибкая методология vs. Традиционная
13.2. Scrum
13.2.1. Происхождения и философия Scrum
13.2.2. Ценности Scrum
13.3.2. Поток процессов Scrum
13.2.4. Роли Scrum
13.2.5. Артефакты Scrum
13.2.6. События Scrum
13.2.7. Пользовательские истории
13.2.8. Расширения Scrum
13.2.9. Гибкие сметы
13.2.10. Масштабирование Scrum
13.3. Внешнее программирование
13.3.1. Обоснование и обзор XP
13.3.2. Жизненный цикл XP
13.3.3. Пять основных ценностей
13.3.4. Двенадцать основных практик в XP
13.3.5. Роли участников
13.3.6. Промышленный XP
13.3.7. Критическая оценка XP
13.4. Разработка ПО на основе повторного использования
13.4.1. Повторное использование ПО
13.4.2. Уровни повторного использования кода
13.4.3. Конкретные методы повторного использования
13.4.4. Разработка на основе компонентов
13.4.5. Преимущества и проблемы повторного использования
13.4.6. Планирование повторного использования
13.5. Системная архитектура и шаблоны проектирования ПО
13.5.1. Архитектурное проектирование
13.5.2. Общие архитектурные образцы
13.5.3. Отказоустойчивые архитектуры
13.5.4. Архитектуры распределенных систем
13.5.5. Модели проектирования
13.5.6. Гамма-паттерны
13.5.7. Паттерны проектирования взаимодействия
13.6. Архитектура облачных приложений
13.6.1. Основы *облачных вычислений*
13.6.2. Качество облачных приложений
13.6.3. Архитектурные стили
13.6.4. Модели проектирования
13.7. Тесты ПО: TDD, ATDD и BDD
13.7.1. Проверка и валидация ПО
13.7.2. Тесты ПО
13.7.3. *Test Driven Development* (TDD)
13.7.4. *Acceptance Test Driven Development* (ATDD)
13.7.5. *Behavior Driven Development* (BDD)
13.7.6. BDD и Cucumber
13.8. Совершенствование процессов ПО
13.8.1. Совершенствование процессов ПО
13.8.2. Процесс совершенствования процесса
13.8.3. Модели зрелости
13.8.4. Модель CMMI
13.8.5. CMMI V13.0
13.8.6. CMMI и гибкая методология разработки

13.9. Качество продукта ПО: *SQuaRE*
- 13.9.1. Качество ПО
- 13.9.2. Модели качества продукта ПО
- 13.9.3. Семья ISO/IEC 13.000
- 13.9.4. ISO/IEC 13.010: модель и характеристики качества
- 13.9.5. ISO/IEC 13.0113: качество данных
- 13.9.6. ISO/IEC 13.013.: измерение качества ПО
- 13.9.7. ISO/IEC 13.013., 13.013. у 13.013.: метрики качества программного обеспечения и данных
- 13.9.8. ISO/IEC 13.040: оценка ПО
- 13.9.9. Процесс сертификации

13.10. Введение в *DevOps*
- 13.10.1. Понятие DevOps
- 13.10.2. Главные практики

Модуль 14. Разработка требований

14.1. Введение в разработку требований
- 14.1.1. Значимость требований
- 14.1.2. Понятие требования
- 14.1.3. Измерения требований
- 14.1.4. Уровни и виды требований
- 14.1.5. Характеристики требований
- 14.1.6. Разработка требований
- 14.1.7. Процесс разработки требований
- 14.1.8. *Рамки* для разработки требований
- 14.1.9. Эффективные практики в разработки требований
- 14.1.10. Бизнес-анализ

14.2. Источники требований
- 14.2.1. Сеть требований
- 14.2.2. *Заинтересованные стороны*
- 14.2.3. Требования бизнеса
- 14.2.4. Документ видения и охвата

14.3. Методы выявления требований
- 14.3.1. Выяснение требований
- 14.3.2. Проблемы разработки требований
- 14.3.3. Контексты обнаружения
- 14.3.4. Интервью
- 14.3.5. Наблюдение и «обучение»
- 14.3.6. Этнография
- 14.3.7. *Мастер-классы*
- 14.3.8. *Фокус-группы*
- 14.3.9. Вопросы
- 14.3.10. *Брейнсторминг* и креативные техники
- 14.3.11. Групповые средства
- 14.3.12. Анализ системного интерфейса
- 14.3.13. Анализ документов и «археология»
- 14.3.14. Случай использования и сценарии
- 14.3.15. Прототипы
- 14.3.16. Обратное проектирование
- 14.3.17. Повторное использование требований
- 14.3.18. Эффективные практики элицитации

14.4. Пользовательские требования
- 14.4.1. Люди
- 14.4.2. Случаи использования и пользовательские истории
- 14.4.3. Сценарии
- 14.4.4. Виды сценариев
- 14.4.5. Как обнаружить сценарии?

14.5. Техники прототипирования
- 14.5.1. Прототипирование
- 14.5.2. Прототипы, согласно их охвату
- 14.5.3. Прототипы, согласно темпоральности
- 14.5.4. Точность прототипа
- 14.5.5. Прототипы пользовательского интерфейса
- 14.5.6. Оценка прототипа

14.6. Анализ требований
- 14.6.1. Анализ требований
- 14.6.2. Передовая практика анализа требований
- 14.6.3. Словарь данных
- 14.6.4. Приоритеты требований

14.7. Документация требований
- 14.7.1. Документ спецификации требований
- 14.7.2. Структура и содержание SRS (особой документации к ПО)
- 14.7.3. Документация на естественном языке
- 14.7.4. EARS: *Простой подход к синтаксису требований*
- 14.7.5. Нефункциональные требования
- 14.7.6. Атрибуты и шаблоны в виде таблицы
- 14.7.7. Эффективные практики спецификации

14.8. Проверка и согласование требований
- 14.8.1. Валидация требований
- 14.8.2. Методы проверки требований
- 14.8.3. Согласование требований

14.9. Моделирование и управление требованиями
- 14.9.1. Моделирование требований
- 14.9.2. Перспектива пользователя
- 14.9.3. Перспектива данных
- 14.9.4. Функциональная перспектива или перспектива, ориентированная на поток
- 14.9.5. Поведенческая перспектива
- 14.9.6. Нестабильность требований
- 14.9.7. Процесс управления требованиями
- 14.9.8. Инструменты управления требованиями
- 14.9.9. Передовая практика в управлении требованиями

14.10. Критические системы и формальная спецификация
- 14.10.1. Критические системы
- 14.10.2. Спецификация с учетом риска
- 14.10.3. Формальная спецификация

Модуль 15. Процессы программной инженерии

15.1. Основы программной инженерии
- 15.1.1. Характеристики программного обеспечения
- 15.1.2. Основные процессы в программной инженерии
- 15.1.3. Модели процессов разработки программного обеспечения
- 15.1.4. Стандартная справочная система для процесса разработки программного обеспечения: стандарт ISO/IEC 12207

15.2. Унифицированный процесс разработки программного обеспечения
- 15.2.1. Унифицированный процесс
- 15.2.2. Измерения унифицированного процесса
- 15.2.3. Процесс разработки, управляемый случаями использования
- 15.2.4. Фундаментальные рабочие процессы унифицированных процессов

15.3. Планирование в контексте гибкой разработки программного обеспечения
- 15.3.1. Характеристики гибкой разработки программного обеспечения
- 15.3.2. Различные временные горизонты планирования в гибкой методологии разработки
- 15.3.3. Рамки разработки по гибкому методу разработки Scrum и планирование временных горизонтов
- 15.3.4. Пользовательские истории как единица планирования и оценки
- 15.3.5. Общие методы получения оценки
- 15.3.6. Шкалы для интерпретации оценок
- 15.3.7. *Покер-планирование*
- 15.3.8. Общие типы планирования: планирование поставок и планирование итераций

15.4. Стили разработки распределенного программного обеспечения и сервис-ориентированные программные архитектуры
- 15.4.1. Модели коммуникации в распределенных программных системах
- 15.4.2. Промежуточный уровень *ПО промежуточного слоя*
- 15.4.3. Архитектурные паттерны для распределенных систем
- 15.4.4. Общий процесс проектирования программных услуг
- 15.4.5. Аспекты проектирования программных услуг
- 15.4.6. Состав услуг
- 15.4.7. Архитектура веб-сервисов
- 15.4.8. Компоненты инфраструктуры и SOA

15.5. Введение в разработку программного обеспечения на основе моделей
- 15.5.1. Понятие модели
- 15.5.2. Разработка программного обеспечения на основе моделей
- 15.5.3. Система разработки на основе моделей MDA
- 15.5.4. Элементы модели трансформации

15.6. Проектирование графического пользовательского интерфейса
- 15.6.1. Принципы проектирования пользовательского интерфейса
- 15.6.2. Архитектурные паттерны проектирования интерактивных систем: Model-View-Controller (MVC)
- 15.6.3. Пользовательский опыт (UX *User Experience*)
- 15.6.4. Разработка, ориентированная на пользователя
- 15.6.5. Анализ и процесс проектирования графического пользовательского интерфейса
- 15.6.6. Удобство пользовательских интерфейсов
- 15.6.7. Доступность пользовательских интерфейсов

15.7. Разработка веб-приложений
- 15.7.1. Характеристики веб-приложений
- 15.7.2. Пользовательский интерфейс веб-приложения
- 15.7.3. Разработка навигации
- 15.7.4. Основной протокол взаимодействия для веб-приложений
- 15.7.5. Архитектурные стили для веб-приложений

15.8. Стратегии и методы тестирования программного обеспечения и факторы качества программного обеспечения
- 15.8.1. Стратегии тестирования
- 15.8.2. Разработка пробных кейсов
- 15.8.3. Компромиссы между затратами и качеством
- 15.8.4. Модели качества
- 15.8.5. Нормативная семья ISO/IEC 25000 (*SQuaRE*)
- 15.8.6. Модель качества продукта (ISO 2501n)
- 15.8.7. Модель качества данных (ISO 2501n)
- 15.8.8. Управление качеством ПО

15.9. Введение в метрики в программной инженерии
- 15.9.1. Основные понятия: меры, метрики и показатели
- 15.9.2. Виды метрик в программной инженерии
- 15.9.3. Процесс измерения
- 15.9.4. ISO 250215. Внешние метрики и качество использования
- 15.9.5. Объектно-ориентированные метрики

15.10. Сопровождение и реинжиниринг программного обеспечения
- 15.10.1. Процесс технического обслуживания
- 15.10.2. Стандартная схема процесса технического обслуживания. ISO/EIEC 115.64
- 15.10.3. Модель процесса реинжиниринга программного обеспечения
- 15.10.4. Обратное проектирование

Модуль 16. Системная интеграция

16.1. Введение в информационные системы предприятия
- 16.1.1. Введение в информационные системы предприятия
- 16.1.2. Что такое информационная система?
- 16.1.3. Измерения информационных систем
- 16.1.4. Бизнес-процессы и информационные системы
- 16.1.5. Отдел SI/TI

16.2. Возможности и потребности в информационных системах в бизнесе
- 16.2.1. Организации и информационные системы
- 16.2.2. Характеристики организаций
- 16.2.3. Влияние информационных систем на предприятие
- 16.2.4. Информационные системы для конкурентного преимущества
- 16.2.5. Использование систем в деловом администрировании и управлении

16.3. Основные концепции информационных систем и технологий
- 16.3.1. Данные, информация и знания
- 16.3.2. Информационные системы и технологии
- 16.3.3. Технологические компоненты
- 16.3.4. Классификация и типы информационных систем
- 16.3.5. Архитектуры, основанные на услугах и бизнес-процессах
- 16.3.6. Формы системной интеграции

16.4. Системы для интегрированного управления ресурсами предприятия
- 16.4.1. Потребности бизнеса
- 16.4.2. Интегрированная информационная система предприятия
- 16.4.3. Приобретение vs. Развитие
- 16.4.4. Внедрение ERP
- 16.4.5. Последствия для управления
- 16.4.6. Основные поставщики ERP

16.5. Информационные системы управления цепочками поставок и взаимоотношениями с клиентами
- 16.5.1. Определение цепочки поставок
- 16.5.2. Эффективное управление цепочками поставок
- 16.5.3. Роль информационных систем
- 16.5.4. Решения для управления цепью поставок
- 16.5.5. Управление цепью поставок
- 16.5.6. Роль информационных систем
- 16.5.7. Внедрение CRM-системы
- 16.5.8. Критические факторы успеха при внедрении CRM
- 16.5.9. CRM, e-CRM и другие тенденции

16.6. Принятие решений об инвестициях в ИКТ и планирование информационных систем
- 16.6.1. Критерии для принятия решений об инвестициях в ИКТ
- 16.6.2. Увязка проекта с планом управления и бизнес-планом
- 16.6.3. Последствия для менеджмента
- 16.6.4. Перепроектирование бизнес-процессов
- 16.6.5. Решение руководства о методологии внедрения
- 16.6.6. Необходимость планирования информационных систем
- 16.6.7. Цели, участники и сроки
- 16.6.8. Структура и разработка системного плана
- 16.6.9. Мониторинг и обновление

16.7. Соображения безопасности при использовании ИКТ
- 16.7.1. Анализ рисков
- 16.7.2. Безопасность информационных систем
- 16.7.3. Практические советы

16.8. Целесообразность реализации проекта ИКТ и финансовые аспекты в проектах информационных систем
- 16.8.1. Описание и цели
- 16.8.2. Участники EVS
- 16.8.3. Техники и практики
- 16.8.4. Структура затрат
- 16.8.5. Финансовый прогноз
- 16.8.6. Бюджеты

16.9. Бизнес-интеллект (БИ)
- 16.9.1. Что такое бизнес-интеллект?
- 16.9.2. Стратегия и внедрение БИ
- 16.9.3. Настоящее и будущее БИ

16.10. ISO/IEC 12207
- 16.10.1. Что такое "ISO/IEC 12207"?
- 16.10.2. Анализ информационных систем
- 16.10.3. Проектирование информационных систем
- 16.10.4. Внедрение и принятие информационной системы

Модуль 17. Повторное использование программного обеспечения

17.1. Обзор повторного использования ПО
- 17.1.1. Что такое повторное использование ПО?
- 17.1.2. Преимущества и недостатки повторного использования ПО
- 17.1.3. Основные методы повторного использования ПО

17.2. Введение в паттерны проектирования
- 17.2.1. Что такое шаблон проектирования?
- 17.2.2. Каталог основных паттернов проектирования
- 17.2.3. Как использовать паттерны для решения проблем проектирования
- 17.2.4. Как выбрать лучший шаблон дизайна

17.3. Паттерны разработки
- 17.3.1. Паттерны разработки
- 17.3.2. Паттерн *Abstract Factory*
- 17.3.3. Пример применения паттерна *Abstract Factory*
- 17.3.4. Паттерн *Builder*
- 17.3.5. Пример применения *Builder*
- 17.3.6. Паттерн Abstract Factory vs. *Builder*

17.4. Паттерны разработки (II)
- 17.4.1. Паттерн *Factory Method*
- 17.4.2. *Factory Method* vs. *Abstract Factory*
- 17.4.3. Паттерн *Singleton*

17.5. Структурные паттерны
- 17.5.1. Структурные паттерны
- 17.5.2. Паттерн *Adapter*
- 17.5.3. Паттерн *Bridge*

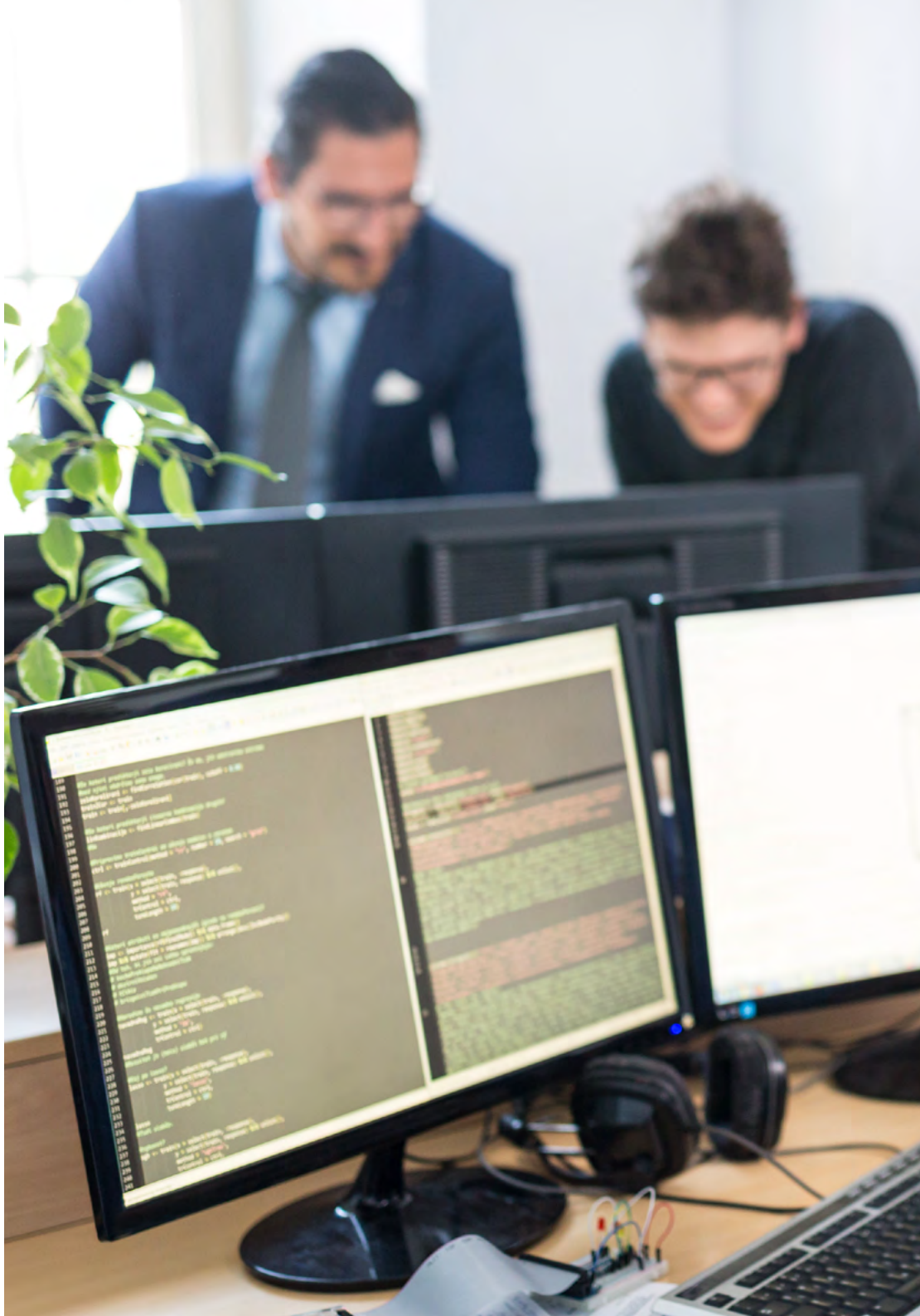
17.6. Структурные паттерны (II)
- 17.6.1. Паттерн *Composite*
- 17.6.2. Паттерн *Decorador*

17.7. Структурные паттерны (III)
- 17.7.1. Паттерн *Facade*
- 17.7.2. Паттерн *Proxy*

17.8. Паттерны поведения
- 17.8.1. Понятие паттернов поведения
- 17.8.2. Паттерн поведения: цепочка обязанностей
- 17.8.3. Паттерн поведения Порядок

17.9. Паттерны поведения (II)
- 17.9.1. Паттерн *Интерпретатор*
- 17.9.2. Паттерн Итератор
- 17.9.3. Паттерн Наблюдатель
- 17.9.4. Паттерн Стратегия

17.10. *Рамки*
- 17.10.1. Понятие *рамок*
- 17.10.2. Разработка с помощью *рамок*
- 17.10.3. Паттерн *Model View Controller*
- 17.10.4. *Рамки* для дизайна графического пользовательского интерфейса
- 17.10.5. *Рамки* для разработки веб-приложений
- 17.10.6. *Рамки* для управления постоянством объектов в базах данных



Модуль 18. Услуги информационных технологий

18.1. Цифровая трансформация (I)
- 18.1.1. Инновации в бизнесе
- 18.1.2. Управление производством
- 18.1.3. Финансовое управление

18.2. Цифровая трансформация (II)
- 18.2.1. Маркетинг
- 18.2.2. Управление человеческими ресурсами
- 18.2.3. Система интеграционной информации

18.3. Исследование кейса
- 18.3.1. Презентация компании
- 18.3.2. Методологии для анализа приобретения ИТ
- 18.3.3. Определение затрат, преимуществ и рисков
- 18.3.4. Экономическая оценка инвестиции

18.4. Руководство и управление ИКТ
- 18.4.1. Определение управления ИТ и информационными системами
- 18.4.2. Разница между руководством и управлением ИСТ
- 18.4.3. Рамки для руководства и управления ИСТ
- 18.4.4. Руководство и управление ИСТ и стандарты

18.5. Корпоративное руководство ИКТ
- 18.5.1. Что значит хорошее корпоративное руководство?
- 18.5.2. Предшественники руководства ИКТ
- 18.5.3. Нормы ISO/IEC 318.00:2008
- 18.5.4. Внедрение хорошего руководства ИКТ
- 18.5.5. Руководство ИКТ и лучшие практики
- 18.5.6. Корпоративное руководство. Обзор и тенденции

18.6. Цели контроля информации и сопутствующих технологий (COBIT)
- 18.6.1. Рамка применения
- 18.6.2. Домен: Планирование и организация
- 18.6.3. Домен: приобретение и внедрение
- 18.6.4. Домен: доставка и поддержка
- 18.6.5. Область: мониторинг и оценка
- 18.6.6. Применение руководства COBIT

18.7. Библиотека инфраструктуры информационных технологий (ITIL)
- 18.7.1. Введение в ITIL
- 18.7.2. Стратегия обслуживания
- 18.7.3. Проектирование услуг
- 18.7.4. Переход на обслуживание
- 18.7.5. Работа службы
- 18.7.6. Улучшение качества обслуживания

18.8. Система управления услугами
- 18.8.1. Основные принципы UNE-ISO/IEC 20000-1
- 18.8.2. Структура нормативной серии UNE-ISO/IEC 20000-1
- 18.8.3. Требования к системе управления сервером SGS
- 18.8.4. Разработка и переход на новые или модифицированные услуги
- 18.8.5. Процессы предоставления услуг
- 18.8.6. Группы процессов

18.9. Система управления активами программного обеспечения
- 18.9.1. Обоснование необходимости
- 18.9.2. Предшественники
- 18.9.3. Презентация нормы 19770
- 18.9.4. Внедрение управления

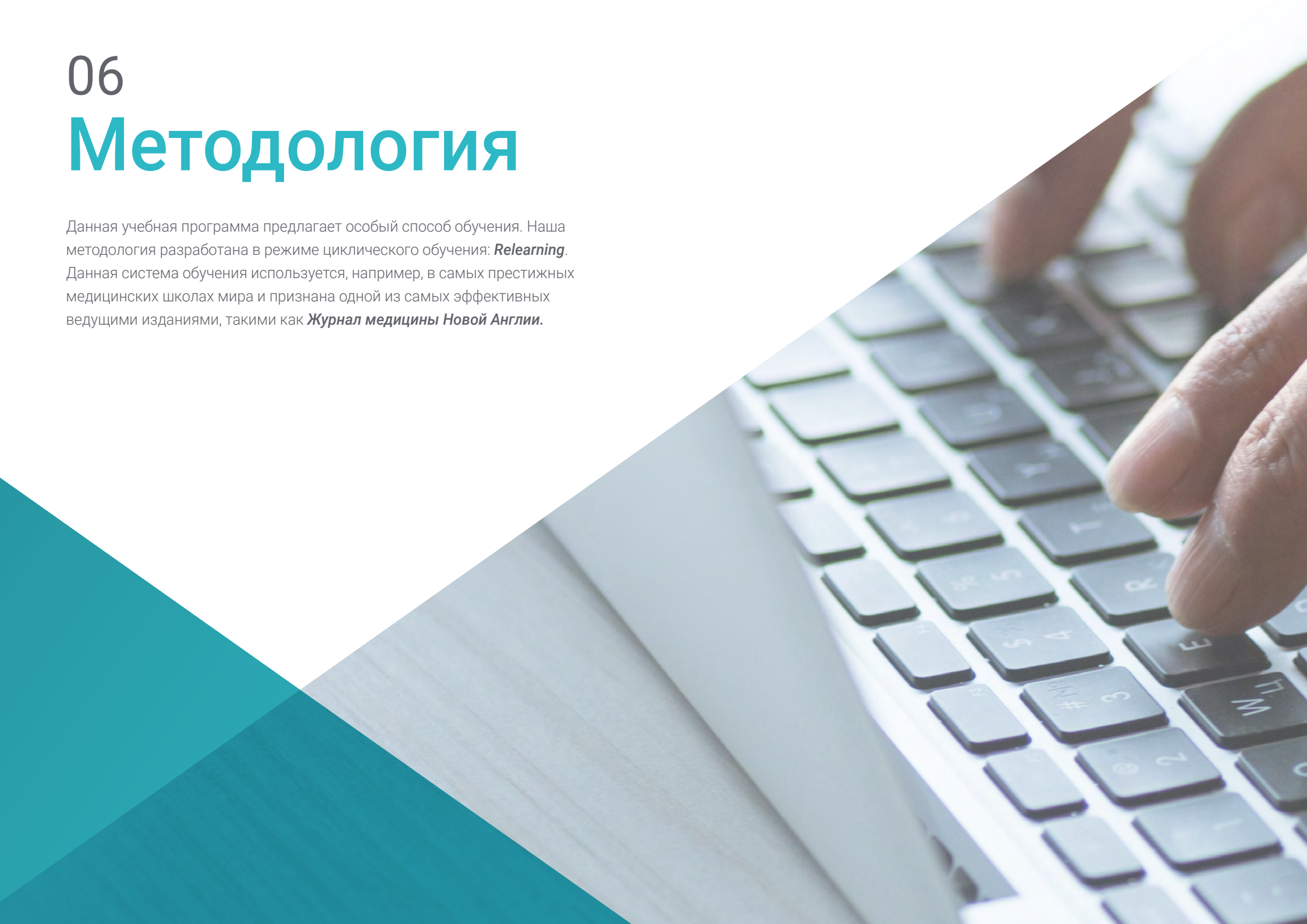
18.10. Управление непрерывной работой бизнеса
- 18.10.1. План непрерывной работы бизнеса
- 18.10.2. Внедрение BCM

Комплексная программа, которая станет основополагающей для вашего профессионального развития"

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*

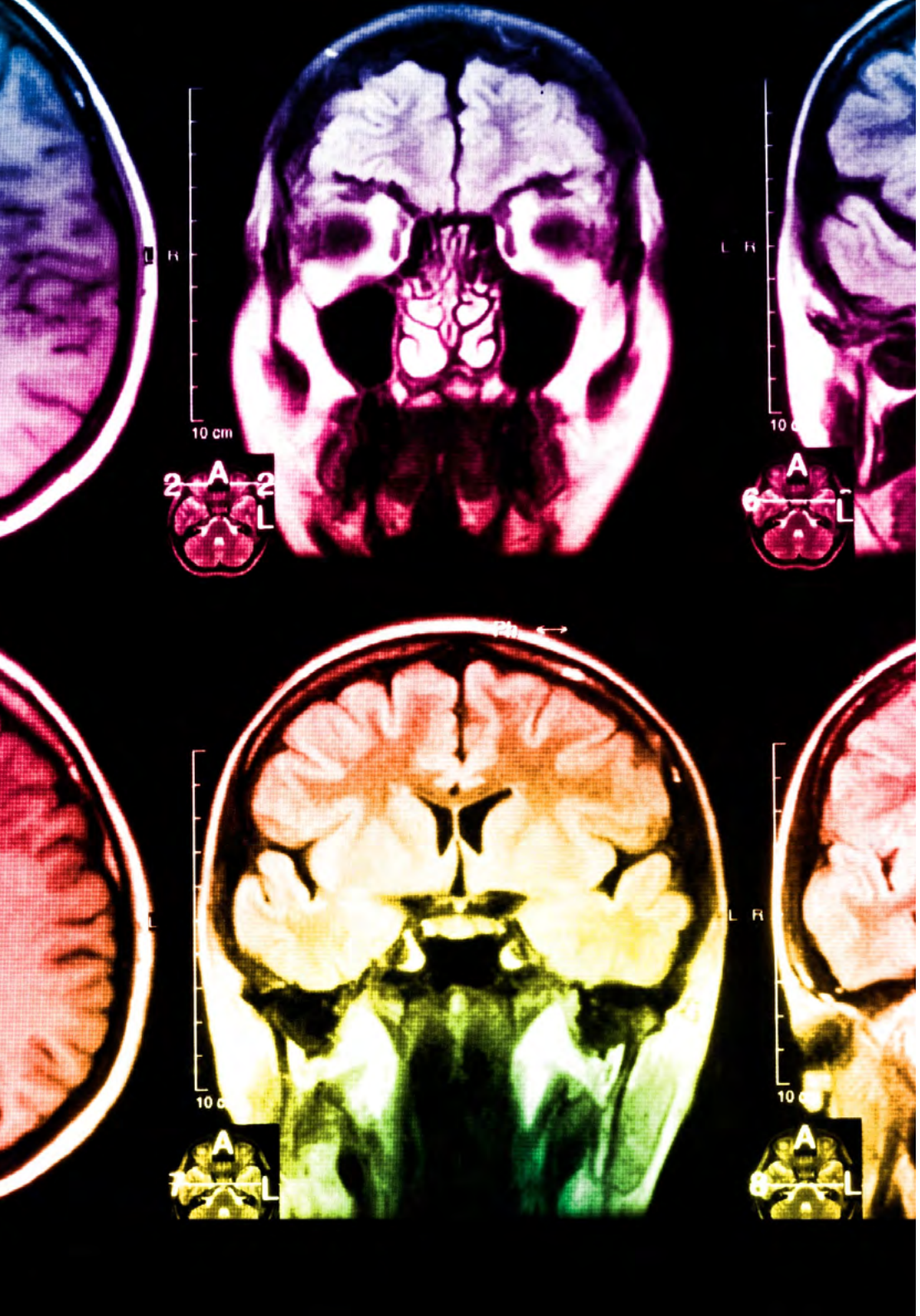
TECH эффективно объединяет метод кейсов с системой 100% онлайн-обучения, основанной на повторении, которая сочетает различные дидактические элементы в каждом уроке.

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.

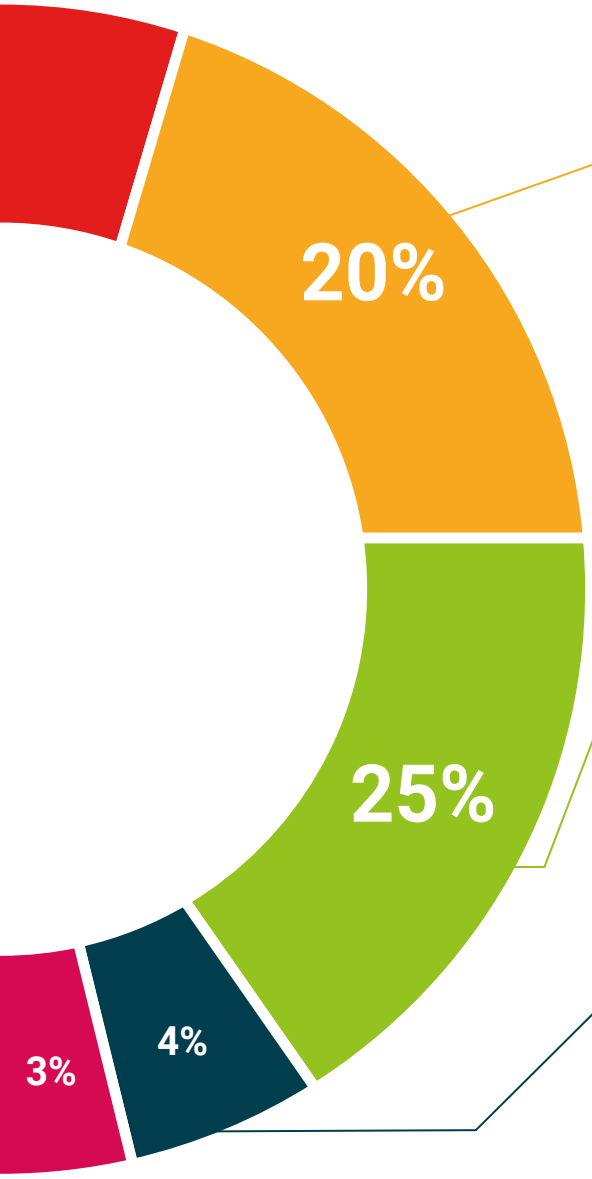
Так называемый метод обучения у эксперта укрепляет знания и память, а также формирует уверенность в наших будущих сложных решениях.

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

Профессиональная магистерская специализация в области программной инженерии гарантирует, помимо самого строгого и современного обучения, получение диплома о прохождении Профессиональной магистерской специализации, выдаваемого TECH Технологическим университетом.

Успешно пройдите эту программу и получите диплом без хлопот, связанных с поездками и оформлением документов”

Данная **Профессиональная магистерская специализация в области программной инженерии** содержит самую полную и современную программу на рынке.

После прохождения аттестации студент получит по почте* с подтверждением получения соответствующий диплом **Профессиональной магистерской специализации,** выданный **TECH Технологическим университетом.**

Диплом, выданный **TECH Технологическим университетом**, подтверждает квалификацию, полученную на Профессиональной магистерской специализации, и соответствует требованиям, обычно предъявляемым биржами труда, конкурсными экзаменами и комитетами по оценке карьеры.

Диплом: **Профессиональная магистерская специализация в области программной инженерии**

Количество учебных часов: **3000 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

ПРОФЕССИОНАЛЬНАЯ МАГИСТЕРСКАЯ СПЕЦИАЛИЗАЦИЯ

в области

Программная инженерия

Данный диплом специализированной программы, присуждаемый Университетом, соответствует 3000 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

Данный диплом всегда должен сопровождаться официально признанным университетским дипломом, выданным компетентным органом для профессиональной деятельности в каждой стране. unique TECH code: AFWOR23S techtitute.com/certificates


Профессиональная магистерская специализация по области Программная инженерия

Общее распределение учебного плана

Курс	Предмет	Часы	Характер
1º	Методологии, разработка и качество в программной инженерии	167	Обязат
1º	Управление проектами прорамного обеспечения	167	Обязат
1º	Платформы для разработки программного обеспечения	167	Обязат
1º	Вычисления на веб-клиенте	167	Обязат
1º	Вычисления на веб-серверах	167	Обязат
1º	Управление безопасностью	167	Обязат
1º	Безопасность в информационных системах	167	Обязат
1º	Безопасность в программном обеспечении	167	Обязат
1º	Качество и аудит информационных систем	167	Обязат
1º	Администрирование веб-серверов	167	Обязат
2º	Безопасность в онлайн-приложениях	167	Обязат
2º	Программная инженерия	167	Обязат
2º	Продвинутая программная инженерия	167	Обязат
2º	Разработка требований	167	Обязат
2º	Процессы программной инженерии	167	Обязат
2º	Системная интеграция	167	Обязат
2º	Повторное использование программного обеспечения	164	Обязат
2º	Услуги информационных технологий	164	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

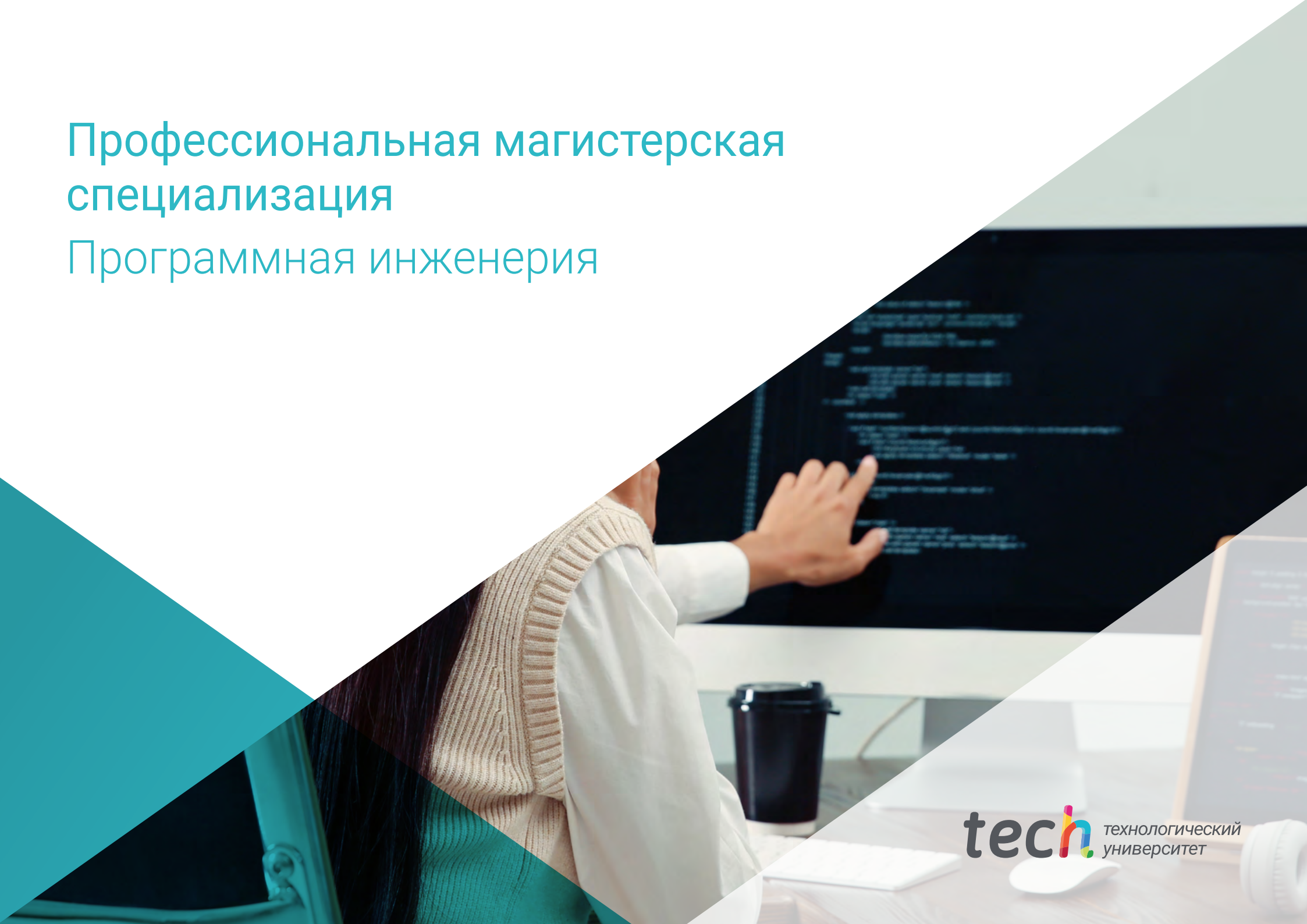
Профессиональная магистерская специализация

Программная инженерия

- Формат: **онлайн**
- Продолжительность: **2 года**
- Учебное заведение: **TECH Технологический университет**
- Режим обучения: **16ч./неделя**
- Расписание: **по своему усмотрению**
- Экзамены: **онлайн**

Профессиональная магистерская специализация

Программная инженерия



tech технологический университет