

Специализированная  
магистратура

Программирование  
с нуля



## Специализированная магистратура Программирование с нуля

- » Формат: онлайн
- » Продолжительность: 12 месяцев
- » Учебное заведение: TECH Global University
- » Аккредитация: 60 ECTS
- » Расписание: по своему усмотрению
- » Экзамены: онлайн

Веб-доступ: [www.techtute.com/ru/information-technology/master-degree/master-artificial-intelligence-cybersecurity](http://www.techtute.com/ru/information-technology/master-degree/master-artificial-intelligence-cybersecurity)

# Оглавление

01

Презентация программы

---

стр. 4

02

Почему стоит учиться в TECH?

---

стр. 8

03

Учебный план

---

стр. 12

04

Цели обучения

---

стр. 24

05

Возможности карьерного роста

---

стр. 30

06

Методика обучения

---

стр. 34

07

Преподавательский состав

---

стр. 44

08

Квалификация

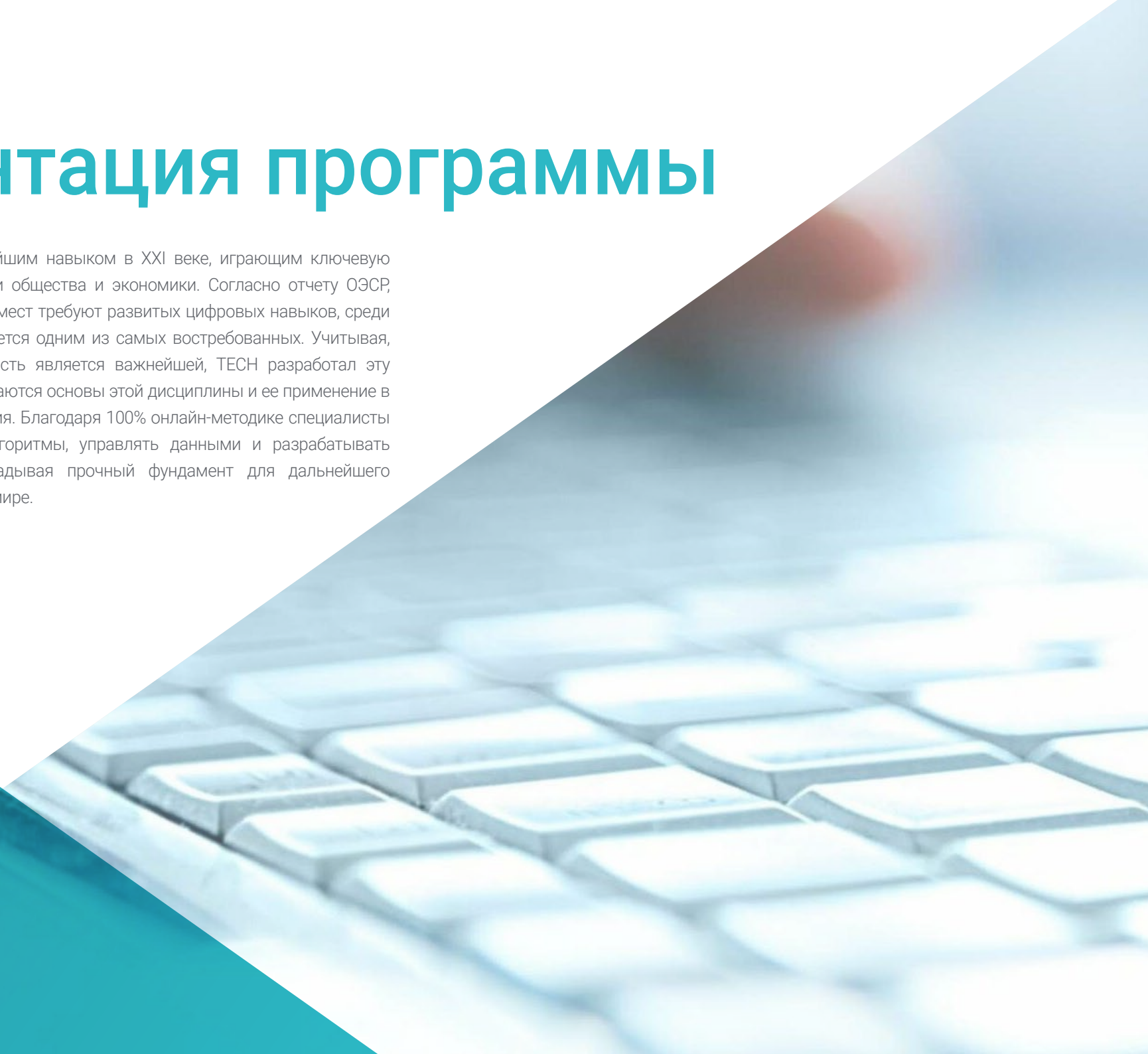
---

стр. 48

# 01

# Презентация программы

Программирование стало важнейшим навыком в XXI веке, играющим ключевую роль в цифровой трансформации общества и экономики. Согласно отчету ОЭСР, более 60% современных рабочих мест требуют развитых цифровых навыков, среди которых программирование является одним из самых востребованных. Учитывая, что в цифровую эпоху эта область является важнейшей, ТЕСН разработал эту программу, в которой рассматриваются основы этой дисциплины и ее применение в языках глобального использования. Благодаря 100% онлайн-методике специалисты узнают, как структурировать алгоритмы, управлять данными и разрабатывать функциональные проекты, закладывая прочный фундамент для дальнейшего продвижения в технологическом мире.



“

Никогда еще навыки программирования не были так доступны. С помощью Специализированной магистратуры от TECH вы сможете освоить такие языки, как Python и JavaScript, находясь в любом месте, благодаря 100% онлайн-методике. Записывайтесь сейчас!"

Программирование лежит в основе цифровой трансформации, которой подвергаются современные общества. От разработки мобильных приложений до автоматизации бизнес-процессов - программирование является фундаментальным навыком, позволяющим пробиться в постоянно меняющейся рабочей среде. В этом смысле профессионалы, освоившие эту область, не только найдут лучшие возможности для роста, но и повысят способность адаптироваться и процветать в мире, где технологии постоянно развиваются.

Осознавая эту потребность, TECH разработал Специализированную магистратуру в области программирования с нуля - комплексную программу, которая позволит получить самые современные знания в этой области. Учебная программа, разработанная отраслевыми экспертами, охватывает все - от основ алгоритмов и структур данных до практического использования таких языков, как *Python*, *JavaScript* и *HTML*. В программу также входят специализированные модули по веб-дизайну, разработке приложений и управлению базами данных. Таким образом, специалисты получают технические знания и разовьют аналитические и творческие навыки решения проблем с помощью программирования. Все это с практическим подходом, который обеспечит реализацию изученных концепций.

Специализируясь в этой области, студенты не только найдут новые возможности в технологическом секторе, но и преуспеют в таких отраслях, как финансы, здравоохранение или логистика, где программирование играет ключевую роль. Кроме того, их обучат руководить проектами развития, автоматизировать процессы и оптимизировать ресурсы, что повысит их трудовую привлекательность и облегчит доступ к высококонкурентным возможностям.

Наконец, 100% онлайн-режим этой программы обеспечит необходимую гибкость для совмещения обучения с другими личными или профессиональными обязанностями. TECH предоставит студентам инновационную платформу, доступную 24 часа в сутки. Кроме того, методика *Relearning* оптимизирует процесс обучения за счет повторения ключевых понятий в различных контекстах, что способствует постепенному и эффективному усвоению материала.

Данная **Специализированная магистратура в области программирования с нуля** содержит самую полную и современную образовательную программу на рынке. Основными особенностями обучения являются:

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



*От базовых алгоритмов до разработки приложений - эта программа поможет вам пройти путь от нуля до эксперта в кратчайшие сроки. Вы станете профессионалом, который нужен технологической индустрии!"*

“

*Войдите в мир программирования с непревзойденным учебным планом. Вы освоите Python, JavaScript и веб-разработку с нуля благодаря самой полной Специализированной магистратуре на рынке”*

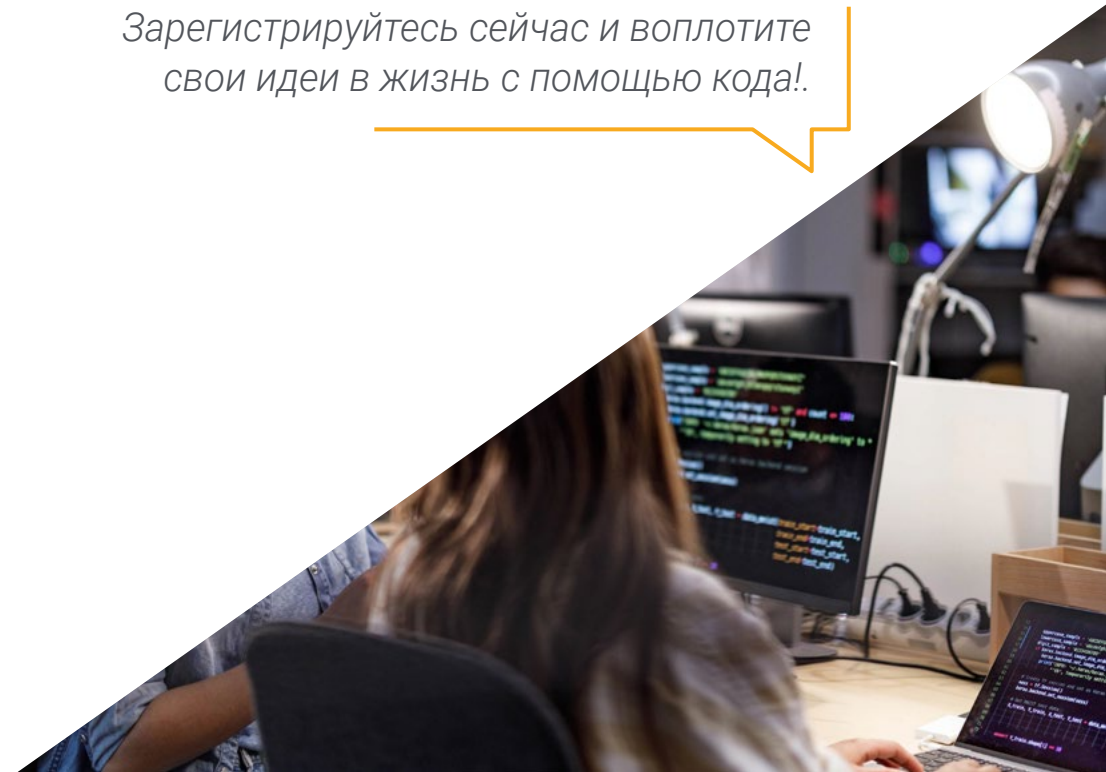
В преподавательский состав входят профессионалы в области программирования, которые привносят в программу свой опыт работы, а также признанные специалисты из ведущих сообществ и престижных университетов.

Мультимедийное содержание программы, разработанное с использованием новейших образовательных технологий, позволит специалисту пройти обучение с учетом ситуации и контекста, то есть в интерактивной среде, которая обеспечит погружение в учебный процесс, запрограммированный на обучение в реальных ситуациях.

В центре внимания этой программы — проблемно-ориентированное обучение, с помощью которого студент должен попытаться разрешить различные ситуации из профессиональной практики, возникающие в течение учебного курса. Для этого специалисту будет помогать инновационная интерактивная видеосистема, созданная признанными и опытными специалистами.

*При поддержке опытных преподавателей, специалистов в области технологий, эта программа поможет вам шаг за шагом овладеть самыми востребованными на рынке языками. Начните сегодня и постройте лучшее будущее!*

*В этой Специализированной магистратуре вы откроете для себя инновационный подход, сочетающий теорию и реальные проекты. Зарегистрируйтесь сейчас и воплотите свои идеи в жизнь с помощью кода!.*



02

# Почему стоит учиться в ТЕСН?

ТЕСН – крупнейший в мире цифровой университет. Имея впечатляющий каталог из более чем 14 000 академических программ, доступных на 11 языках, он позиционируется как лидер по трудоустройству с показателем 99%. Кроме того, университет располагает огромным преподавательским составом, включающим более 6 000 преподавателей с высочайшим международным авторитетом.



“

*Пройдите обучение в крупнейшем в мире цифровом университете и обеспечьте себе профессиональный успех. Будущее начинается в TECH”*

### Лучший онлайн-университет в мире по версии FORBES

Авторитетный журнал Forbes, специализирующийся на бизнесе и финансах, отметил TECH как «лучший онлайн-университет в мире». Об этом недавно сообщили в статье цифровой версии издания, где рассматривается успешный кейс этого учебного заведения, «благодаря его академическому предложению, отбору преподавательского состава и инновационному методу обучения, ориентированному на подготовку профессионалов будущего».

**Forbes**

Лучший  
онлайн-университет  
в мире

Самые полные  
учебные  
программы

### Самые полные учебные программы в университетской среде

TECH предлагает наиболее полные учебные программы, охватывающие как фундаментальные концепции, так и ключевые научные достижения в каждой конкретной области. Кроме того, эти программы постоянно обновляются, чтобы обеспечить студентам передовое академическое образование и наиболее востребованные профессиональные навыки. Таким образом, программы TECH дают студентам значительное преимущество для успешного карьерного роста.

### Лучший международный преподавательский состав

Преподавательский состав TECH включает более 6 000 специалистов с мировым признанием. Среди профессоров, исследователей и топ-менеджеров транснациональных корпораций — Исаяя Ковингтон, тренер "Бостон Селтикс", Магда Романска, главный исследователь Harvard MetaLAB, Игнасио Вистуба, председатель отделения трансляционной молекулярной патологии в MD Anderson Cancer Center, Д.У. Пайн, креативный директор журнала TIME и другие.

Топовые  
международные  
преподаватели

### Уникальный метод обучения

TECH — первый университет, использующий метод *Relearning* во всех своих учебных программах. Это лучшая методология онлайн-обучения, сертифицированная международными агентствами образовательного качества. Кроме того, эта инновационная академическая модель дополняется "Методом кейсов", формируя уникальную стратегию онлайн-обучения. В программу также включены передовые учебные ресурсы, среди которых подробные видеоматериалы, инфографики и интерактивные конспекты.



Наиболее  
эффективная  
методология

### Крупнейший цифровой университет в мире

TECH — крупнейший в мире цифровой университет. Мы — крупнейшее образовательное учреждение с самым обширным цифровым каталогом учебных программ, полностью онлайн, охватывающим большинство областей знаний. Мы предлагаем самое большое количество программ с выдачей дипломов собственного образца, а также официальных программ бакалавриата и программ последипломной подготовки в мире. В общей сложности более 14 000 университетских программ на одиннадцати языках, что делает нас крупнейшим образовательным учреждением в мире.

№1  
в мире

Крупнейший в мире  
онлайн-университет

#### Официальный онлайн-университет NBA

TECH — официальный онлайн-университет NBA. Благодаря нашему партнерству с крупнейшей баскетбольной лигой мы предлагаем студентам эксклюзивные образовательные программы, а также широкий спектр учебных материалов, посвященных бизнесу лиги и другим аспектам спортивной индустрии. Каждая программа имеет уникальный учебный план и включает выдающихся приглашенных лекторов — профессионалов с выдающейся спортивной карьерой, которые поделятся своим опытом по самым актуальным темам.

#### Лидеры по трудоустройству

TECH удалось стать университетом-лидером по трудоустройству. 99% студентов получают работу по специальности в течение одного года после окончания любой из программ университета. Столько же студентов сразу же добиваются карьерного роста. Все это благодаря методологии обучения, эффективность которой основана на приобретении практических навыков, необходимых для профессионального развития.



#### Google Partner Premier

Американский технологический гигант присвоил TECH статус Google Partner Premier. Эта награда, доступная лишь 3% компаний мира, подчеркивает эффективный, гибкий и адаптированный подход, который этот университет предоставляет своим студентам. Признание не только подтверждает высокий уровень строгости, производительности и инвестиций в цифровую инфраструктуру TECH, но и ставит этот университет среди ведущих технологических компаний мира.



#### Университет, получивший самые высокие оценки от своих студентов

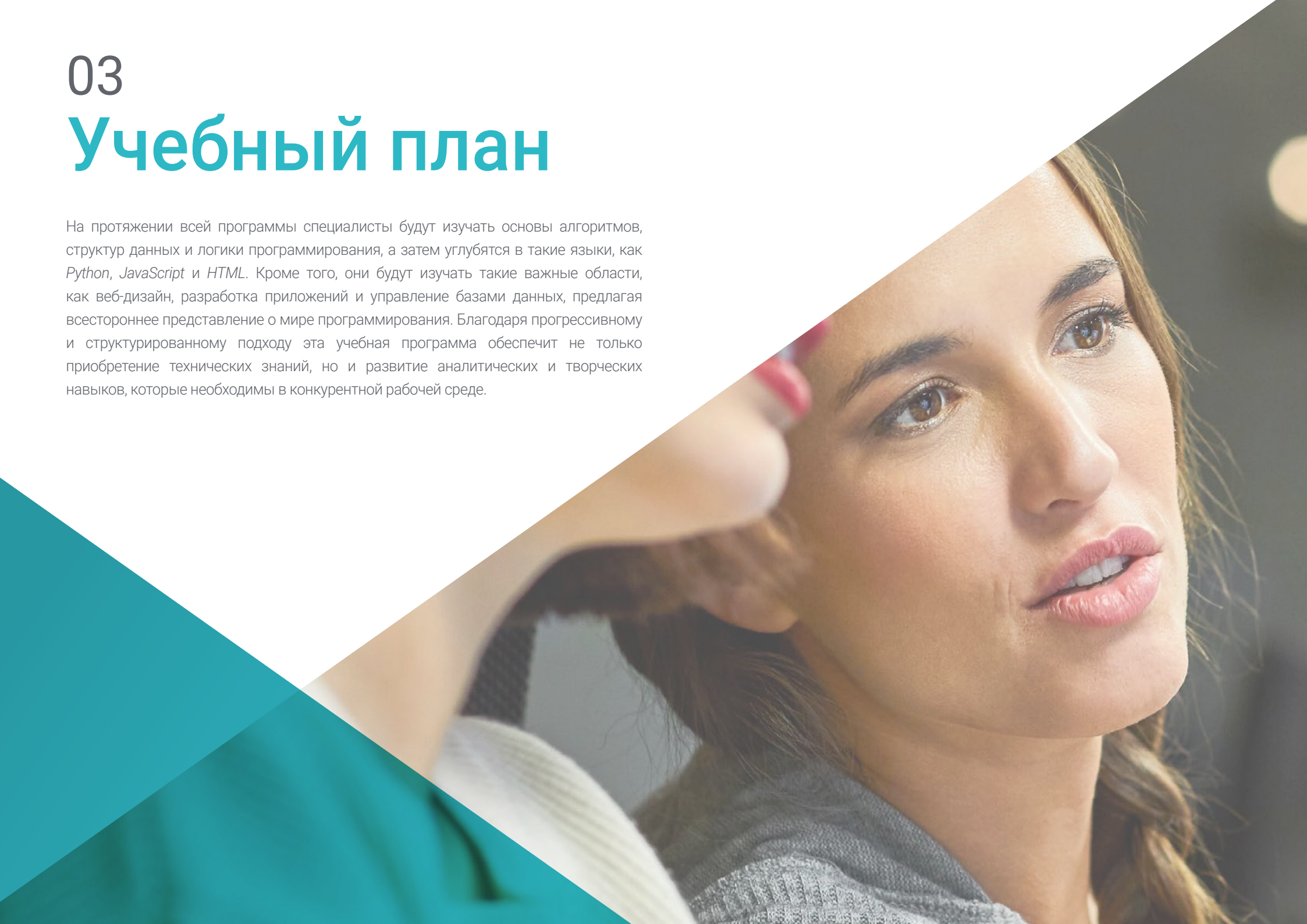
Студенты признали TECH самым высоко оцененным университетом в мире на ведущих платформах с отзывами, отметив его высший рейтинг — 4,9 из 5, основанный на более чем 1 000 рецензиях. Эти результаты укрепляют позиции TECH как ведущего международного университета, отражая его превосходство и положительное влияние образовательной модели.



# 03

## Учебный план

На протяжении всей программы специалисты будут изучать основы алгоритмов, структур данных и логики программирования, а затем углубятся в такие языки, как *Python*, *JavaScript* и *HTML*. Кроме того, они будут изучать такие важные области, как веб-дизайн, разработка приложений и управление базами данных, предлагая всестороннее представление о мире программирования. Благодаря прогрессивному и структурированному подходу эта учебная программа обеспечит не только приобретение технических знаний, но и развитие аналитических и творческих навыков, которые необходимы в конкурентной рабочей среде.



“

*Данный курс последипломного образования - идеальный вариант для начала карьеры в технологическом секторе и приобретения необходимых инструментов, чтобы проявить себя в любой профессиональной области, связанной с программированием”*

## Модуль 1. Программирование и разработка программного обеспечения с нуля

- 1.1. Программное и аппаратное обеспечение. Взаимосвязь и различия
  - 1.1.1. Программное обеспечение
  - 1.1.2. Различия между программным и аппаратным обеспечением
  - 1.1.3. Взаимосвязь между программным и аппаратным обеспечением и программированием
- 1.2. Программирование. Основные вопросы
  - 1.2.1. Программирование
  - 1.2.2. Цели и применение
  - 1.2.3. Программы, исходный код, компиляция и исполнение
  - 1.2.4. Ошибки: Синтаксические ошибки, ошибки выполнения, логические ошибки
- 1.3. Программирование с нуля
  - 1.3.1. Структура программы
  - 1.3.2. Алгоритмы: Структура и примеры
  - 1.3.3. Взаимосвязь между алгоритмами и программами
  - 1.3.4. Решение задач с помощью алгоритмов
- 1.4. Парадигмы и типы языков программирования
  - 1.4.1. Парадигмы программирования
    - 1.4.1.1. Императивные парадигмы
    - 1.4.1.2. Объектно-ориентированная парадигма
    - 1.4.1.3. Функциональная парадигма
    - 1.4.1.4. Декларативная парадигма
  - 1.4.2. Низкоуровневые vs высокоуровневые языки
  - 1.4.3. Компилируемые vs интерпретируемые языки
- 1.5. Перевод языков программирования
  - 1.5.1. Компилятор. Процесс компиляции
  - 1.5.2. Интерпретатор. Процесс интерпретации
  - 1.5.3. Различия между компиляцией и интерпретацией
- 1.6. Биты, двоичные операции и логические элементы
  - 1.6.1. Бит. Двоичное представление
  - 1.6.2. Основные операции с битами: AND, OR, XOR, NOT
  - 1.6.3. Преобразование между двоичной и десятичной системами
  - 1.6.4. Логические элементы: OR, AND, XOR, NOT, NOR и NAND





- 1.7. Разработка алгоритмов с нуля
  - 1.7.1. Разработка алгоритма с нуля
  - 1.7.2. Последовательные, условные и повторяющиеся алгоритмы
  - 1.7.3. Рекурсивные алгоритмы и их сравнение с итерационными алгоритмами
- 1.8. Элементы программы: Компоненты и структура
  - 1.8.1. Ввод и вывод данных
  - 1.8.2. Переменные и константы: Применение
  - 1.8.3. Обработка и манипулирование данными
  - 1.8.4. Основные функции и процедуры с использованием блок-схем
- 1.9. Структуры управления с помощью блок-схем
  - 1.9.1. Структуры управления. Роль в программировании
  - 1.9.2. Условные структуры: Примеры с блок-схемами
  - 1.9.3. Повторяющиеся структуры: Примеры с блок-схемами
- 1.10. Жизненный цикл и модели разработки программного обеспечения
  - 1.10.1. Жизненный цикл программного обеспечения. Стадии
  - 1.10.2. Модели разработки: Каскадная, итеративная и гибкая
  - 1.10.3. Тестирование и сопровождение в разработке программного обеспечения

## Модуль 2. Бэкенд-разработка I: Python с нуля

- 2.1 Python с нуля. Установка
  - 2.1.1. Язык Python. Характеристики
  - 2.1.2. Установка Python на Windows, macOS и Linux
  - 2.1.3. Конфигурация сред разработки: IDE и редакторы кода
  - 2.1.4. Первая программа на Python: "Hello, World!"
- 2.2 Синтаксис и переменные в Python
  - 2.2.1. Структура кода в Python: Отступы
  - 2.2.2. Комментарии в Python
  - 2.2.3. Переменные и типы данных в Python
  - 2.2.4. Арифметические и логические операции в Python
- 2.3 Контроль потока: Условные типы
  - 2.3.1. Структуры управления
  - 2.3.2. Условные операторы: if, elif, else
  - 2.3.3. Тернарная условная операция

- 2.4. Циклы Python
  - 2.4.1. Использование циклов в программировании
  - 2.4.2. Циклы "for" и "while"
  - 2.4.3. Управление потоком в циклах: break и continue
  - 2.4.4. Вложенные циклы
- 2.5. Функции в Python
  - 2.5.1. Функция Python. Полезность
  - 2.5.2. Параметры и аргументы функции
  - 2.5.3. Возвращаемые значения
  - 2.5.4. Предопределенные vs. созданные пользователем функции
- 2.6. Списки и кортежи в Python
  - 2.6.1. Создание и использование списков в Python
  - 2.6.2. Общие операции со списками: Добавлять, удалять, изменять
  - 2.6.3. Кортежи: Различия со списками
  - 2.6.4. Итерация по спискам и кортежам
- 2.7. Словари и множества в Python
  - 2.7.1. Словари: Ключ-значение
  - 2.7.2. Методы работы со словарями
  - 2.7.3. Множества: Применение
  - 2.7.4. Сравнение словарей и множеств
- 2.8. Обработка файлов в Python с нуля
  - 2.8.1. Открытие и закрытие файлов
  - 2.8.2. Способы открытия: Чтение, запись и добавление
  - 2.8.3. Чтение и запись текстовых файлов
- 2.9. Обработка ошибок и исключений
  - 2.9.1. Типы исключений
  - 2.9.2. Использование Try, Except для управления ошибками
  - 2.9.3. Создание персонализированных исключений
- 2.10. Лучшие практики и отладка в Python
  - 2.10.1. Отладка: Назначение
  - 2.10.2. Методы отладки: Использование print и breakpoints
  - 2.10.3. Лучшие практики в написании кода

### Модуль 3. Бэкенд-разработка II – Алгоритмы и структуры данных на Python с нуля

- 3.1. Алгоритмы поиска в структурах данных
  - 3.1.1. Назначение алгоритмов поиска в структурах данных
  - 3.1.2. Линейный поиск: Реализация и примеры использования
  - 3.1.3. Бинарный поиск: Примеры
  - 3.1.4. Сравнение эффективности: Линейный поиск vs. бинарный поиск
- 3.2. Алгоритмы сортировки в структурах данных (I). Основные методы сортировки пузырьком (Bubble Sort) и сортировки вставками (Insertion Sort)
  - 3.2.1. Сортировка пузырьком (Bubble Sort): Реализация и анализ
  - 3.2.2. Сортировка вставками (Insertion Sort): Реализация и примеры использования
  - 3.2.3. Сравнение пузырьковой сортировки (Bubble Sort) и сортировки вставками (Insertion Sort)
- 3.3. Алгоритмы сортировки в структурах данных (II). Продвинутое методы сортировки: Selection Sort, Merge Sort и Quick Sort
  - 3.3.1. Сортировка выбором (Selection Sort). Реализация и анализ
  - 3.3.2. Сортировка слиянием (Merge Sort). Внедрение
  - 3.3.3. Быстрая сортировка (Quick Sort). Внедрение
  - 3.3.4. Сравнение эффективности алгоритмов сортировки
- 3.4. Рекурсивные алгоритмы поиска в структурах данных
  - 3.4.1. Рекурсия: Применение
  - 3.4.2. Прямая и косвенная рекурсия
  - 3.4.3. Рекурсивные алгоритмы: Факториал и числа Фибоначчи. Примеры
- 3.5. Сложность алгоритмов поиска в структурах данных
  - 3.5.1. Алгоритмическая сложность. Измерение эффективности
  - 3.5.2. Нотация Big O
  - 3.5.3. Анализ сложности алгоритмов поиска и сортировки
- 3.6. Продвинутое структуры данных
  - 3.6.1. Деревья: Терминология
  - 3.6.2. Бинарные деревья: Операции
  - 3.6.3. Деревья бинарного поиска (BST): Поиск, вставка и удаление

- 3.7 Алгоритмы на графах
  - 3.7.1 Графы. Представление
  - 3.7.2 Алгоритмы на графах: DFS и BFS
  - 3.7.3 Сравнение DFS и BFS
- 3.8 Динамическое программирование
  - 3.8.1 Динамическое программирование. Применение
  - 3.8.2 Различия между динамическим программированием и рекурсией
  - 3.8.3 Оптимизация с помощью динамического программирования
- 3.9 Методы оптимизации алгоритмов поиска данных
  - 3.9.1 Важность алгоритмической оптимизации поиска данных
  - 3.9.2 Методы оптимизации: Мемоизация (Memoization)
  - 3.9.3 *Divide and Conquer*: Разделяй и властвуй
- 3.10 Другие алгоритмы в Python
  - 3.10.1 Алгоритмы перестановки и комбинирования
  - 3.10.2 Базовые алгоритмы хеширования
  - 3.10.3 Алгоритмы подсчета и генерации подмножеств

## Модуль 4. Бэкенд-разработка III - Flask, создание API и базовой архитектуры с нуля

- 4.1. Flask как бэкенд-фреймворк
  - 4.1.1. Бэкенд-фреймворк. Цели
  - 4.1.2. Flask. Характеристики
  - 4.1.3. Подготовка среды разработки и установка Flask
  - 4.1.4. Первый проект с Flask: "Hello, World!"
- 4.2. HTTP-маршруты и запросы во Flask
  - 4.2.1. Маршрут: Работа в веб-приложении
  - 4.2.2. Методы HTTP во Flask: GET, POST, PUT и DELETE
  - 4.2.3. Маршруты во Flask с параметрами и данными
  - 4.2.4. Организация маршрутов в проекте
- 4.3. Контроллеры и обработка ответов во Flask
  - 4.3.1. Контроллер: Роли и обязанности
  - 4.3.2. Типы ответов во Flask: Текст, JSON и HTML
  - 4.3.3. Создание контроллеров для API во Flask
  - 4.3.4. Операции CRUD в контроллерах
- 4.4. RESTful API
  - 4.4.1. RESTful API. Принципы
  - 4.4.2. Методы и конвенции HTTP в RESTful API
  - 4.4.3. Создание RESTful API во Flask
  - 4.4.4. Разработка API для задач с операциями CRUD
- 4.5. Базы данных и Flask с SQLite
  - 4.5.1. Базы данных в веб-приложениях
  - 4.5.2. Подключение к SQLite в проектах Flask
  - 4.5.3. Создание таблиц и моделей с помощью SQLAlchemy
  - 4.5.4. CRUD-запросы в SQLite для работы с данными
- 4.6. Аутентификация и базовая безопасность в API
  - 4.6.1. Аутентификация и авторизация в API
  - 4.6.2. Создание системы аутентификации пользователей
  - 4.6.3. Использование токенов для аутентификации во Flask
  - 4.6.4. Защита маршрутов и пользовательских данных в API
- 4.7. Валидация данных и обработка ошибок
  - 4.7.1. Обработка ошибок и исключений во Flask
  - 4.7.2. Валидация данных в запросах API
  - 4.7.3. Создание пользовательских сообщений об ошибках
  - 4.7.4. Стратегии валидации и обработки ошибок в CRUD
- 4.8. Структурирование масштабируемых API
  - 4.8.1. Организация и структура масштабируемого проекта Flask
  - 4.8.2. Модулирование и разделение ответственности в API
  - 4.8.3. Базовая оптимизация API для повышения производительности и масштабируемости
  - 4.8.4. Организационная стратегия для крупных проектов

- 4.9. Коммуникация в реальном времени с помощью WebSockets
  - 4.9.1 WebSockets. Области применения
  - 4.9.2 Внедрение WebSockets во Flask с помощью Flask-SocketIO
  - 4.9.3 Коммуникация в реальном времени в приложениях Flask
- 4.10. Развертывание и сопровождение приложений
  - 4.10.1. Подготовка приложений Flask к производству
  - 4.10.2. Развертывание на популярных платформах, таких как Heroku и Render
  - 4.10.3. Использование Docker для развертывания в контейнерах
  - 4.10.4. Мониторинг и обслуживание бэкенд-приложений

## Модуль 5. Объектно-ориентированное программирование и шаблоны проектирования с нуля

- 5.1. Объектно-ориентированное программирование (ООП) с нуля
  - 5.1.1. Объектно-ориентированное программирование
  - 5.1.2. Различия между ООП и структурным программированием
  - 5.1.3. Элементы ООП: Классы, объекты, методы и атрибуты
- 5.2. Классы и объекты в Python
  - 5.2.1. Создание классов и объектов в Python
  - 5.2.2. Атрибуты экземпляра и атрибуты класса
  - 5.2.3. Специальные методы (init, str, repr и т.д.)
  - 5.2.4. Статические методы и методы класса: Использование
- 5.3. Инкапсуляция и абстракция в классах
  - 5.3.1. Инкапсуляция: Полезность
  - 5.3.2. Модификаторы доступа в Python
    - 5.3.2.1. Публичные, защищенные и приватные
  - 5.3.3. Абстракция: Скрытие деталей и повышение простоты
  - 5.3.4. Использование свойств (@property) для контроля доступа
- 5.4. Наследование в Python. Полезность ООП
  - 5.4.1. Наследование: Полезность ООП
  - 5.4.2. Создание производных классов и множественное наследование в Python
  - 5.4.3. Наследуемые методы и атрибуты и перегрузка при наследовании
  - 5.4.4. Иерархии классов и управление базовыми классами
- 5.5. Полиморфизм и перегрузка в Python
  - 5.5.1. Полиморфизм: Duck Typing
  - 5.5.2. Полиморфизм с помощью классов и методов в Python
  - 5.5.3. Перегрузка и переопределение методов в Python
  - 5.5.4. Полиморфизм в разработке программного обеспечения. Применение и преимущества
- 5.6. Связи между классами и проектирование сложных структур
  - 5.6.1. Виды связей: Ассоциация, агрегация и композиция
  - 5.6.2. Различия между агрегацией и композицией Примеры
  - 5.6.3. Проектирование сложных структур с использованием связей между классами
- 5.7. Шаблоны проектирования и принципы SOLID
  - 5.7.1. Значение шаблонов проектирования
  - 5.7.2. Применение шаблонов проектирования в проектах ООП. Преимущества
  - 5.7.3. Классификация шаблонов проектирования
  - 5.7.4. Принципы SOLID и их значение в объектно-ориентированном проектировании
- 5.8. Порождающие шаблоны проектирования
  - 5.8.1. Цели порождающих шаблонов проектирования
  - 5.8.2. Шаблон Singleton
  - 5.8.3. Шаблон Factory и фабричный метод (Factory Method)
  - 5.8.4. Шаблон Builder
- 5.9. Структурные шаблоны проектирования
  - 5.9.1. Цели структурных шаблонов проектирования
  - 5.9.2. Шаблон Adapter
  - 5.9.3. Шаблон Decorator
  - 5.9.4. Шаблон Facade
- 5.10. Поведенческие шаблоны проектирования
  - 5.10.1. Поведенческие шаблоны. Применение
  - 5.10.2. Шаблон Observer
  - 5.10.3. Шаблон Strategy

**Модуль 6. Фронтенд I - HTML и CSS с нуля**

- 6.1. HTML с нуля
  - 6.1.1. HTML. Цель в веб-разработке
  - 6.1.2. Структура HTML-документа: DOCTYPE, <html>, <head>, <body>
  - 6.1.3. Семантические и контентные теги <header>, <nav>, <section>, <footer>
  - 6.1.4. Основные элементы: Абзацы (<p>), списки (<ul>, <ol>), ссылки (<a>), изображения (<img>)
  - 6.1.5. Лучшие практики в области HTML
- 6.2. Текстовые и мультимедийные элементы в HTML
  - 6.2.1. Основные текстовые теги: <p>, заголовки, списки, жирный и курсивный текст
  - 6.2.2. Вставка мультимедиа: Атрибуты <img>, <audio>, <video>
  - 6.2.3. Основные атрибуты для доступности (alt, aria-label)
- 6.3. Формы в HTML
  - 6.3.1. Структура и компоненты форм: <form>, <input>, <label>, <button>
  - 6.3.2. Типы полей ввода: Текст, e-mail, пароль, кнопки отправки
  - 6.3.3. Валидация в HTML5: Валидация полей на стороне клиента
  - 6.3.4. Формы с базовой валидацией. Примеры
- 6.4. CSS с нуля
  - 6.4.1. Язык CSS с нуля: Применение и связь с HTML
  - 6.4.2. Синтаксис CSS: Селекторы, свойства и значения
  - 6.4.3. Применение встроенных, внутренних и внешних стилей
  - 6.4.4. Расширенные селекторы: По типу, классу, ID, псевдоклассы (:hover, :focus)
- 6.5. Блочная модель (Box Model) в CSS
  - 6.5.1. Блочная модель: Значение в CSS
  - 6.5.2. Ключевые свойства: margin, padding, border, width, height
  - 6.5.3. Использование свойства box-sizing для точного контроля блочной модели
  - 6.5.4. Разработка, основанная на блочной модели. Примеры
- 6.6. Шрифты и стили текста в CSS
  - 6.6.1. Свойства цвета и шрифта: color, font-family, font-size
  - 6.6.2. Расширенные стили текста: Жирный, курсив, выравнивание (text-align)
  - 6.6.3. Межстрочный и межбуквенный интервал: line-height, letter-spacing
  - 6.6.4. Единицы измерения в CSS (px, em, rem) и их использование в шрифтах

- 6.7. Верстка макетов с помощью CSS – Flexbox
  - 6.7.1. Flexbox: Назначение
  - 6.7.2. Свойства Flexbox: justify-content, align-items, flex-direction
  - 6.7.3. Расположение и выравнивание элементов в Flexbox
  - 6.7.4. Примеры макетов с Flexbox
- 6.8. CSS Grid и отзывчивый дизайн с помощью CSS
  - 6.8.1. CSS Grid: Строки, столбцы и области
  - 6.8.2. Медиа-запросы: Структура и применение на различных устройствах
  - 6.8.3. Отзывчивый дизайн для мобильных телефонов, планшетов и настольных компьютеров
  - 6.8.4. Настройка шрифтов и жидких макетов в отзывчивом дизайне
- 6.9. Анимация и переходы в CSS
  - 6.9.1. Переходы: Свойство transition, эффекты при наведении (:hover)
  - 6.9.2. Анимация с помощью CSS: Использование @keyframes, базовая анимация
  - 6.9.3. Техники сглаживания переходов и анимации на веб-странице
- 6.10. Веб-доступность в дизайне
  - 6.10.1. Веб-доступность: Важность
  - 6.10.2. Дизайн доступных сайтов. Лучшая практика
  - 6.10.3. ARIA-атрибуты и инструменты проверки доступности

**Модуль 7. Фронтенд II – JavaScript с нуля**

- 7.1. JavaScript с нуля
  - 7.1.1. Язык JavaScript
  - 7.1.2. Интеграция JavaScript в HTML
  - 7.1.3. Первая программа на JavaScript: "Hello, World!"
- 7.2. Переменные и типы данных в JavaScript
  - 7.2.1. Объявление переменных с помощью var, let и const
  - 7.2.2. Типы данных: Числа, строки, булевы значения
  - 7.2.3. Преобразование между типами данных
- 7.3. Управляющие структуры в JavaScript
  - 7.3.1. Условные операторы: if, else if, else
  - 7.3.2. Циклы: for, while, do...while
  - 7.3.3. Switch-case: Альтернатива множественным условным операторам
  - 7.3.4. Break и continue в циклах

- 7.4. Функции в JavaScript
  - 7.4.1. Объявление функций
  - 7.4.2. Параметры, возвращаемые значения и область видимости
  - 7.4.3. Стрелочные функции (=>) и анонимные функции
  - 7.4.4. Обратные вызовы и рекурсия в функциях
- 7.5. Манипуляции с DOM (Document Object Model) с помощью JavaScript
  - 7.5.1. DOM: Структура HTML-документа
  - 7.5.2. Выбор элементов DOM (getElementById, querySelector)
  - 7.5.3. Манипуляция элементами: Изменение текста, стилей и атрибутов
  - 7.5.4. События: click, input, submit и другие
- 7.6. Массивы и объекты в JavaScript
  - 7.6.1. Объявление и использование массивов
  - 7.6.2. Распространенные методы массивов: push, pop, map, filter
  - 7.6.3. Создание и использование объектов
  - 7.6.4. Итерация по массивам и объектам
- 7.7. Обещания и асинхронность в JavaScript
  - 7.7.1. Асинхронность и использование обратных вызовов в JavaScript
  - 7.7.2. Обещание в JavaScript: Создание и управление
  - 7.7.3. Использование async и await в JavaScript
- 7.8. API и fetch в JavaScript
  - 7.8.1. API в JavaScript: Назначение
  - 7.8.2. Потребление REST API с помощью fetch
  - 7.8.3. Обработка ошибок и статус запросов
- 7.9. Локальное хранилище в веб-браузерах
  - 7.9.1. Локальное хранение и хранение сессий в веб-сервисах и приложениях
  - 7.9.2. Хранение и восстановление данных в локальном хранилище
    - 7.9.2. IndexedDB как база данных для браузера
  - 7.9.3. Работа с файлами cookie в JavaScript
  - 7.9.4. Хранение данных в браузере: Примеры
- 7.10. Лучшие практики в JavaScript и инструменты для разработчиков
  - 7.10.1. Код в JavaScript: Лучшие практики
  - 7.10.2. Использование средств разработки браузеров на JavaScript
  - 7.10.3. Отладка и обработка ошибок в JavaScript

## Модуль 8. Фронтенд III - React.js с нуля

- 8.1. React.js с нуля
  - 8.1.1. ReactJS как библиотека для разработки веб-приложений
  - 8.1.2. Компоненты и виртуальный DOM в ReactJS: Архитектура и принцип работы
  - 8.1.3. Установка и настройка с помощью фреймворков NextJS
  - 8.1.4. Первый компонент в React: "Hello, World!"
- 8.2. JavaScript XML или JSX и компоненты в React
  - 8.2.1. JSX: Синтаксис и характеристики
  - 8.2.2. Создание функциональных компонентов в React.js
  - 8.2.3. Использование пропсов для передачи данных между компонентами
  - 8.2.4. Функциональные vs. классовые компоненты для разработки в React.js
- 8.3. Состояние и события в React.js
  - 8.3.1. Состояние компонентов в React
  - 8.3.2. Использование useState для управления состоянием
  - 8.3.3. Обработка событий в React.js: onClick, onChange и др
  - 8.3.4. Примеры управления состояниями и обработки событий в React.js
- 8.4. Жизненный цикл компонентов и эффекты в React
  - 8.4.1. Жизненный цикл компонентов в React
  - 8.4.2. Использование useEffect для работы с эффектами в React
  - 8.4.3. Компоненты со сборкой, обновлением и разборкой в React
- 8.5. Маршрутизация с помощью React Router
  - 8.5.1. Одностраничные приложения (SPA, Single-Page Application) и маршрутизация в веб-приложениях
  - 8.5.2. Установка и настройка React Router
  - 8.5.3. Создание маршрутов и навигации между страницами с помощью React Router
- 8.6. Формы и валидация в React
  - 8.6.1. Создание интерактивных форм в React
  - 8.6.2. Обработка пользовательского ввода и отправка данных в React
  - 8.6.3. Валидация форм в реальном времени в React
- 8.7. Потребление API в React
  - 8.7.1. Потребление API с помощью fetch и axios в React
  - 8.7.2. Обработка состояний загрузки, успеха и ошибки в React
  - 8.7.3. Обновление компонентов на основе данных API в React

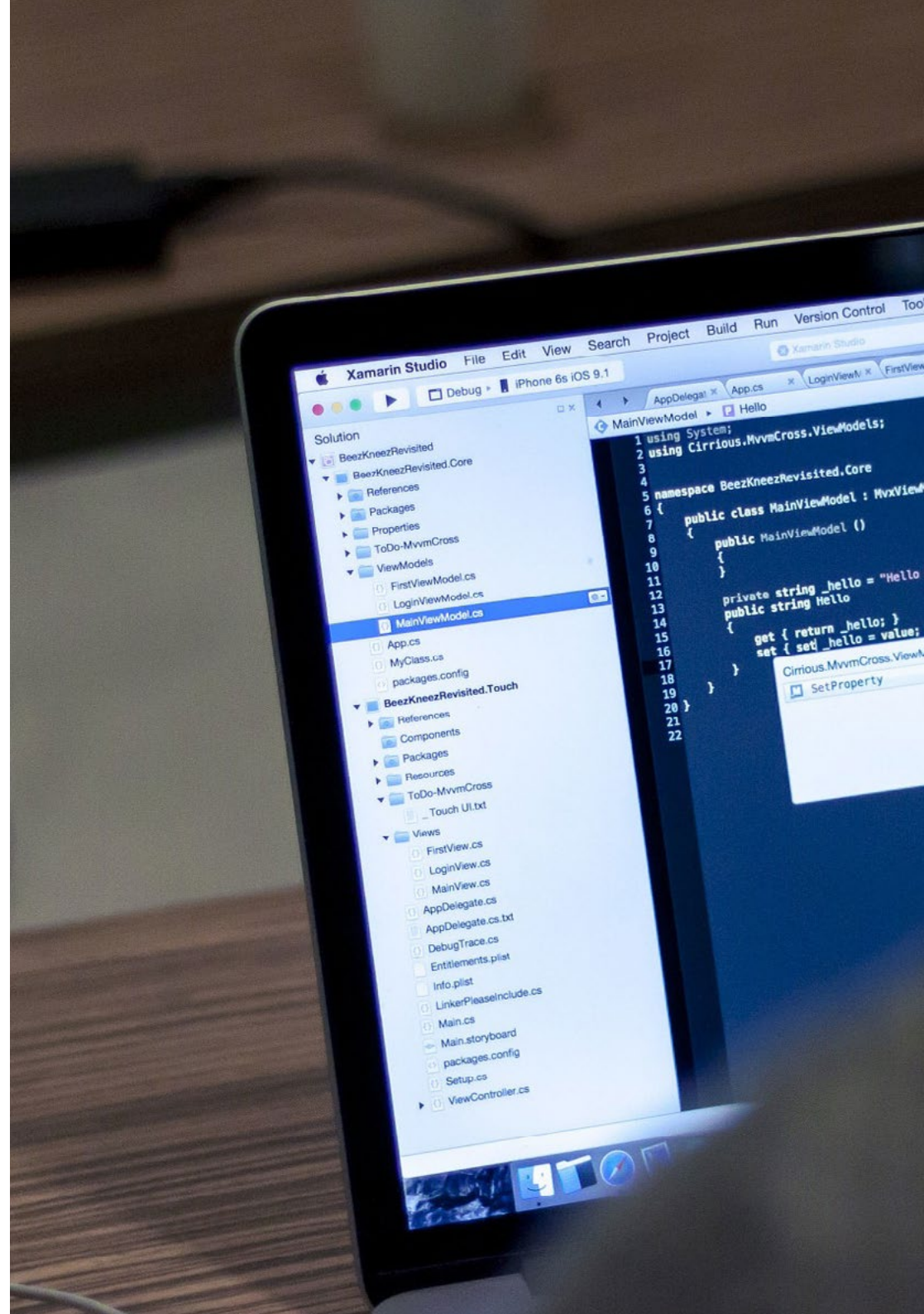
- 8.8. Многократно используемые компоненты и внешние библиотеки в React
  - 8.8.1. Многократно используемые компоненты в React
  - 8.8.2. Создание многократно используемых компонентов в React
  - 8.8.3. Использование внешних библиотек, таких как Material UI и Bootstrap, в React
- 8.9. Глобальное управление состоянием в React
  - 8.9.1. Глобальное управление состоянием с помощью встроенных средств: Context API и пользовательские хуки
  - 8.9.2. Внешние библиотеки для управления данными
  - 8.9.3. Сравнение подходов к глобальному управлению состоянием. Примеры
- 8.10. Развертывание и оптимизация приложений React
  - 8.10.1. Подготовка приложения React к производству
  - 8.10.2. Развертывание на таких платформах, как Netlify и Vercel
  - 8.10.3. Оптимизация производительности: Ленивая загрузка, мемоизация, серверные компоненты и разделение кода
  - 8.10.4. Мониторинг и техническое обслуживание приложений React в рабочей среде. Инструменты и анализ производительности

## Модуль 9. Управление и оптимизация баз данных с нуля

- 9.1. База данных с нуля
  - 9.1.1. Базы данных: Типы
  - 9.1.2. Реляционные vs. нереляционные базы данных
  - 9.1.3. Языки программирования SQL и NoSQL
- 9.2. Моделирование реляционных данных
  - 9.2.1. Реляционная модель базы данных
  - 9.2.2. Таблицы, строки и столбцы в реляционной базе данных
  - 9.2.3. Первичные и внешние ключи: Связи между таблицами
  - 9.2.4. Нормализация: 1NF, 2NF, 3NF
- 9.3. Язык SQL: DML и DDL
  - 9.3.1. SQL: Язык структурированных запросов
  - 9.3.2. Запросы на создание и удаление: CREATE, DROP
  - 9.3.3. Запросы SELECT, INSERT, UPDATE и DELETE
  - 9.3.4. Фильтрация и сортировка данных с помощью SQL
- 9.4. Расширенные SQL-запросы
  - 9.4.1. Join: INNER JOIN и OUTER JOIN
  - 9.4.2. Подзапросы и вложенные запросы
  - 9.4.3. Агрегатные функции SQL: SUM, AVG, COUNT
- 9.5. Базы данных NoSQL и MongoDB
  - 9.5.1. Базы данных NoSQL
  - 9.5.2. Сравнение SQL и NoSQL
  - 9.5.3. MongoDB: База данных документов
  - 9.5.4. Гибкие схемы в NoSQL
- 9.6. Оптимизация баз данных
  - 9.6.1. Важность оптимизации запросов
  - 9.6.2. Использование индексов в реляционных базах данных
  - 9.6.3. Оптимизация баз данных NoSQL
- 9.7. Безопасность в базах данных
  - 9.7.1. Безопасность в базах данных
  - 9.7.2. Шифрование чувствительных данных
  - 9.7.3. Управление пользователями и разрешениями в базах данных
  - 9.7.4. Стратегии защиты баз данных от атак
- 9.8. Масштабируемость баз данных
  - 9.8.1. Масштабируемость в базах данных
  - 9.8.2. Горизонтальное и вертикальное разбиение
  - 9.8.3. Репликация и кластеризация баз данных
- 9.9. Резервное копирование и восстановление данных
  - 9.9.1. Важность резервного копирования баз данных
  - 9.9.2. Автоматические и ручные методы резервного копирования
  - 9.9.3. Восстановление данных в реляционных базах данных и NoSQL
- 9.10. Внедрение баз данных в проекты
  - 9.10.1. Разработка баз данных для реального проекта
  - 9.10.2. Интеграция баз данных с бэкенд-приложениями

**Модуль 10.** Инструменты разработки с нуля: Linux, контроль версий, CI/CD, Docker и Agile-методологии

- 10.1. Linux с нуля
  - 10.1.1. Linux
  - 10.1.2. Различия между Linux и другими операционными системами
  - 10.1.3. Популярные дистрибутивы Linux для разработчиков
  - 10.1.4. Конфигурация и настройка среды разработки
  - 10.1.5. Текстовые редакторы в Linux
- 10.2. Использование терминала в Linux с нуля
  - 10.2.1. Терминал: Применение и принцип работы
  - 10.2.2. Команды просмотра и управления файлами на терминале
  - 10.2.3. Разрешения файлов и каталогов в Linux
  - 10.2.4. Перенаправление команд и использование вводов pipe для оптимизации задач
- 10.3. Контроль версий с помощью Git с нуля
  - 10.3.1. Git: Облачные провайдеры
  - 10.3.2. Создание и управление репозиториями
  - 10.3.3. Рабочий процесс: git init, git add, git commit и git status
  - 10.3.4. Работа с ветками: Создание, слияние и разрешение конфликтов
- 10.4. Командная работа с GitHub с нуля
  - 10.4.1. GitHub: Удаленные репозитории
  - 10.4.2. Подключение локального репозитория к GitHub: git remote. Начальная настройка
  - 10.4.3. Синхронизация с удаленными репозиториями
  - 10.4.4. Pull requests и совместная проверка кода
- 10.5. CI/CD (I) - непрерывная интеграция (CI) с GitHub Actions с нуля
  - 10.5.1. Непрерывная интеграция (CI)
  - 10.5.2. Настройка рабочих процессов в GitHub Actions
  - 10.5.3. Автоматизация тестирования и развертывания
- 10.6. Docker с нуля
  - 10.6.1. Docker и контейнеры
  - 10.6.2. Установка и настройка Docker
  - 10.6.3. Создание и управление контейнерами Docker
  - 10.6.4. Dockerfiles: Создание персонализированных изображений





- 10.7. CI/CD (II) - непрерывная доставка (CD) с помощью Docker и GitHub Actions с нуля
  - 10.7.1. Непрерывная доставка (CD)
  - 10.7.2. Настройка конвейеров CD с помощью Docker и GitHub Actions
  - 10.7.3. Автоматизированное развертывание с помощью Docker Compose
- 10.8. Agile-методологии с нуля (I). Принципы и ценности
  - 10.8.1. Agile-методологии: Принципы
  - 10.8.2. Agile-манифест Основопологающие ценности и принципы
  - 10.8.3. Сравнение с традиционными методологиями: Каскадная vs. Agile
- 10.9. Agile-методологии (II): Scrum с нуля
  - 10.9.1. Scrum и его применение
  - 10.9.2. Ключевые роли в Scrum: Владелец продукта, scrum-мастер и команда разработчиков
  - 10.9.3. Артефакты в Scrum Бэклог продукта, бэклог спринта и рост
  - 10.9.4. События Scrum: Планирование спринта, ежедневный Scrum, обзор спринта и ретроспектива
- 10.10. Agile-методологии (III): Kanban и метрики с нуля
  - 10.10.1. Канбан и его визуальный подход
  - 10.10.2. Ключевые элементы в Kanban: Столбцы, карточки и WIP-лимиты
  - 10.10.3. Agile-метрики: Burnup, burndown charts, velocity и lead time

“ Вы будете обучаться, не выходя из дома и в своем собственном темпе, используя инновационную онлайн-методику и необходимую вам гибкость. Не ждите больше, чтобы сделать карьеру в сфере технологий! Записывайтесь уже сегодня”

# 04

## Цели обучения

Основная цель этой Специализированной магистратуры - дать специалистам необходимые навыки для уверенного и эффективного начала карьеры в мире программирования. На протяжении всей программы они будут развивать глубокое понимание фундаментальных концепций программирования, таких как алгоритмы, структуры данных и логика программирования, которые являются основой для формирования более продвинутых навыков. Кроме того, они овладеют языками программирования, широко используемыми в индустрии (*Python, JavaScript* и *HTML*), что позволит им уверенно справляться с различными технологическими проектами.

“

*Вы приобретете необходимые навыки для руководства проектами разработки, управления базами данных и оптимизации процессов, что повысит вашу профессиональную конкурентоспособность в технологическом секторе”*



## Общие цели

---

- ♦ Овладеть основами программирования и разработки программного обеспечения, определив основные элементы программы, структуры управления и взаимосвязь между программным и аппаратным обеспечением
- ♦ Понять основы языка Python и его базовый синтаксис, создать подходящую среду для разработки бэкенда
- ♦ Приобрести твердое понимание фундаментальных алгоритмов и структур данных
- ♦ Освоить основы разработки бэкенда и его роль в архитектуре программного обеспечения
- ♦ Изучить принципы объектно-ориентированного программирования и их применение для создания гибкого, многократно используемого и сопровождаемого программного обеспечения с использованием Python в качестве языка программирования
- ♦ Рассмотреть фундаментальные концепции HTML и CSS для структурирования и оформления веб-страниц
- ♦ Провести анализ языка JavaScript, от основ до продвинутых методов
- ♦ Ознакомиться с фундаментальными концепциями React.js для создания интерактивных веб-приложений
- ♦ Управлять типами баз данных и языками запросов к ним, как в реляционных, так и в NoSQL-средах
- ♦ Подробно разобрать автоматизацию тестирования, интеграции и развертывания кода





## Конкретные цели

---

### Модуль 1. Программирование и разработка программного обеспечения с нуля

- ♦ Определить и различать программное и аппаратное обеспечение
- ♦ Обосновать важнейшие понятия программирования
- ♦ Знать основную структуру программы
- ♦ Изучить и проанализировать различные парадигмы программирования

### Модуль 2. Бэкенд-разработка I: Python с нуля

- ♦ Освоить характеристики Python
- ♦ Понимать базовую структуру и синтаксис Python
- ♦ Развить навыки управления потоком с помощью условных обозначений
- ♦ Применять циклы для создания повторяющихся циклов в Python

### Модуль 3. Бэкенд-разработка II – Алгоритмы и структуры данных на Python с нуля

- ♦ Внедрить и сравнить типы алгоритмов поиска в структурах данных
- ♦ Проанализировать алгоритмы сортировки пузырьком, вставкой, выбором, слиянием и быстрой сортировки
- ♦ Изучить алгоритмическую сложность и измерение эффективности с помощью нотации Big O
- ♦ Представить графы и выполнить обходы в глубину (DFS) и в ширину (BFS)

#### Модуль 4. Бэкенд-разработка III - Flask, создание API и базовой архитектуры с нуля

- ♦ Разрабатывать RESTful API с помощью Flask
- ♦ Интегрировать базы данных во Flask-приложения
- ♦ Реализовать аутентификацию и безопасность в API
- ♦ Разработать базовую архитектуру бэкенд-приложения с помощью Flask

#### Модуль 5. Объектно-ориентированное программирование и шаблоны проектирования с нуля

- ♦ Определить ключевые понятия объектно-ориентированного программирования, такие как классы, объекты, атрибуты, методы, инкапсуляция, абстракция, наследование и полиморфизм
- ♦ Понимать использование инкапсуляции и абстракции в классах с помощью языка программирования Python
- ♦ Изучить понятия полиморфизма и перегрузки в языке Python и понять их применение и преимущества
- ♦ Определить типы связей между классами, такие как ассоциация, агрегация и композиция

#### Модуль 6. Фронтенд I - HTML и CSS с нуля

- ♦ Определить основную структуру HTML-документа и ее значение для веб-разработки
- ♦ Использовать HTML для организации и представления контента в сети семантическим и доступным способом: Веб-архитектура
- ♦ Применять стилизацию с помощью CSS для улучшения визуального представления элементов
- ♦ Использовать блочную-модель CSS для структурирования и распределения элементов в интерфейсе



**Модуль 7. Фронтенд II – JavaScript с нуля**

- ♦ Понимать синтаксис и типы данных в JavaScript
- ♦ Научиться структурировать код с помощью функций и управляющих структур
- ♦ Манипулировать DOM для взаимодействия с динамическими веб-страницами
- ♦ Работать с API и управлять асинхронностью с помощью обещаний и `async/await`

**Модуль 8. Фронтенд III - React.js с нуля**

- ♦ Понимать, как использовать JSX для создания декларативных интерфейсов
- ♦ Научиться работать с функциональными компонентами, пропсами и жизненными циклами
- ♦ Управлять локальными и глобальными состояниями с помощью современных инструментов, таких как Context API и Redux Toolkit
- ♦ Реализовывать маршрутизацию для создания одностраничных приложений (SPA)

**Модуль 9. Управление и оптимизация баз данных с нуля**

- ♦ Распознавать типы баз данных и их характеристики
- ♦ Понимать и применять реляционную модель данных
- ♦ Развивать навыки работы с SQL для управления базами данных
- ♦ Использовать расширенные SQL-запросы

**Модуль 10. Инструменты разработки с нуля: Linux, контроль версий, CI/CD, Docker и Agile-методологии**

- ♦ Работать с операционной системой Linux на уровне командной строки
- ♦ Освоить использование Git для контроля версий
- ♦ Реализовывать конвейеры непрерывной интеграция и непрерывной доставки (CI/CD)
- ♦ Создавать и управлять контейнерами Docker



*Хотите овладеть языком будущего?  
Данная Специализированная  
магистратура - первый шаг к  
технологическому будущему,  
о котором вы всегда мечтали.  
Записывайтесь прямо сейчас!"*

# 05

## Возможности карьерного роста

Данная программа открывает широкие возможности для карьерного роста в постоянно развивающемся технологическом секторе. По мере того как компании всех отраслей переходят на цифровые технологии, спрос на специалистов с навыками программирования не только растет, но и расширяется, открывая возможности в таких разных областях, как разработка программного обеспечения, управление базами данных, веб-дизайн и автоматизация процессов. По окончании обучения студенты будут отлично подготовлены для работы в технологических компаниях, инновационных *стартапах* и организациях любого сектора, стремящихся внедрить цифровые решения в свою деятельность.





“

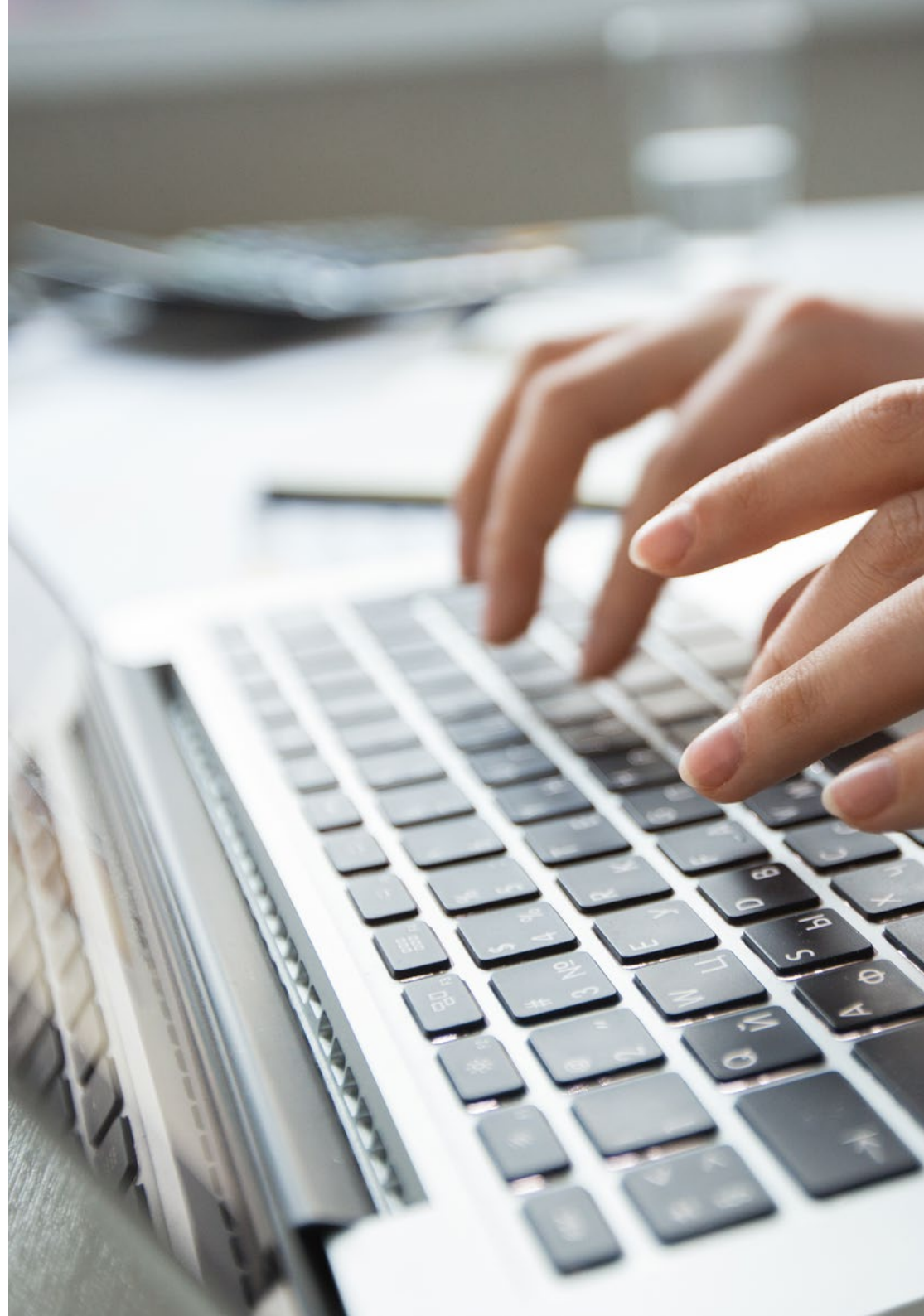
*Вы не только станете экспертом высокого уровня, но и откроете для себя широкий спектр карьерных возможностей, что обеспечит вам успех на постоянно растущем и меняющемся рынке труда”*

### Профиль выпускника

Выпускники будут обладать высокой квалификацией для решения технологических задач будущего. В связи с этим они не только освоят фундаментальные концепции алгоритмов, структур данных и языков программирования, но и получат глубокое понимание инструментов и технологий, наиболее востребованных в отрасли. Обладая солидным опытом в области программирования и разработки приложений, эти специалисты смогут автономно и эффективно справляться с технологическими проектами. Кроме того, они будут отлично разбираться в сложных проблемах, внедрять инновационные решения и управлять системами и базами данных.

*Вы станете универсальным специалистом, обладающим солидной технической подготовкой и стратегическим видением, способным интегрировать программирование в технологические решения, отвечающие потребностям бизнеса и общества.*

- ♦ **Критическое мышление и решение проблем:** Анализировать сложные ситуации и находить инновационные и эффективные решения с помощью программирования
- ♦ **Работа в команде и многопрофильное сотрудничество:** Работать в многопрофильных командах, эффективно сотрудничать с профессионалами из разных областей, такими как дизайнеры, инженеры и аналитики данных, для достижения общих целей в технологических проектах
- ♦ **Тайм-менеджмент и самостоятельность в обучении:** Эффективно распоряжаться временем, укладываясь в сжатые сроки, и самостоятельно и ответственно разрабатывать проекты
- ♦ **Адаптация к новым технологиям:** Управлять новыми инструментами и языками программирования, обеспечивая их постоянное обновление в условиях быстро меняющейся технологической среды



После завершения программы с выдачей диплома собственного образца вы сможете применять свои знания и навыки на следующих должностях:

- 1. Разработчик программного обеспечения:** Отвечает за разработку, создание и техническую поддержку приложений и программного обеспечения в соответствии с потребностями клиента или компании.
- 2. Веб-разработчик:** Управляет созданием, дизайном и технической поддержкой веб-сайтов, обеспечивает их функциональность и оптимизацию на различных устройствах.
- 3. Инженер баз данных:** Отвечает за разработку, внедрение и управление базами данных, гарантируя их эффективность, безопасность и доступность.
- 4. Системный аналитик:** Развивает существующие информационные системы, предлагая улучшения для оптимизации производительности и эффективности процессов.
- 5. Фронтенд-разработчик:** Отвечает за разработку визуальной части приложений и веб-сайтов, улучшая пользовательский опыт.
- 6. Бэкенд-разработчик:** Отвечает за разработку серверной логики и баз данных приложений и веб-сайтов, гарантируя их правильное функционирование.
- 7. Сетевой администратор:** Отвечает за управление и обслуживание сетевых инфраструктур в компании, обеспечивая их стабильность и безопасность.
- 8. Технологический консультант:** Ориентирует и консультирует компании, предоставляет информации о внедрении технологических решений, адаптированных к их потребностям, повышает эффективность и конкурентоспособность бизнеса.
- 9. Разработчик мобильных приложений:** Управляет дизайном, разработкой и технической поддержкой приложений для мобильных устройств, обеспечивая их удобство и производительность.
- 10. Инженер-разработчик программного обеспечения для автоматизации:** Отвечает за проектирование и разработку автоматизированных решений для повышения эффективности бизнес-процессов, используя передовое программирование и технологии.

06

# Методика обучения

TECH – первый в мире университет, объединивший метод **кейс-стади** с **Relearning**, системой 100% онлайн-обучения, основанной на направленном повторении.

Эта инновационная педагогическая стратегия была разработана для того, чтобы предложить профессионалам возможность обновлять свои знания и развивать навыки интенсивным и эффективным способом. Модель обучения, которая ставит студента в центр учебного процесса и отводит ему ведущую роль, адаптируясь к его потребностям и оставляя в стороне более традиционные методологии.



“

*TECH подготовит вас к решению новых задач в условиях неопределенности и достижению успеха в карьере”*

## Студент — приоритет всех программ ТЕСН

В методике обучения ТЕСН студент является абсолютным действующим лицом. Педагогические инструменты каждой программы были подобраны с учетом требований к времени, доступности и академической строгости, которые предъявляют современные студенты и наиболее конкурентоспособные рабочие места на рынке.

В асинхронной образовательной модели ТЕСН студенты сами выбирают время, которое они выделяют на обучение, как они решат выстроить свой распорядок дня, и все это — с удобством на любом электронном устройстве, которое они предпочитают. Студентам не нужно посещать очные занятия, на которых они зачастую не могут присутствовать. Учебные занятия будут проходить в удобное для них время. Вы всегда можете решить, когда и где учиться.

“

*В ТЕСН у вас НЕ будет занятий в реальном времени, на которых вы зачастую не можете присутствовать”*



### Самые обширные учебные планы на международном уровне

TECH характеризуется тем, что предлагает наиболее обширные академические планы в университетской среде. Эта комплексность достигается за счет создания учебных планов, которые охватывают не только основные знания, но и самые последние инновации в каждой области.

Благодаря постоянному обновлению эти программы позволяют студентам быть в курсе изменений на рынке и приобретать навыки, наиболее востребованные работодателями. Таким образом, те, кто проходит обучение в TECH, получают комплексную подготовку, которая дает им значительное конкурентное преимущество для продвижения по карьерной лестнице.

Более того, студенты могут учиться с любого устройства: компьютера, планшета или смартфона.

“

*Модель TECH является асинхронной, поэтому вы можете изучать материал на своем компьютере, планшете или смартфоне в любом месте, в любое время и в удобном для вас темпе”*

## Case studies или метод кейсов

Метод кейсов является наиболее распространенной системой обучения в лучших бизнес-школах мира. Разработанный в 1912 году для того, чтобы студенты юридических факультетов не просто изучали законы на основе теоретических материалов, он также имел цель представить им реальные сложные ситуации. Таким образом, они могли принимать взвешенные решения и выносить обоснованные суждения о том, как их разрешить. В 1924 году он был установлен в качестве стандартного метода обучения в Гарвардском университете.

При такой модели обучения студент сам формирует свою профессиональную компетенцию с помощью таких стратегий, как *обучение действием* (learning by doing) или *дизайн-мышление* (design thinking), используемых такими известными учебными заведениями, как Йель или Стэнфорд.

Этот метод, ориентированный на действия, будет применяться на протяжении всего академического курса, который студент проходит в TECH. Таким образом, они будут сталкиваться с множеством реальных ситуаций и должны будут интегрировать знания, проводить исследования, аргументировать и защищать свои идеи и решения. Все это делается для того, чтобы ответить на вопрос, как бы они поступили, столкнувшись с конкретными сложными событиями в своей повседневной работе.



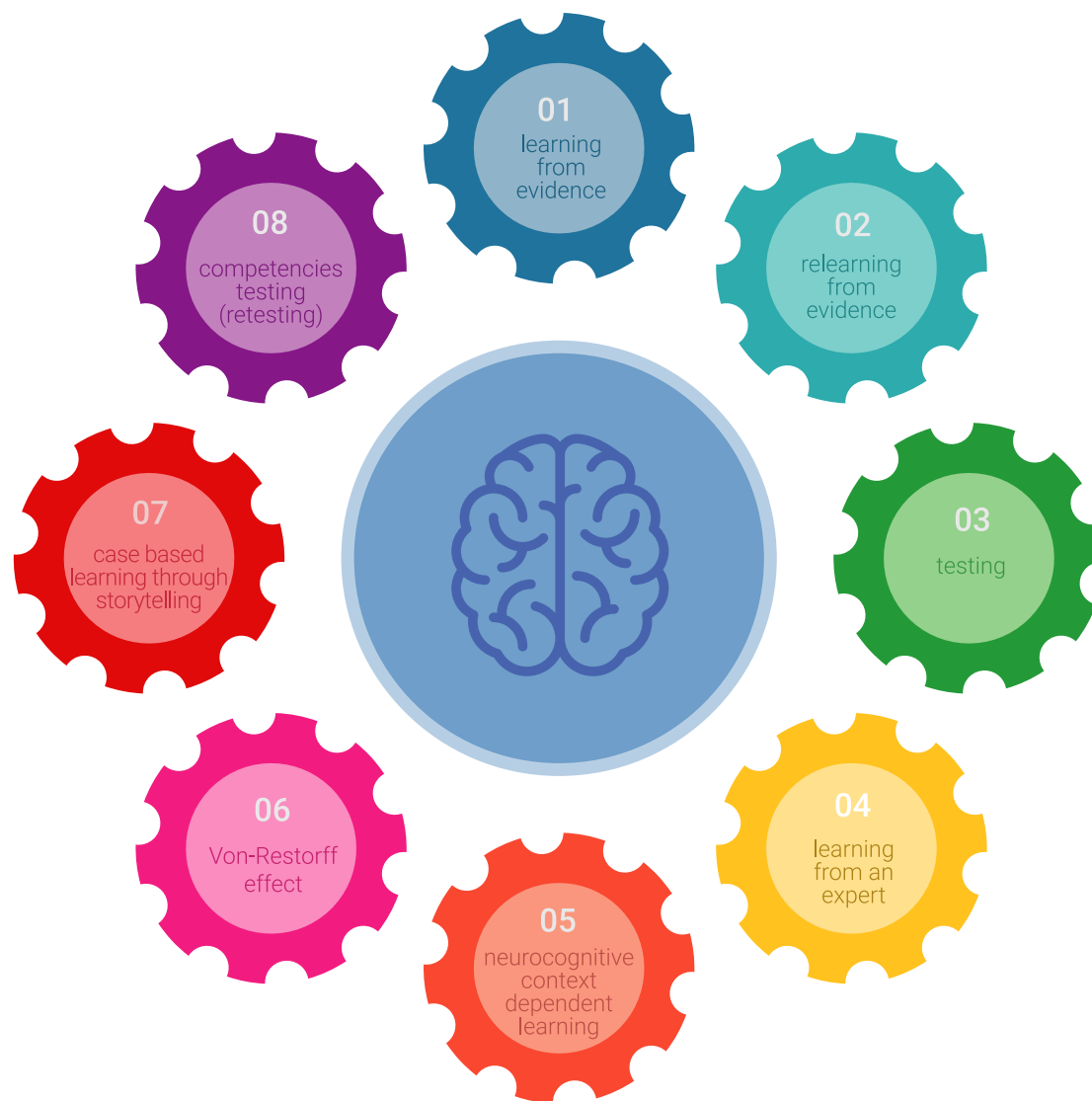
## Метод *Relearning*

В ТЕСН метод кейсов дополняется лучшим методом онлайн-обучения – *Relearning*.

Этот метод отличается от традиционных методик обучения, ставя студента в центр обучения и предоставляя ему лучшее содержание в различных форматах. Таким образом, студент может пересматривать и повторять ключевые концепции каждого предмета и учиться применять их в реальной среде.

Кроме того, согласно многочисленным научным исследованиям, повторение является лучшим способом усвоения знаний. Поэтому в ТЕСН каждое ключевое понятие повторяется от 8 до 16 раз в рамках одного занятия, представленного в разных форматах, чтобы гарантировать полное закрепление знаний в процессе обучения.

*Метод Relearning позволит тебе учиться с меньшими усилиями и большей эффективностью, глубже вовлекаясь в свою специализацию, развивая критическое мышление, умение аргументировать и сопоставлять мнения – прямой путь к успеху.*



## Виртуальный кампус на 100% в онлайн-формате с лучшими учебными ресурсами

Для эффективного применения своей методики ТЕСН предоставляет студентам учебные материалы в различных форматах: тексты, интерактивные видео, иллюстрации, карты знаний и др. Все они разработаны квалифицированными преподавателями, которые в своей работе уделяют особое внимание сочетанию реальных случаев с решением сложных ситуаций с помощью симуляции, изучению контекстов, применимых к каждой профессиональной сфере, и обучению на основе повторения, с помощью аудио, презентаций, анимации, изображений и т.д.

Последние научные данные в области нейронаук указывают на важность учета места и контекста, в котором происходит доступ к материалам, перед началом нового процесса обучения. Возможность индивидуальной настройки этих параметров помогает людям лучше запоминать и сохранять знания в гиппокампе для долгосрочного хранения. Речь идет о модели, называемой *нейрокогнитивным контекстно-зависимым электронным обучением*, которая сознательно применяется в данной университетской программе.

Кроме того, для максимального содействия взаимодействию между наставником и студентом предоставляется широкий спектр возможностей для общения как в реальном времени, так и в отложенном (внутренняя система обмена сообщениями, форумы для обсуждений, служба телефонной поддержки, электронная почта для связи с техническим отделом, чат и видеоконференции).

Этот полноценный Виртуальный кампус также позволит студентам ТЕСН организовывать свое учебное расписание в соответствии с личной доступностью или рабочими обязательствами. Таким образом, студенты смогут полностью контролировать академические материалы и учебные инструменты, необходимые для быстрого профессионального развития.



*Онлайн-режим обучения на этой программе позволит вам организовать свое время и темп обучения, адаптировав его к своему расписанию”*

### Эффективность метода обосновывается четырьмя ключевыми достижениями:

1. Студенты, которые следуют этому методу, не только добиваются усвоения знаний, но и развивают свои умственные способности с помощью упражнений по оценке реальных ситуаций и применению своих знаний.
2. Обучение прочно опирается на практические навыки, что позволяет студенту лучше интегрироваться в реальный мир.
3. Усвоение идей и концепций становится проще и эффективнее благодаря использованию ситуаций, возникших в реальности.
4. Ощущение эффективности затраченных усилий становится очень важным стимулом для студентов, что приводит к повышению интереса к учебе и увеличению времени, посвященному на работу над курсом.

## Методика университета, получившая самую высокую оценку среди своих студентов

Результаты этой инновационной академической модели подтверждаются высокими уровнями общей удовлетворенности выпускников ТЕСН.

Студенты высоко оценивают качество преподавания, учебных материалов, структуру курса и его цели. Благодаря этому университет занял лидирующие позиции в рейтинге global score, став самым высоко оцененным учебным заведением по мнению своих студентов и получив оценку 4,9 из 5.

*Благодаря тому, что ТЕСН идет в ногу с передовыми технологиями и педагогикой, вы можете получить доступ к учебным материалам с любого устройства с подключением к Интернету (компьютера, планшета или смартфона).*

*Вы сможете учиться, пользуясь преимуществами доступа к симулированным образовательным средам и модели обучения через наблюдение, то есть учиться у эксперта (learning from an expert).*



Таким образом, в этой программе будут доступны лучшие учебные материалы, подготовленные с большой тщательностью:



#### Учебные материалы

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

Затем эти материалы переносятся в аудиовизуальный формат, на основе которого строится наш способ работы в интернете, с использованием новейших технологий, позволяющих нам предложить вам отличное качество каждого из источников, предоставленных к вашим услугам.



#### 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.



#### Интерактивные конспекты

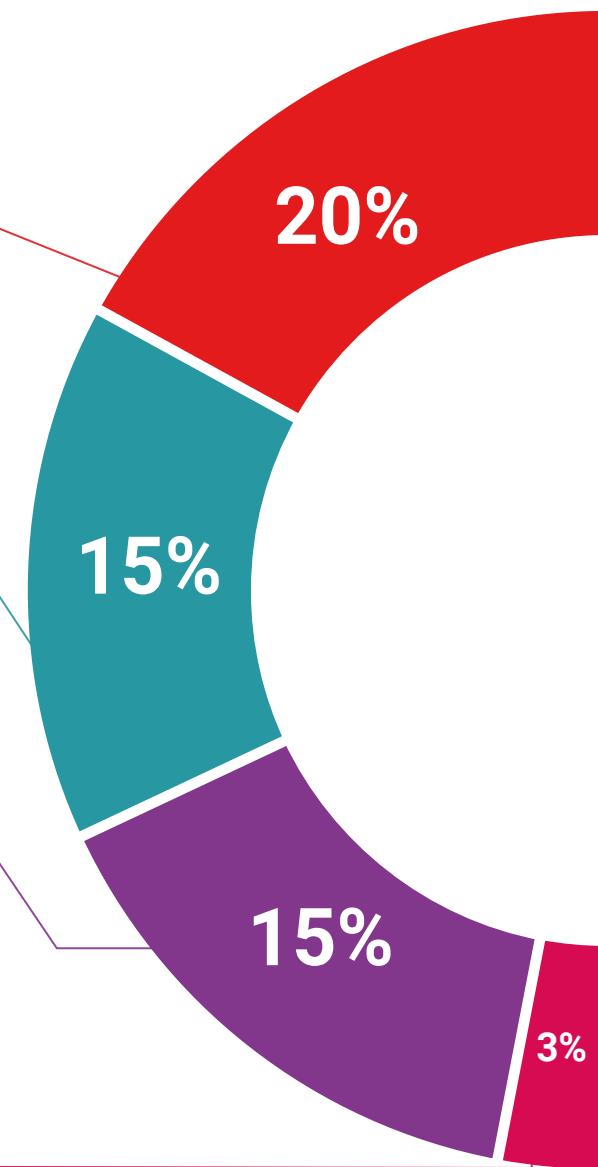
Мы представляем содержание в привлекательной и динамичной форме для воспроизведения на мультимедийных устройствах, которые включают аудио, видео, изображения, диаграммы и концептуальные карты для закрепления знаний.

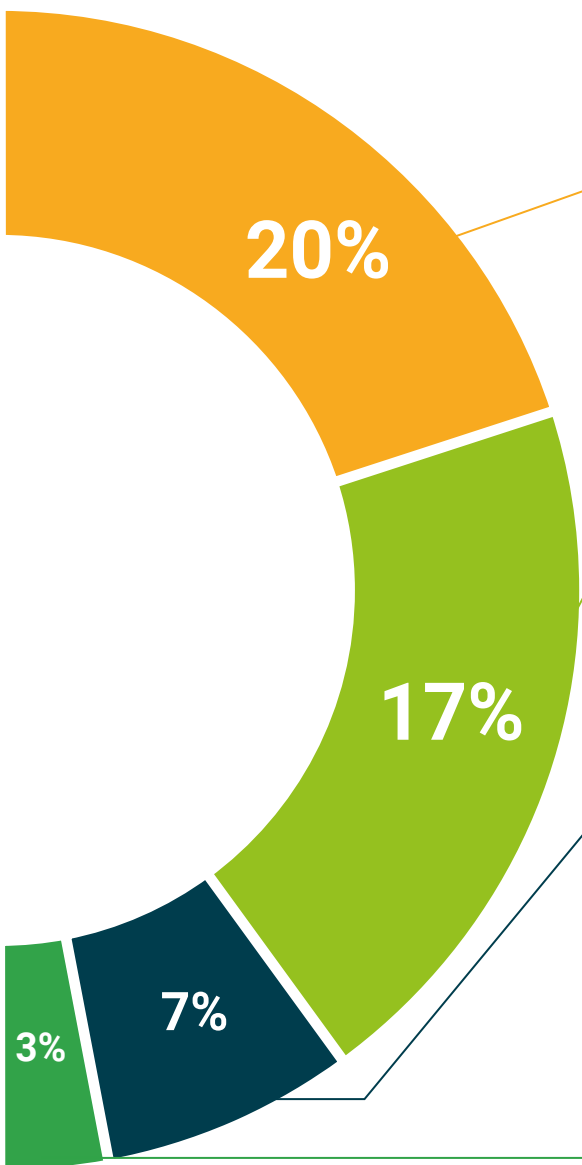
Эта эксклюзивная образовательная система для презентации мультимедийного содержания была награждена Microsoft как "Кейс успеха в Европе".



#### Дополнительная литература

Последние статьи, консенсусные документы, международные рекомендации... В нашей виртуальной библиотеке вы получите доступ ко всему, что необходимо для прохождения обучения.





#### Кейс-стади

Студенты завершат выборку лучших кейс-стади по предмету. Кейсы представлены, проанализированы и преподаются ведущими специалистами на международной арене.



#### Тестирование и повторное тестирование

Мы периодически оцениваем и переоцениваем ваши знания на протяжении всей программы. Мы делаем это на 3 из 4 уровней пирамиды Миллера.



#### Мастер-классы

Существуют научные данные о пользе экспертного наблюдения третьей стороны.

Так называемый метод обучения у эксперта (learning from an expert) укрепляет знания и память, а также формирует уверенность в ваших будущих сложных решениях.



#### Краткие справочные руководства

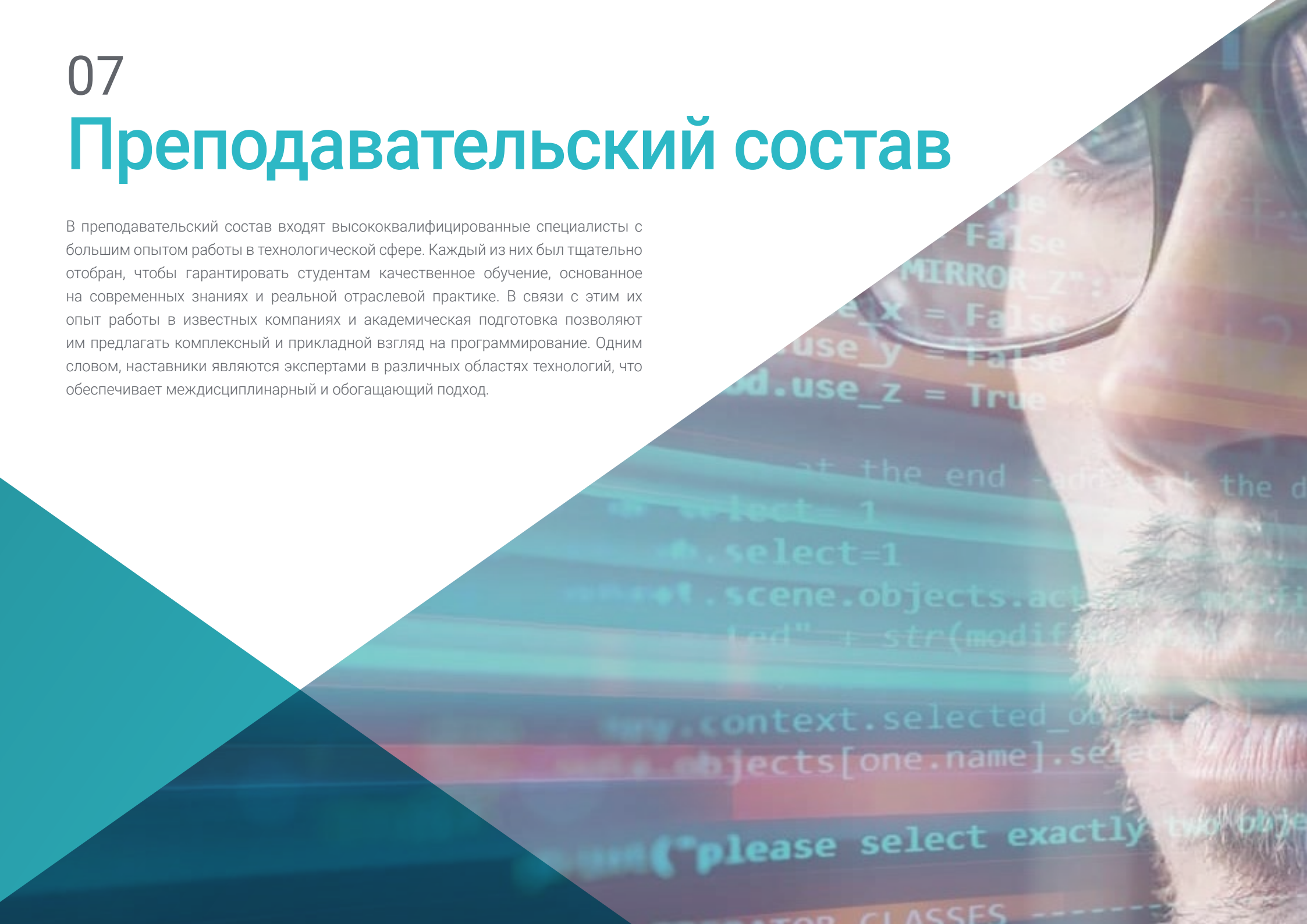
TECH предлагает наиболее актуальные материалы курса в виде карточек или кратких справочных руководств. Это сжатый, практичный и эффективный способ помочь студенту продвигаться в обучении.



07

# Преподавательский состав

В преподавательский состав входят высококвалифицированные специалисты с большим опытом работы в технологической сфере. Каждый из них был тщательно отобран, чтобы гарантировать студентам качественное обучение, основанное на современных знаниях и реальной отраслевой практике. В связи с этим их опыт работы в известных компаниях и академическая подготовка позволяют им предлагать комплексный и прикладной взгляд на программирование. Одним словом, наставники являются экспертами в различных областях технологий, что обеспечивает междисциплинарный и обогащающий подход.



“

*Преподавательский состав является основой успеха программы. Качество и опыт наставников обеспечат всестороннее обучение, которое подготовит вас к уверенной и квалифицированной работе в технологическом секторе”*

## Руководство



### Д-р Лукас Куэста, Хуан Мануэль

- ♦ Старший инженер-программист и аналитик в Indizen - Believe in Talent
- ♦ Старший инженер-программист и аналитик в компаниях Krell Consulting и IMAGiNA Artificial Intelligence
- ♦ Инженер-программист в Intel Corporation
- ♦ Инженер-программист в компании Intelligent Dialogue Systems
- ♦ Доктор наук в области проектирования электронных систем для интеллектуальных сред в политехническом университете Мадрида
- ♦ Степень бакалавра в области инженерии по телекоммуникациям в политехническом университете Мадрида
- ♦ Степень магистра наук в области проектирования электронных систем для интеллектуальных сред в политехническом университете Мадрида



### Д-р Маркес Руис де Лаканаль, Хуан Антонио

- ♦ Разработчик программного обеспечения в GTD Defense & Security Solutions
- ♦ Разработчик программного обеспечения в Solera Inc.
- ♦ Инженер-разработчик и инженер-исследователь в GRVC Sevilla
- ♦ Соучредитель Unmute
- ♦ Соучредитель VR Educa
- ♦ Академический обмен в области инженерии и предпринимательства в Калифорнийском университете Беркли
- ♦ Степень бакалавра в области промышленной инженерии в Университете Севильи

## Преподаватели

### Д-р Луна Перехон, Франсиско

- ♦ Специалист в области компьютерной архитектуры и технологиям
- ♦ Степень доктора в области компьютерной инженерии Севильского университета
- ♦ Университетская степень магистра в области компьютерной инженерии в Севильского университета
- ♦ Степень бакалавра в области инженерного дела в здравоохранении Севильского Университета
- ♦ Степень бакалавра в области компьютерной инженерии и информационных технологий Севильского университета
- ♦ Член: Исследовательская группа в области робототехники и информационных технологий (TEP108)

### Г-н Перис Мильян, Эдуардо

- ♦ Директор технологического направления консалтинга
- ♦ Специалист в области компьютерной инженерии
- ♦ Степень магистра в области стратегического управления информацией и знаниями в организациях
- ♦ Степень магистра в области лидерства и государственного управления
- ♦ Эксперт в области государственного управления
- ♦ Эксперт в области вычислительных систем для Smart CITIES

### Г-н Пи Морель, Ориоль

- ♦ Функциональный аналитик в компании Fihosa
- ♦ Владелец продуктов хостинга и электронной почты в CDmon
- ♦ Функциональный аналитик и инженер-программист в компаниях Atmira и Capgemini
- ♦ Преподаватель в Capgemini, Forms Capgemini и Atmira
- ♦ Степень бакалавра в области технического инженерного дела в Информатике управления Автономного университета Барселоны

- ♦ Степень магистра в области искусственного интеллекта в Католическом университете Авилы
- ♦ MBA в области управления и администрирования бизнеса от IMF Smart Education
- ♦ Степень магистра в области управления информационными системами от IMF Smart Education
- ♦ Аспирантура в области шаблонов проектирования Открытого Университета Каталонии

### Г-н Грильо Эрнандес, Хосе Энрике

- ♦ Разработчик приложений и технологический аналитик
- ♦ Старший разработчик мобильных приложений в компании Globant
- ♦ Разработчик Android в Plexus Tech
- ♦ Старший Android-разработчик в RoadStr
- ♦ Старший разработчик мобильных приложений в Avantgarde IT-Information Technology Services
- ♦ Руководитель проекта в Smartdessa
- ♦ Разработчик в Educatablet
- ♦ Технологический аналитик в Corporate Mobile Solutions
- ♦ Степень магистра в области системной инженерии, полученная в Университете Симона Боливара

### Г-жа Домингес Вальдеррама, Десире

- ♦ *Lead Product & Growth Strategist*
- ♦ Степень магистра в области графического дизайна бизнес-школы Торговой палаты Севильи
- ♦ Эксперт в области UX/UI-дизайна от CoderHouse
- ♦ Эксперт в области технологий и предпринимательства

08

# Квалификация

Специализированная магистратура в области Программирование с нуля гарантирует, помимо самого строгого и современного обучения, получение диплома об окончании Специализированной магистратуры, выдаваемого TECH Global University.



“

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

Данная программа позволит вам получить собственный диплом университета – **Специализированная магистратура в области Программирование с нуля**, одобренный **TECH Global University**, крупнейшим цифровым университетом в мире.

**Tech Global University**, является Официальным Европейским Университетом, признанным правительством Андорры (официальный бюллетень). Андорра является частью Европейского пространства высшего образования (ЕПВО) с 2003 года. ЕПВО – это инициатива, выдвинутая Европейским союзом с целью организации международной системы обучения и гармонизации систем высшего образования стран-участниц этого пространства. Проект способствует распространению общих ценностей, внедрению совместных инструментов и укреплению механизмов обеспечения качества для расширения сотрудничества и мобильности между студентами, исследователями и учеными.

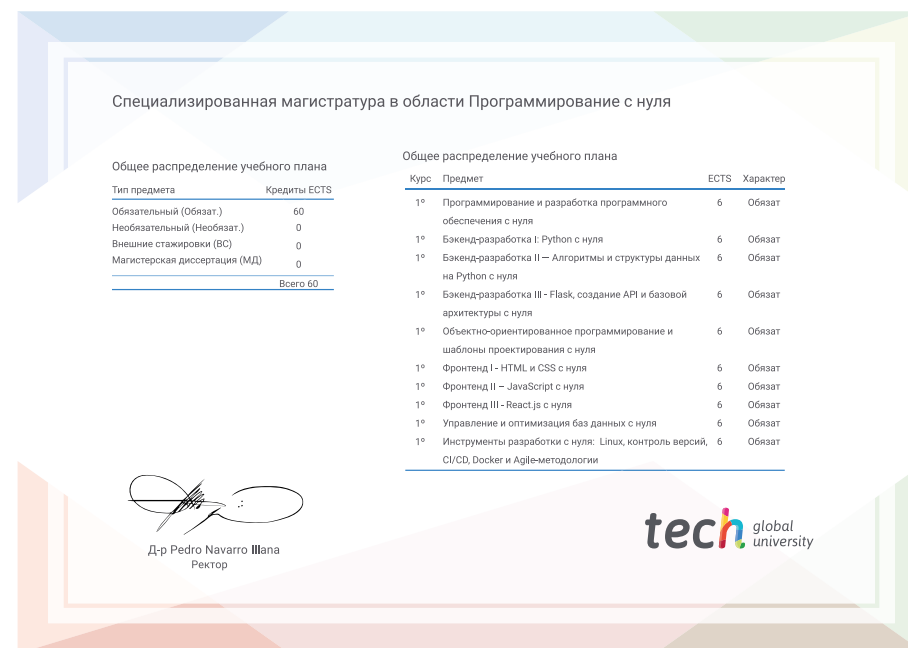
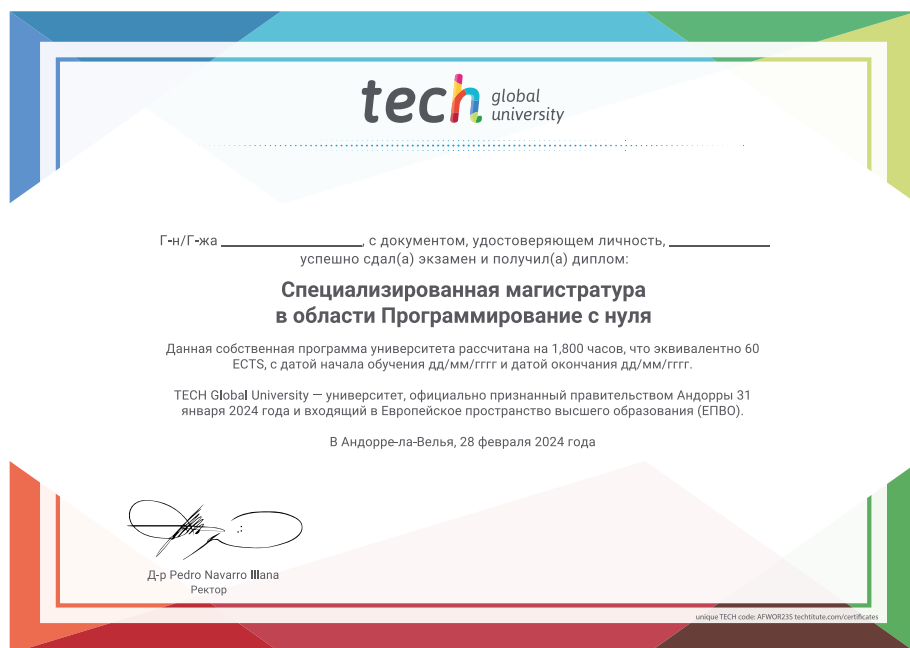
Данный собственный диплом **Tech Global University** – европейская программа непрерывного обучения и повышения квалификации, которая гарантирует приобретение компетенций в своей области знаний, обеспечивая высокую учебную ценность для студента, прошедшего эту программу.

Диплом: **Специализированная магистратура в области Программирование с нуля**

Формат: **онлайн**

Продолжительность: **12 месяцев**

Аккредитация: **60 ECTS**



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

Будущее

Здоровье Доверие Люди

Образование Информация Тьюторы

Гарантия Аккредитация Преподавание

Институты Технология Обучение

Сообщество Обязательства

**tech** global  
university

Специализированная  
магистратура  
Программирование  
с нуля

- » Формат: онлайн
- » Продолжительность: 12 месяцев
- » Учебное заведение: TECH Global University
- » Аккредитация: 60 ECTS
- » Расписание: по своему усмотрению
- » Экзамены: онлайн

Специализированная  
магистратура  
Программирование  
с нуля