

# Специализированная магистратура Разработка на Python



**tech** технологический  
университет

## Специализированная магистратура Разработка на Python

- » Формат: онлайн
- » Продолжительность: 12 месяцев
- » Квалификация: ТЕСН Технологический университет
- » Расписание: по своему усмотрению
- » Экзамены: онлайн

Веб-доступ: [www.techitute.com/ru/information-technology/professional-master-degree/master-python-development](http://www.techitute.com/ru/information-technology/professional-master-degree/master-python-development)

# Оглавление

01

Презентация

---

стр. 4

02

Цели

---

стр. 8

03

Компетенции

---

стр. 12

04

Руководство курса

---

стр. 16

05

Структура и содержание

---

стр. 20

06

Методология

---

стр. 30

07

Квалификация

---

стр. 38

# 01

# Презентация

Python выделяется как многофункциональный и эффективный язык программирования. Во-первых, его четкий и понятный синтаксис облегчает написание и понимание кода, что ускоряет процесс разработки и снижает вероятность ошибок. Кроме того, широкий спектр доступных библиотек и *фреймворков* дает разработчикам доступ к мощным, уже существующим инструментам, что позволяет ускорить и повысить эффективность разработки. Его открытый исходный код и активное сообщество разработчиков также способствуют постоянному совершенствованию и обновлению языка, обеспечивая динамичную и современную среду разработки. Для этого ТЕСН разработала комплексную программу, направленную на обучение профессионалов последним достижениям в области разработки на Python, используя 100% онлайн-методику.



“

*Python славится своей адаптивностью и используется в самых разных областях – от веб-разработки до искусственного интеллекта. Чего вы ждете, чтобы поступить?”*

Python – это высокоуровневый язык программирования, широко используемый среди программистов, потому что он имеет широкий спектр библиотек и фреймворков, которые упрощают общие задачи, позволяя разработчикам сосредоточиться на концепции своих проектов, а не тратить время на технические детали. Кроме того, существенным преимуществом является его универсальность, поскольку он может использоваться в самых разных средах, от веб-разработки до анализа данных и машинного обучения.

Так появилась эта Специализированная магистратура в области разработки на Python – программа, в которой рассматривается расширенная работа с данными и типами в Python, изучаются идентификаторы, ключевые слова, интегральные и булевы типы, а также расширенное форматирование строк и кодировок. Кроме того, будут рассмотрены понятия такие как кортежи, списки и словари, а также методы итерации и лямбда-функции, что обеспечит прочную основу языка.

Также вы погрузитесь в разработку приложений на Python с акцентом на лучшие практики и современные методологии. От архитектуры приложения до развертывания и сопровождения, будут рассмотрены такие аспекты, как передовое проектирование и моделирование, управление зависимостями, шаблоны проектирования, *тестирование* и *отладка*, оптимизация производительности, стратегии внедрения и распространения.

Аналогичным образом, программист будет углубляться в веб- и мобильную разработку на Python, изучая такие *фреймворки*, как Django и Flask, а также разработку API и веб-сервисов. Кроме того, в центре внимания будет дизайн интерфейса и пользовательский опыт (UI/UX), от использования инструментов проектирования до улучшения доступности и юзабилити. В завершении будут рассмотрены вопросы управления данными и их анализа с использованием языка Python и таких инструментов, как NumPy, Pandas и Matplotlib.

Таким образом, эта программа представляет собой уникальную возможность, благодаря полностью адаптируемому академическому предложению в онлайн-формате. Благодаря такому подходу профессионалы получают большую свободу в управлении своим рабочим временем, что позволит им гармонично сочетать ежедневную работу с личными занятиями.

Данная **Специализированная магистратура в области разработки программного обеспечения** содержит самую полную и современную образовательную программу на рынке. Основными особенностями обучения являются:

- ♦ Разбор практических кейсов, представленных экспертами в области разработки программного обеспечения
- ♦ Наглядное, схематичное и исключительно практичное содержание курса предоставляет теоретическую и практическую информацию по тем дисциплинам, которые необходимы для профессиональной практики
- ♦ 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



*Комплексный и специализированный курс по разработке на Python, который подготовит вас к решению реальных задач в мире разработки программного обеспечения"*

“

*Благодаря этой Специализированной магистратуре, полученной на 100% в режиме онлайн, вы углубите свои знания в области frontend- и backend-разработки, управления базами данных и стратегии размещения приложений в магазинах приложений”*

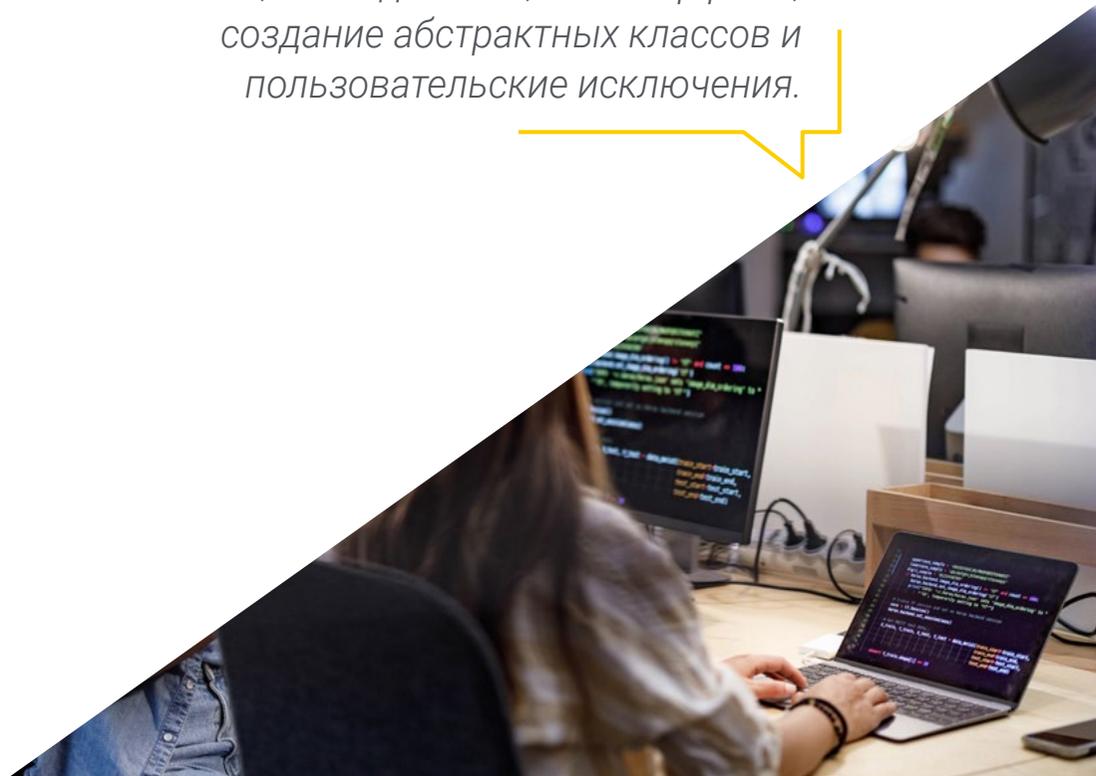
В преподавательский состав программы входят профессионалы из данного сектора, которые привносят в обучение опыт своей работы, а также признанные специалисты из ведущих сообществ и престижных университетов.

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

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

*Вы узнаете обо всем, начиная с основ, таких как переменные и типы данных, и заканчивая передовыми методами визуализации данных и стратегиями оптимизации производительности и хранения. Записывайтесь прямо сейчас!*

*Вы погрузитесь в объектно-ориентированное программирование (ООП), а также в такие темы, как классы, наследование, полиморфизм, создание абстрактных классов и пользовательские исключения.*



# 02

## Цели

Эта Специализированная магистратура была разработана с четкой целью подготовки высококомпетентных и специализированных профессионалов в обширной области программирования на этом высокопроизводительном языке. Благодаря комплексным модулям студенты получают глубокое погружение в синтаксис и функциональность Python, от продвинутой работы с данными до мастерства в объектно-ориентированном программировании (ООП) и эффективного проектирования веб- и мобильных приложений. Благодаря тщательному и современному педагогическому подходу эта программа гарантирует уникальный образовательный опыт, который сделает программистов еще более востребованными экспертами в индустрии разработки программного обеспечения. Приготовьтесь расширить границы своих возможностей и отправиться в образовательное путешествие, которое не только повысит ваши навыки работы с Python, но и даст вам инструменты для достижения успеха и лидерства в динамичном мире развития технологий.



““

*Станьте лидером в области разработки программного обеспечения, вооружившись передовыми знаниями и специализированными навыками для решения современных технологических задач”*



## Общие цели

- ♦ Обеспечить всеобъемлющее понимание Python
- ♦ Обучить продвинутой работе с данными и типами в Python
- ♦ Применять принципы объектно-ориентированного программирования (ООП) в Python
- ♦ Продвигать использование лучших практик и современных методологий в разработке программного обеспечения
- ♦ Предоставить комплексное обучения в области веб- и мобильной разработки на Python
- ♦ Интегрировать принципы UI/UX в разработку программного обеспечения
- ♦ Обучить настройке и использованию инструментов и сред разработки данных
- ♦ Углубиться в использование структур данных и функций в Python
- ♦ Обучить передовым методам визуализации данных с помощью Matplotlib
- ♦ Научить стратегиям оптимизации производительности и хранения данных



*Благодаря обширной библиотеке, которую предлагает эта программа, наполненной самыми инновационными мультимедийными ресурсами, вы получите инструменты, которые помогут вам выделиться и стать лидером в динамичном мире технологического развития"*





## Конкретные цели

---

### Модуль 1. Программирование на Python

- ♦ Обеспечить настройку и эффективное использование среды разработки Python
- ♦ Понимать передовые концепции программирования

### Модуль 2. Расширенные возможности управления данными и потоками с помощью Python

- ♦ Освоить соглашения и методы работы с идентификаторами и ключевыми словами
- ♦ Применять сложные структуры данных и оперировать с ними

### Модуль 3. Объектно-ориентированное программирование (ООП) в Python

- ♦ Освоить создание и использование классов и объектов в Python
- ♦ Применять принципы наследования и полиморфизма в Python

### Модуль 4. Разработка приложений на Python

- ♦ Специализироваться на продвинутой разработке и моделировании современных приложений
- ♦ Обучиться оптимизации, развертыванию и обслуживанию приложений

### Модуль 5. Веб- и мобильная разработка с помощью Python

- ♦ Использовать популярные *фреймворки* Python
- ♦ Подготовиться к разработке и размещению мобильных приложений

### Модуль 6. Пользовательский интерфейс и пользовательский опыт с Python

- ♦ Усвоить методы адаптивного и гибкого дизайна
- ♦ Подготовить к тестированию юзабилити и анализу поведения пользователей

### Модуль 7. Обработка данных и больших данных с помощью Python

- ♦ Управлять методами и функциями контроля потока при обработке данных
- ♦ Продвигать лучшие практики программирования и обработку ошибок на Python

### Модуль 8. Структуры данных и функции Python

- ♦ Создавать и использовать функции продвинутыми методами
- ♦ Считывать, записывать, обрабатывать файлы

### Модуль 9. Управление данными в среде Python с помощью NumPy и Pandas

- ♦ Создавать и работать с массивами с помощью NumPy
- ♦ Формировать навыки визуализации данных с помощью Matplotlib

### Модуль 10. Продвинутые методики и практические применения в NumPy и Pandas

- ♦ Развить специализированный опыт загрузки и хранения данных в различные источники
- ♦ Обучить передовым стратегиям оптимизации и преобразования данных

# 03

## Компетенции

Программа не только даст технические знания, но и будет направлена на развитие способностей, которые превратят студентов в выдающихся профессионалов в области разработки программного обеспечения. Разовьются способности использовать инновационные и эффективные решения на основе мастерского владения Python, начиная с продвинутой обработки данных и заканчивая созданием высокопроизводительных веб- и мобильных приложений. Кроме того, программисты вооружатся специальными навыками объектно-ориентированного программирования (ООП), дизайна пользовательского интерфейса и пользовательского опыта (UI/UX), а также продвинутого анализа данных с помощью таких библиотек, как NumPy и Pandas.

“

*Будет поощряться развитие важнейших «мягких» навыков, таких как критическое мышление, решение проблем, сотрудничество в команде и адаптивность”*



## Общие профессиональные навыки

---

- ♦ Развить практические навыки программирования
- ♦ Овладеть передовыми навыками управления потоками данных
- ♦ Управлять разработкой и внедрением объектно-ориентированного программного обеспечения
- ♦ Обучиться целостной разработке приложений на языке Python
- ♦ Освоить разработку и управление веб- и мобильными приложениями
- ♦ Контролировать дизайн пользовательского интерфейса и пользовательского опыта с помощью Python
- ♦ Развивать навыки работы с данными и их анализа с помощью Python
- ♦ Приобрести навыки в продвинутом управлении файлами и моделировании на Python
- ♦ Формировать передовые навыки работы с данными с помощью NumPy и Pandas
- ♦ Углубить навыки работы с данными с помощью NumPy и Pandas

“

*Вы столкнетесь с реальными проблемами и сможете добиться успеха на конкурентном и постоянно развивающемся рынке труда. Выбирайте TECH!”*





## Конкретные профессиональные навыки

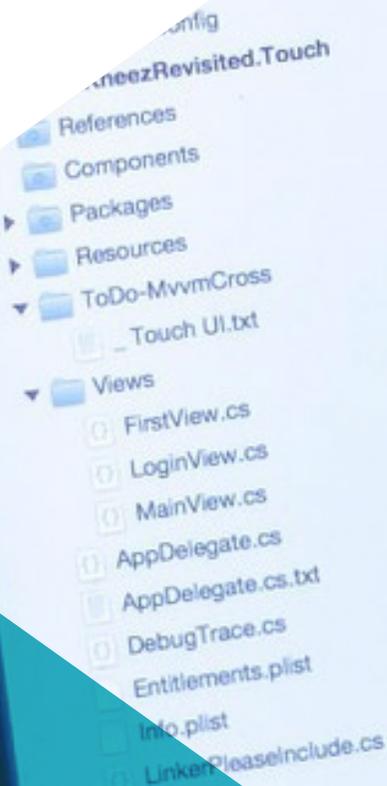
---

- Обучиться продвинутым методам работы с данными в Python
- Овладеть продвинутым использованием функций в Python
- Реализовать передовые концепции объектно-ориентированного программирования (ООП), такие как абстрактные классы и пользовательские исключения
- Контролировать тестирование и *дебаггинг*
- Разрабатывать API и веб-сервисы
- Освоить использование инструментов UI/UX-дизайна с помощью Python
- Использовать основные библиотеки для работы с данными в Python
- Применять различные структуры данных на практике
- Использовать Pandas для управления структурированными данными
- Продвигать компетенции в области анализа и манипулирования временными рядами и сложными данными

# 04

# Руководство курса

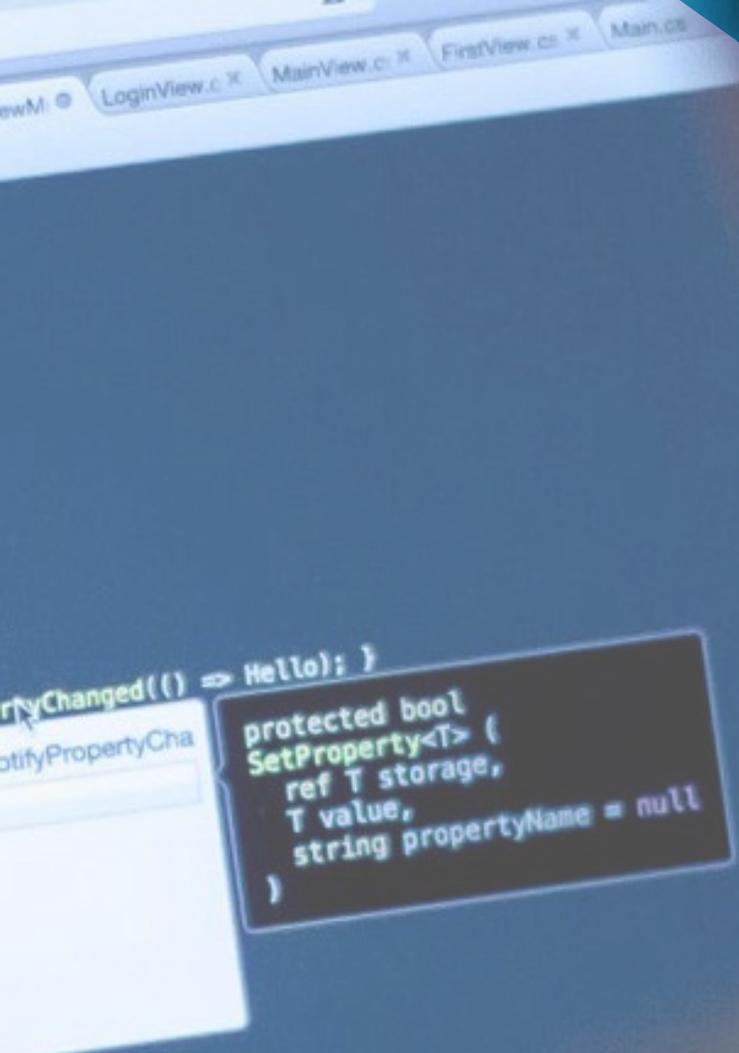
Педагоги этой Специализированной магистратуры – эксперты отрасли, обладающие обширным практическим опытом разработки программного обеспечения с доказанным послужным списком успешных проектов. Сочетая теорию с практическим применением, эти наставники не только передают передовые знания Python, но и делятся своим реальным опытом, предоставляя студентам ценную информацию о лучших методах и актуальных тенденциях в отрасли. Кроме того, его педагогический подход будет отличаться стремлением предоставить студентам самую современную техническую подготовку для успешной работы в динамичной области разработки программного обеспечения.



```
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22
```

```
public class BeezKneezRevisited.Core  
{  
    public class MainViewModel : MvxViewModel  
    {  
        {  
            private string _hello = "Hello MOFO";  
            public string Hello  
            {  
                get { return _hello; }  
                set { set _hello = value; RaisePrope  
            }  
        }  
    }  
}
```

Cirrious.MvvmCross.ViewModels.MvxN  
SetProperty



“

*Вы добьетесь успеха вместе с лучшими и приобретете знания и навыки, необходимые для работы в индустрии разработки на Python”*

## Руководство



### Г-н Матос Родригес, Дионис

- ♦ Инженер по обработке данных в Wide Agency Sadexo
- ♦ Консультант по данным в Tokiota
- ♦ Инженер по обработке данных в Devoteam
- ♦ BI-разработчик в Ibermática
- ♦ Разработчик прикладных программ в Johnson Controls
- ♦ Разработчик баз данных в Suncapital Spain
- ♦ Старший веб-разработчик в Deadlock Solutions
- ♦ QA-аналитик в Metaconcept
- ♦ Магистр в области больших данных и аналитики в EAE Business School
- ♦ Степень магистра в области системного анализа и проектирования
- ♦ Степень бакалавра в области компьютерной инженерии в Университете APEC

## Преподаватели

### Г-н Вильяр Валор, Хавьер

- ♦ Директор и партнер-основатель компании Impulsa2
- ♦ *Главный операционный директор (COO)* в Summa Insurance Brokers
- ♦ Директор по трансформации и профессиональному совершенству в Johnson Controls
- ♦ Степень магистра в области профессионального *коучинга*
- ♦ Executive MBA в Emlyon Business School, Франция
- ♦ Степень магистра в области управления Качеством Университета EOI
- ♦ Компьютерная инженерия в университете Acción Pro-Educación y Cultura (UNAPEC)

### Г-н Хиль Контрерас, Армандо

- ♦ *Ведущий специалист по большим данным* в Johnson Controls
- ♦ *Специалист по большим данным* в Opensistemas S.A
- ♦ Аудитор Фонда Creatividad y Tecnología S.A. (CYTSA)
- ♦ Аудитор государственного сектора в компании PricewaterhouseCoopers Auditores
- ♦ Степень магистра в области *науки о данных* в Университетском Центре технологий и искусства
- ♦ Степень магистра MBA в области международных отношений и бизнеса в Центре финансовых исследований (CEF)
- ♦ Степень бакалавра в области экономики в Технологическом институте Санто-Доминго

### Г-жа Хиль Контрерас, Милагрос

- ♦ *Создатель контента* в MPCTech LLC
- ♦ Руководитель проекта
- ♦ *Внештатный IT-писатель*
- ♦ MBA Университета Комплутенсе в Мадриде.
- ♦ Степень бакалавра делового администрирования Технологического института Санто-Доминго

### Г-жа Дельгадо Фелис, Бенедит

- ♦ Административный помощник и оператор электронного наблюдения в Национальном управлении по контролю над наркотиками
- ♦ Обслуживание клиентов в Cáceres y Equipos
- ♦ Рекламации и обслуживание клиентов в компании Express Parcel Services (EPS)
- ♦ Специалист по Microsoft Office в Национальной школе информатики
- ♦ Социальный коммуникатор Католического Университета Санто-Доминго



*Воспользуйтесь возможностью узнать о последних достижениях в этой области, чтобы применить их в своей повседневной практике"*

# 05

# Структура и содержание

Учебный план был тщательно разработан, чтобы обеспечить студентам всесторонний опыт. От глубокого погружения в синтаксис и расширенные функциональные возможности Python до специализации в объектно-ориентированном программировании (ООП), разработке веб- и мобильных приложений и экспертной работе с данными с помощью таких библиотек, как NumPy и Pandas, – каждый модуль будет тщательно структурирован, чтобы дать прочные знания и практические навыки. Кроме того, будут рассмотрены такие важные темы, как дизайн пользовательского интерфейса и пользовательский опыт (UI/UX), передовые методы работы с данными, оптимизация производительности и хранения данных.





“

*Вы не только полностью овладеете языком Python, но и будете готовы уверенно решать динамичные задачи, ориентированные на технологическую промышленность”*

## Модуль 1. Программирование на Python

- 1.1. Создание и выполнение программ на Python
  - 1.1.1. Конфигурация сред разработки
  - 1.1.2. Выполнение скриптов Python
  - 1.1.3. Интегрированные средства разработки (IDE)
- 1.2. Данные в Python
  - 1.2.1. Простейшие типы (*int*, *float*, *str*)
  - 1.2.2. *Casting* и преобразование типов данных в Python
  - 1.2.3. Неизменяемость и хранение данных в Python
- 1.3. Ссылки на объекты в Python
  - 1.3.1. Ссылки в памяти
  - 1.3.2. Идентичность vs. Равенство
  - 1.3.3. Управление ссылками и сборка мусора
- 1.4. Сбор данных в Python
  - 1.4.1. Списки и типовые операции с ними
  - 1.4.2. Кортежи и их неизменяемость
  - 1.4.3. Словари и доступ к данным
- 1.5. Логические операции в Python
  - 1.5.1. Булевы операторы
  - 1.5.2. Условные выражения
  - 1.5.3. *Оценка с коротким замыканием*
- 1.6. Арифметические операторы в Python
  - 1.6.1. Арифметические операции в Python
  - 1.6.2. Операторы деления
  - 1.6.3. Приоритет и ассоциативность
- 1.7. Ввод/вывод данных в Python
  - 1.7.1. Считывание данных со стандартного ввода
  - 1.7.2. Запись данных в стандартный вывод
  - 1.7.3. Управление файлами
- 1.8. Создание и вызов функций в Python
  - 1.8.1. Синтаксис функций
  - 1.8.2. Параметры и аргументы
  - 1.8.3. Возвращаемые значения и анонимные функции



- 1.9. Использование *строк* в Python
  - 1.9.1. Работа со *строками* и их форматирование
  - 1.9.2. Общие *строковые* методы
  - 1.9.3. Интерполяция и *F-строки*
- 1.10. Обработка ошибок и исключений в Python
  - 1.10.1. Распространенные типы исключений
  - 1.10.2. Блоки *try-except*
  - 1.10.3. Создание персонализированных исключений

## Модуль 2. Расширенные возможности управления данными и потоками с помощью Python

- 2.1. Идентификаторы и ключевые слова в Python
  - 2.1.1. Правила для имен переменных
  - 2.1.2. Зарезервированные слова в Python
  - 2.1.3. Соглашения об именах
- 2.2. Целые и булевы типы в Python
  - 2.2.1. Целые типы
  - 2.2.2. Операции, специфичные для булевых чисел
  - 2.2.3. Преобразования и представления
- 2.3. Типы с плавающей точкой и комплексные числа в Python
  - 2.3.1. Точность и репрезентативность
  - 2.3.2. Операции с плавающей запятой
  - 2.3.3. Использование комплексных чисел в вычислениях
- 2.4. Форматирование *строк* и кодировок в Python
  - 2.4.1. Расширенные методы форматирования
  - 2.4.2. Кодировки *Unicode* и UTF-8
  - 2.4.3. Работа со специальными символами
- 2.5. Коллекции: Кортежи, списки и словари в Python
  - 2.5.1. Сравнение и противопоставление типов
  - 2.5.2. Методы, специфичные для конкретных типов
  - 2.5.3. Эффективность и выбор подходящего типа
- 2.6. *Наборы* и *замороженные наборы* в Python
  - 2.6.1. Создание и операции с *наборами*
  - 2.6.2. *Замороженные наборы*
  - 2.6.3. Практическое применение и производительность

- 2.7. Итерация и копирование коллекций в Python
  - 2.7.1. Циклы *for* и генераторы списков
  - 2.7.2. Поверхностное копирование vs. Глубокое копирование
  - 2.7.3. Итераторы и генераторы
- 2.8. Использование *лямбда-функций* в Python
  - 2.8.1. Синтаксис и создание *лямбда-функций*
  - 2.8.2. Применение в фильтрах и картах
  - 2.8.3. Ограничения и оптимальные практики
- 2.9. Структуры управления: Условия и циклы в Python
  - 2.9.1. Структуры *if-else* и *elif*
  - 2.9.2. Циклы *while* и *for*
  - 2.9.3. Управление потоком с помощью функций *break continue else*
- 2.10. Продвинутое функции и методы Python
  - 2.10.1. Рекурсивные функции
  - 2.10.2. Функции высшего порядка
  - 2.10.3. Декораторы функций

## Модуль 3. Объектно-ориентированное программирование (ООП) в Python

- 3.1. Объектно-ориентированное программирование (ООП) в Python
  - 3.1.1. Классы и объекты
  - 3.2.1. Инкапсуляция и абстракция
  - 3.3.1. Объектно-ориентированное программирование (ООП) в Python
- 3.2. Создание классов и объектов в Python
  - 3.2.1. Классы в ООП Python
  - 3.2.2. Инстанцирование и методы инициализации
  - 3.2.3. Атрибуты и методы
- 3.3. Атрибуты и методы в Python
  - 3.3.1. Атрибуты экземпляра vs. Атрибуты класса
  - 3.3.2. Методы экземпляра, класса и статические методы
  - 3.3.3. Инкапсуляция и сокрытие информации
- 3.4. Принципы наследования и полиморфизма в Python
  - 3.4.1. Единичное и множественное наследование
  - 3.4.2. Перезапись и расширение методов
  - 3.4.3. Полиморфизм и *утиная типизация*

- 3.5. Свойства и доступ к атрибутам в Python
    - 3.5.1. *Геттеры и сеттеры*
    - 3.5.2. Декоратор *@property*
    - 3.5.3. Управление доступом и валидация
  - 3.6. Пользовательские классы и коллекции в Python
    - 3.6.1. Создание типов коллекций
    - 3.6.2. Специальные методы (*\_\_len\_\_, \_\_getitem\_\_*)
    - 3.6.3. Индивидуальные итераторы
  - 3.7. Агрегация и композиция классов в Python
    - 3.7.1. Отношения между классами
    - 3.7.2. Агрегация vs. Композиция
    - 3.7.3. Управление жизненным циклом объектов
  - 3.8. Использование декораторов в классах Python
    - 3.8.1. Декораторы для методов
    - 3.8.2. Декораторы классов
    - 3.8.3. Области применения и примеры использования
  - 3.9. Абстрактные классы и методы в Python
    - 3.9.1. Абстрактные классы
    - 3.9.2. Абстрактные методы и их реализация
    - 3.9.3. Использование ABC (*абстрактный базовый класс*)
  - 3.10. Исключения и обработка ошибок в ООП в Python
    - 3.10.1. Индивидуальные исключения в классах
    - 3.10.2. Работа с исключениями в методах
    - 3.10.3. Лучшие практики в области исключений и ООП
- Модуль 4. Разработка приложений на Python**
- 4.1. Архитектура приложений на Python
    - 4.1.1. Проектирование программного обеспечения
    - 4.1.2. Общие архитектурные шаблоны
    - 4.1.3. Оценка потребностей и требований
  - 4.2. Разработка и проектирование приложений на языке Python
    - 4.2.1. Использование языка UML и диаграмм
    - 4.2.2. Моделирование данных и информационных потоков
    - 4.2.3. Принципы SOLID и модульный дизайн
  - 4.3. Управление зависимостями и библиотеками в Python
    - 4.3.1. Работа с пакетами с помощью Pip
    - 4.3.2. Использование виртуальных сред
    - 4.3.3. Устранение конфликтов зависимостей
  - 4.4. Шаблоны проектирования в разработке на Python
    - 4.4.1. Творческие, структурные и поведенческие шаблоны
    - 4.4.2. Практическое применение шаблонов
    - 4.4.3. Рефакторинг и шаблоны
  - 4.5. Тестирование и *дебаггинг* приложений на Python
    - 4.5.1. Стратегии *тестирования* (унитарные, интеграционные)
    - 4.5.2. Использование *фреймворков* в тестировании
    - 4.5.3. Методы и инструменты *дебаггинга*
  - 4.6. Безопасность и аутентификация в Python
    - 4.6.1. Безопасность в приложениях
    - 4.6.2. Реализация аутентификации и авторизации
    - 4.6.3. Предотвращение уязвимостей
  - 4.7. Оптимизация и повышение производительности приложений на Python
    - 4.7.1. Анализ производительности
    - 4.7.2. Методы оптимизации кода
    - 4.7.3. Эффективное управление ресурсами и данными
  - 4.8. Развертывание и распространение приложений на Python
    - 4.8.1. Стратегии развертывания
    - 4.8.2. Использование контейнеров и оркестраторов
    - 4.8.3. Распространение и регулярные обновления
  - 4.9. Обслуживание и обновление в Python
    - 4.9.1. Контроль жизненного цикла программного обеспечения
    - 4.9.2. Стратегии сопровождения и рефакторинга
    - 4.9.3. Обновление и миграция системы
  - 4.10. Документация и техническая поддержка в Python
    - 4.10.1. Создание качественной документации
    - 4.10.2. Инструменты для документации
    - 4.10.3. Стратегии техподдержки и связи с пользователями

## Модуль 5. Веб- и мобильная разработка с помощью Python

- 5.1. Веб-разработка на Python
  - 5.1.1. Структура и компоненты веб-разработки
  - 5.1.2. Технологии веб-разработки
  - 5.1.3. Тренды веб-разработки
- 5.2. Популярные фреймворки Python
  - 5.2.1. *Django*, *Flask* и альтернативы
  - 5.2.2. Сравнение и выбор фреймворков
  - 5.2.3. Интеграция с *Frontend*
- 5.3. Разработка *FrontEnd* HTML, CSS и JavaScript с Python
  - 5.3.1. HTML и CSS
  - 5.2.3. JavaScript и работа с DOM
  - 5.3.3. Фреймворки и библиотеки *Frontend*
- 5.4. *Backend* и базы данных с помощью Python
  - 5.4.1. Разработка *Backend* с использованием Python
  - 5.4.2. Управление реляционных и нереляционных баз данных
  - 5.4.3. Интеграция *Backend-Frontend*
- 5.5. API и веб-сервисы с помощью Python
  - 5.5.1. Проектирование *RESTful* API
  - 5.5.2. Внедрение и документирование API
  - 5.5.3. Потребление и безопасность в API
- 5.6. Мобильная разработка на Python
  - 5.6.1. Платформы для мобильной разработки (нативные, гибридные)
  - 5.6.2. Инструменты и среды разработки
  - 5.6.3. Адаптация приложений для мобильных устройств
- 5.7. Платформы для мобильной разработки с Python
  - 5.7.1. Android и iOS
  - 5.7.2. Фреймворки для кросс-платформенной разработки
  - 5.7.3. Тестирование и развертывание на мобильных устройствах
- 5.8. Дизайн и UX в мобильных приложениях с помощью Python
  - 5.8.1. Разработка мобильных интерфейсов
  - 5.8.2. Юзабилити и пользовательский опыт с Python
  - 5.8.3. Инструменты для создания прототипов и дизайна

- 5.9. Тестирование и дебаггинг мобильных приложений на Python
  - 5.9.1. Стратегии тестирования на мобильных устройствах
  - 5.9.2. Инструменты дебаггинга и мониторинга
  - 5.9.3. Автоматизация тестирования
- 5.10. Размещение публикаций в магазинах приложений
  - 5.10.1. Процесс размещения в App Store и Google Play
  - 5.10.2. Соответствие нормам и политики приложений
  - 5.10.3. Стратегии маркетинга и продвижения

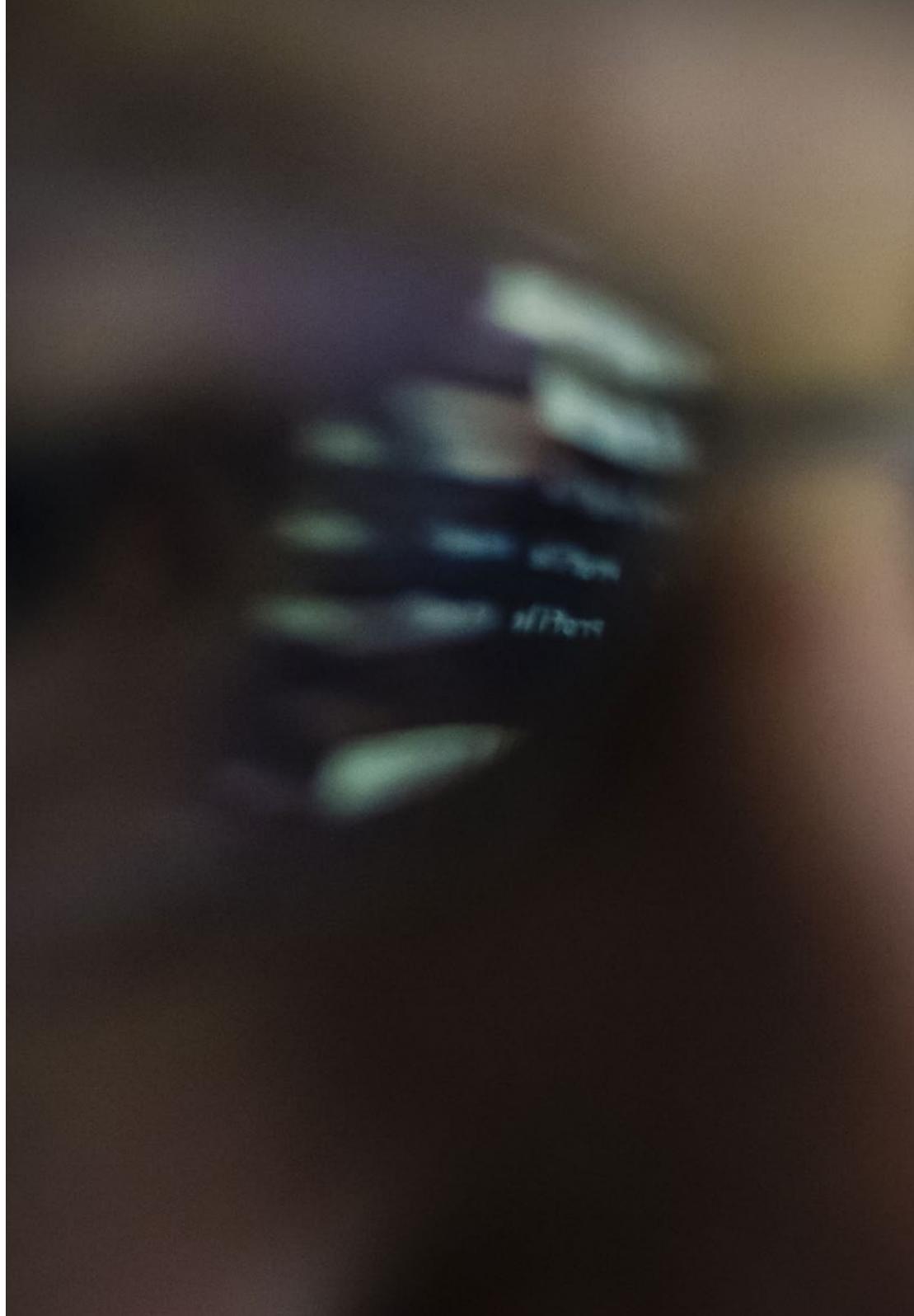
## Модуль 6. Пользовательский интерфейс и пользовательский опыт с Python

- 6.1. Проектирование пользовательских интерфейсов с помощью Python
  - 6.1.1. Разработка UI с помощью Python
  - 6.1.2. Взаимодействие пользователя и цифрового устройства с помощью Python
  - 6.1.3. Дизайн, ориентированный на пользователя с Python
- 6.2. Инструменты для проектирования UI/UX с помощью Python
  - 6.2.1. Программное обеспечение разработки и создания прототипов
  - 6.2.2. Инструменты для совместной работы *Feedback*
  - 6.2.3. Интеграция дизайна в процесс разработки
- 6.3. Отзывчивый и адаптивный дизайн с помощью Python
  - 6.3.1. Методы отзывчивого дизайна
  - 6.3.2. Адаптация к различным устройствам и экранам
  - 6.3.3. Тестирование и обеспечение качества
- 6.4. Анимации и переходы с помощью Python
  - 6.4.1. Создание эффективной анимации с помощью Python
  - 6.4.2. Инструменты и библиотеки для анимации
  - 6.4.3. Влияние на UX и производительность
- 6.5. Доступность и удобство использования в Python
  - 6.5.1. Веб-доступность
  - 6.5.2. Инструменты и методы оценки
  - 6.5.3. Внедрение передовых практик
- 6.6. Прототипирование и вайрфреймы с помощью Python
  - 6.6.1. Создание вайрфреймов и мокапов
  - 6.6.2. Инструменты для быстрого прототипирования
  - 6.6.3. Тестирование удобства использования и *Feedback*

- 6.7. Тестирование удобства использования с Python
  - 6.7.1. Способы и методы тестирования удобства использования
  - 6.7.2. Анализ и совершенствование на основе результатов
  - 6.7.3. Инструменты для тестирования удобства использования
- 6.8. Анализ пользовательского поведения с помощью Python
  - 6.8.1. Методы анализа и *Tracking*
  - 6.8.2. Интерпретация и метрики данных
  - 6.8.3. Постоянное совершенствование на основе данных
- 6.9. Улучшения на основе *Feedback* с помощью Python
  - 6.9.1. Управление и анализ *Feedback*
  - 6.9.2. Циклы *Feedback* и непрерывного совершенствования
  - 6.9.3. Стратегии реализации эффективных изменений
- 6.10. Будущие тенденции в UI/UX с помощью Python
  - 6.10.1. Инновации и новые тенденции
  - 6.10.2. Влияние новых технологий на UI/ UX
  - 6.10.3. Подготовка к будущему дизайна

## Модуль 7. Обработка данных и больших данных с помощью Python

- 7.1. Использование Python для работы с данными
  - 7.1.1. Python в области науки о данных и аналитики
  - 7.1.2. Важнейшие библиотеки для данных
  - 7.1.3. Применение и примеры
- 7.2. Конфигурация сред разработки использования с Python
  - 7.2.1. Установка и инструменты Python
  - 7.2.2. Конфигурация виртуальных сред
  - 7.2.3. Интегрированные средства разработки (IDE)
- 7.3. Переменные, типы данных и операторы в Python
  - 7.3.1. Переменные и простейшие типы данных
  - 7.3.2. Структура данных
  - 7.3.3. Арифметические и логические операторы
- 7.4. Контроль потока: Условные обозначения и циклы
  - 7.4.1. Условные управляющие типы (*if, else, elif*)
  - 7.4.2. Циклы (*for, while*) и управление потоком
  - 7.4.3. Списочные представления, генераторные выражения



- 
- 7.5. Функции и модульность в Python
    - 7.5.1. Применение функций
    - 7.5.2. Параметры, аргументы и возвратные значения
    - 7.5.3. Модульность и повторное использование кода
  - 7.6. Обработка ошибок и исключений в Python
    - 7.6.1. Ошибки и исключения
    - 7.6.2. Обработка исключений с помощью *try-except*
    - 7.6.3. Создание персонализированных исключений
  - 7.7. Инструмент *IPython*
    - 7.7.1. Инструмент *IPython*
    - 7.7.2. Использование *IPython* для анализа данных
    - 7.7.3. Отличия от стандартного интерпретатора *Python*
  - 7.8. *Jupyter Notebooks*
    - 7.8.1. *Jupyter Notebooks*
    - 7.8.2. Использование блокнотов для анализа данных
    - 7.8.3. Публикация блокнотов *Jupyter*
  - 7.9. Передовые методы кодирования на Python
    - 7.9.1. Стиль и условные обозначения (SPP 8)
    - 7.9.2. Документация и комментарии
    - 7.9.3. Стратегии тестирования и дебаггинга
  - 7.10. Информационные источники и сообщества Python
    - 7.10.1. Онлайн-ресурсы и документация
    - 7.10.2. Сообщества и форумы
    - 7.10.3. Обучение и совершенствование в Python

## Модуль 8. Структуры данных и функции Python

- 8.1. Множества в Python
  - 8.1.1. Операции и методы
  - 8.1.2. Различия и практическое применение
  - 8.1.3. Итерация и генераторы
- 8.2. Словари и их использование в Python
  - 8.2.1. Создание и работа со словарями
  - 8.2.2. Доступ и управление данными
  - 8.2.3. Паттерны и современные методы

- 8.3. Генераторы списков и словари в Python
  - 8.3.1. Синтаксис и примеры
  - 8.3.2. Эффективность и читабельность
  - 8.3.3. Практическое применение
- 8.4. Функции обработки данных в Python
  - 8.4.1. Создание функций
  - 8.4.2. Сфера применения и пространство имен
  - 8.4.3. Анонимные функции и *лямбда-функции*
- 8.5. Аргументы и возвращаемые значения функций в Python
  - 8.5.1. Позиционные и именованные аргументы
  - 8.5.2. Функции с множественным возвратом
  - 8.5.3. Аргументы переменных и ключевые слова
- 8.6. *Лямбда-функции* и функции высшего порядка в Python
  - 8.6.1. Использование *лямбда-функций*
  - 8.6.2. Функции *Map, Filter* и *Reduce*
  - 8.6.3. Применения в обработке данных
- 8.7. Работа с файлами в Python
  - 8.7.1. Считывание и запись файлов
  - 8.7.2. Работа с двоичными и текстовыми файлами
  - 8.7.3. Передовые методы и обработка исключений
- 8.8. Считывание и запись текстовых и двоичных файлов в Python
  - 8.8.1. Форматы и кодировка файлов
  - 8.8.2. Управление большими файлами
  - 8.8.3. Сериализация и десериализация (*JSON, pickle*)
- 8.9. Контексты и операции над файлами
  - 8.9.1. Использование менеджера контекста (*with*)
  - 8.9.2. Методы обработки файлов
  - 8.9.3. Безопасность и обработка ошибок
- 8.10. Библиотеки моделирования Python
  - 8.10.1. *Scikit-learn*
  - 8.10.2. *TensorFlow*
  - 8.10.3. *Pytorch*

## Модуль 9. Управление данными в среде Python с помощью NumPy и Pandas

- 9.1. Создание *Arrays* и работа с ними в NumPy
  - 9.1.1. NumPy
  - 9.1.2. Основные операции с *Arrays*
  - 9.1.3. Обработка и трансформация *Arrays*
- 9.2. Векторные операции с *Arrays*
  - 9.2.1. Векторизация
  - 9.2.2. Универсальные функции (*ufunc*)
  - 9.2.3. Эффективность и производительность
- 9.3. Индексация и сегментация в NumPy
  - 9.3.1. Доступ к элементам и **Slicing**
  - 9.3.2. Расширенная и булева индексация
  - 9.3.3. Переупорядочивание и отбор
- 9.4. Серия в Pandas и *DataFrames*
  - 9.4.1. Pandas
  - 9.4.2. Структуры данных в Pandas
  - 9.4.3. Работа с *DataFrames*
- 9.5. Индексирование и выборка в Pandas
  - 9.5.1. Доступ к данным в сериях и *DataFrames*
  - 9.5.2. Методы отбора и фильтрации
  - 9.5.3. Использование *loc* и *iloc*
- 9.6. Операции с помощью Pandas
  - 9.6.1. Арифметические операции и выравнивание
  - 9.6.2. Агрегация и статистические функции
  - 9.6.3. Преобразование и применение функций
- 9.7. Работа с неполными данными в Pandas
  - 9.7.1. Обнаружение и работа с нулевыми значениями
  - 9.7.2. Заполнение и удаление неполных данных
  - 9.7.3. Стратегии работы с неполными данными
- 9.8. Функции и применения в Pandas
  - 9.8.1. Конкатенация и объединение данных
  - 9.8.2. Группировка и агрегирование (*groupby*)
  - 9.8.3. *Pivot Tables* и *Crosstabs*

- 9.9. Визуализация с помощью Matplotlib
  - 9.9.1. Matplotlib
  - 9.9.2. Создание графики и ее настройка
  - 9.9.3. Интеграция с Pandas
- 9.10. Настройка графиков в Matplotlib
  - 9.10.1. Стили и конфигурации
  - 9.10.2. Продвинутое графики (*scatter*, *bar*, etc.)
  - 9.10.3. Создание комплексных визуализаций

## Модуль 10. Продвинутое методики и практические применения в NumPy и Pandas

- 10.1. Загрузка данных из разнообразных источников
  - 10.1.1. Импорт из CSV, Excel и баз данных
  - 10.1.2. Чтение данных из API и веб-страниц
  - 10.1.3. Стратегии управления большими данными
- 10.2. Хранение данных в Python
  - 10.2.1. Экспорт в различные форматы
  - 10.2.2. Эффективность хранения
  - 10.2.3. Безопасность и конфиденциальность данных
- 10.3. Стратегии очистки данных в Python
  - 10.3.1. Выявление и устранение несоответствий
  - 10.3.2. Нормализация и преобразование данных
  - 10.3.3. Автоматизация процессов чистки
- 10.4. Усовершенствованное преобразование данных в Pandas
  - 10.4.1. Методы обработки и переработки
  - 10.4.2. Объединение и реструктуризация *DataFrames*
  - 10.4.3. Применение регулярных выражений в Pandas
- 10.5. Комбинация *DataFrames* в Pandas
  - 10.5.1. *Merge*, *Join* и конкатенация
  - 10.5.2. Управление конфликтами и ключами
  - 10.5.3. Эффективные стратегии комбинирования
- 10.6. Усовершенствованное преобразование данных в Pandas
  - 10.6.1. *Pivot* и *Melt*
  - 10.6.2. Техники *Reshape* и транспонирование
  - 10.6.3. Применение в анализе данных
- 10.7. Временные ряды в Pandas
  - 10.7.1. Управление временем и датами
  - 10.7.2. *Resampling* и *Window Functions*
  - 10.7.3. Анализ трендов и сезонности
- 10.8. Расширенное управление индексами в Pandas
  - 10.8.1. Многоуровневые и иерархические индексы
  - 10.8.2. Расширенные возможности выделения и управления
  - 10.8.3. Оптимизация запросов
- 10.9. Стратегии оптимизации производительности
  - 10.9.1. Повышение скорости и эффективности
  - 10.9.2. Использование Cython и Numba
  - 10.9.3. Распараллеливание и распределенная обработка
- 10.10. Практические проекты по работе с данными
  - 10.10.1. Анализ реальных примеров использования
  - 10.10.2. Интеграция методов Python
  - 10.10.3. Стратегии решения сложных проблем с данными



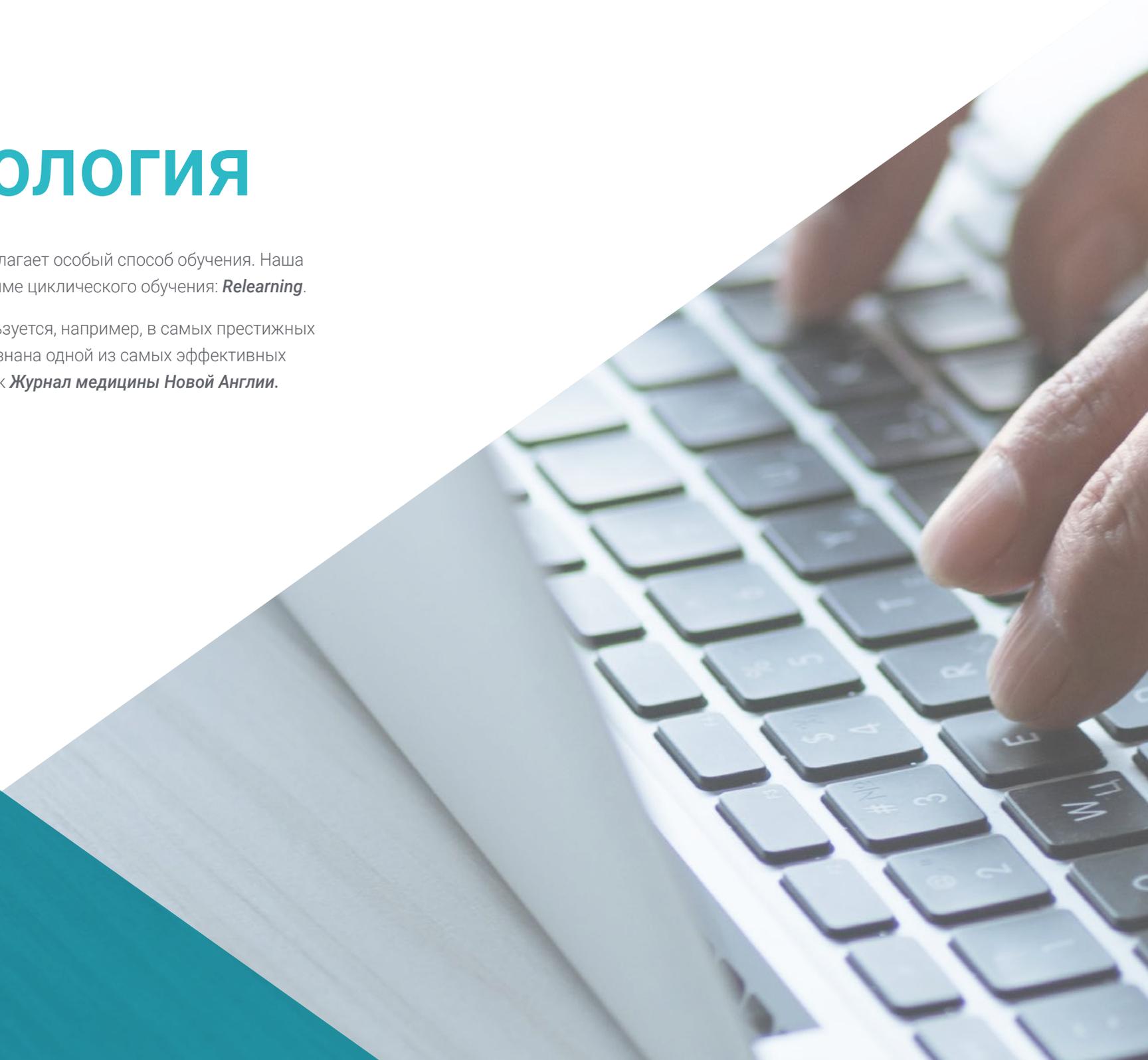
*Эта инновационная программа будет подстраиваться под вас, чтобы держать вас в курсе последних тенденций и технологий, гарантируя, что вы будете находиться на острие инноваций в области разработки программного обеспечения"*

06

# Методология

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

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



“

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

## Исследование кейсов для контекстуализации всего содержания

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

“

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



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



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

## Инновационный и отличный от других метод обучения

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

“

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

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

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

## Методология Relearning

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

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

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

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

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





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

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

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

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

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



#### Учебный материал

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

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



#### Мастер-классы

Существуют научные данные о пользе экспертного наблюдения третьей стороны.

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



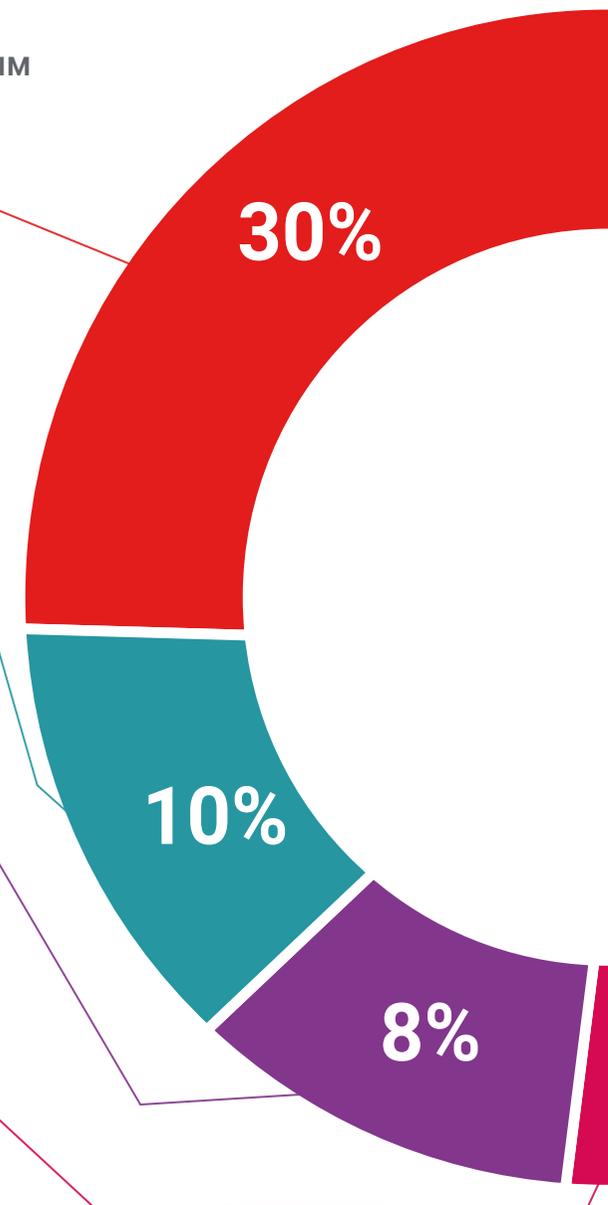
#### Практика навыков и компетенций

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



#### Дополнительная литература

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





#### Метод кейсов

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



#### Интерактивные конспекты

Мы представляем содержание в привлекательной и динамичной мультимедийной форме, которая включает аудио, видео, изображения, диаграммы и концептуальные карты для закрепления знаний. Эта уникальная обучающая система для представления мультимедийного содержания была отмечена компанией Microsoft как "Европейская история успеха".



#### Тестирование и повторное тестирование

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



07

# Квалификация

Специализированная магистратура в области разработки на Python гарантирует, помимо самого строгого и современного обучения, получение диплома об окончании Специализированной магистратуры, выдаваемого ТЕСН Технологическим университетом.



“

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

Данная **Специализированная магистратура в области разработки на Python** содержит самую полную и современную программу на рынке.

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

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

Диплом: **Специализированная магистратура в области разработки на Python**

Формат: **онлайн**

Продолжительность: **12 месяцев**



\*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.

Будущее

Здоровье Доверие Люди

Образование Информация Тьюторы

Гарантия Аккредитация Преподавание

Институты Технология Обучение

Сообщество Обязательство

Персональное внимание Институты

Знания Настоящее Качество

Веб обучение Разработка на Python

Развитие Институты

Виртуальный класс Языки

**tech** технологический университет

Специализированная  
магистратура

Разработка на Python

- » Формат: онлайн
- » Продолжительность: 12 месяцев
- » Квалификация: TESH Технологический университет
- » Расписание: по своему усмотрению
- » Экзамены: онлайн

# Специализированная магистратура Разработка на Python