Бауманский учебный центр «Специалист»

Программа курса Python для DevOps и сетевых инженеров

Код курса: **ПИТДЕВ**

Тема	Ак. часов
Модуль 1. Основы Python для инфраструктурных задач • Преимущества автоматизации: примеры из реальной практики. • Настройка среды: интерпретатор Python, менеджер пакетов, изолированные окружения. • **Практика:** Рабочее пространство. *Базовая настройка инструментов разработки.* • **Практика:** Первая программа. *Генерация сетевых диапазонов.* • **Практика:** Разработка утилиты для перечисления всех адресов в подсети. *Создание инструмента анализа подсетей.* • Введение в Docker. Базовые команды, использование docker-compose. • **Практика:** Взаимодействие Python с СУБД Postgres. *Интеграция с базой данных.* • **Практика:** VS Code с плагинами для Python, Docker, YAML. *Оптимизация среды разработки.*	**4**
Модуль 2. Структуры данных для работы с сетевыми объектами • Числовые операции: конвертация IP-адресов между форматами. • **Практика:** Текстовая обработка. *Анализ выводов CLI.* • Коллекции: организация информации об интерфейсах и их состояниях. • **Практика:** Условные конструкции. *Фильтрация оборудования.* • **Практика:** Создание парсера для преобразования вывода "show interfaces" в структурированный формат. *Преобразование неструктурированных данных.* • Словари для облачных сетей: организация данных о VPC, security groups, load balancers. • Списки подсетей в облаке: анализ IP-пула, проверка пересечений CIDR-блоков. • **Практика:** преобразование облачных network policies в словарь Python для анализа. *Анализ политик безопасности.* • Словари для метрик: хранение данных о latency, packet loss, bandwidth. • **Практика:** Модуль os и pathlib. *Работа с файловой системой.* • **Практика:** Модуль subprocess для работы с Docker. *Автоматизация Docker.* • **Практика:** Обработка ошибок. *Создание устойчивых скриптов.* • **Практика:** Утилита для проверки доступности сервиса по URL с обработкой таймаутов и ошибок. *Мониторинг сервисов.*	**6**
Модуль 3. Алгоритмы для анализа инфраструктуры • Алгоритмы поиска и фильтрации. • **Практика:** Линейный поиск. *Валидация IP-адресов.* • Бинарный поиск: быстрый поиск в отсортированных логах/configs. • **Практика:** Фильтрация по маске. *Селекция объектов.* • Сортировка: упорядочивание данных для отчетов (например, по потреблению CPU). • **Практика:** Поисковые алгоритмы. *Определение принадлежности к подсетям.* • Упорядочивание: систематизация физических адресов. • **Практика:** Анализ текста. *Парсинг логов.* • **Практика:** Утилита для статистики по сетевым инцидентам. *Анализ инцидентов.* • **Практика:** Функция диагностики доступности сетевого оборудования. *Проверка оборудования.* • **Практика:** Алгоритмы распределения нагрузки. *Имплементация балансировщиков.* • Round-robin для балансировки трафика. • Least connections для выбора наименее нагруженного сервера.	**6**

Модуль 4. WEB API для управления облачной и сетевой инфраструктурой **Тема 4.1: Основы HTTP и REST.** • Методы GET, POST, PUT, DELETE. • Коды ответа. Заголовки. **Тема 4.2: Библиотека requests** • Установка и настройка. • Выполнение запросов GET (получение данных), POST (создание), PUT(обновление). • **Практика:** Работа с заголовками, параметрами, телом запроса (JSON). *Формирование HTTP-запросов.* • **Практика:** Обработка ответов, проверка статус-кода. *Анализ API-ответов.* **Тема 4.3: Аутентификация в API** • **Практика:** Токены (Bearer Token). *Имплементация аутентификации.* • Basic Auth. • JWT Tokens. • **Практика:** Программа для получения статистики работающих интерфейсов через API. *Сбор метрик через API.*	**6**
Модуль 5. Объектно-ориентированный подход к управлению оборудованием • **Практика:** Создание классов. *Проектирование модели сетевого узла.* • **Практика:** Функциональность объектов. *Методы подключения и опроса.* • Иерархия классов: специализированные типы устройств. • **Практика:** Защита данных. *Безопасное хранение аутентификационной информации.* • **Практика:** Реализация класса маршрутизатора. *Создание объекта сетевого устройства.*	**6**
Модуль 6. Параллельная обработка сетевых устройств • Принципы параллелизма: одновременный опрос оборудования. • Библиотека asyncio: организация конкурентных подключений. • **Практика:** Асинхронные функции. *Параллельная диагностика и сбор данных.* • **Практика:** Управление процессами. *Обработка сбоев и таймаутов.* • **Практика:** Одновременная проверка состояния группы устройств. *Массовый опрос оборудования.* • **Практика:** Asyncio: параллельные HTTP-запросы к API. *Конкурентные API-вызовы.* • **Практика:** Асинхронная проверка доступности списка эндпоинтов сервиса. *Массовая проверка сервисов.*	**6**
Модуль 7. Полезные дополнительные модули для работы • Библиотека ipaddress: для продвинутой работы с IP-сетями в облаке. • pyyaml + json: для работы с конфигами инфраструктуры. • pandas: для анализа больших объемов сетевых метрик. • Модуль Docker. Работа с контейнерами и образами.	**6**
Аудиторная нагрузка в классе с преподавателем	**40 ак.ч. + 40 ак.ч. самостоятельно**

По окончании обучения на курсе проводится итоговая аттестация. Аттестация проводится в виде теста на последнем занятии или на основании оценок практических работ, выполняемых во время обучения на курсе.

Контакты центра

Для частных лиц
+7 (495) 232-32-16

Для организаций
+7 (495) 780-48-44

E-mail
info@specialist.ru