Полная программа

Clojure Developer

Clojure Developer

Длительность курса: 136 часов

Модуль 1. Основы языка Clojure

Тема 1

Знакомство с Clojure, идеология, синтаксис и основные особенности

Цель занятия

познакомиться с синтаксисом языка, основными типами данных;
научиться решать простые алгоритмические задачи;
запустить REPL и выполнить в нем код.

Краткое содержание

поговорим о том, что из себя представляет язык Clojure,
обсудим предисторию его появления и основную идеологию;
на примерах разберём синтаксис языка, типы данных,
научимся определять переменные и функции,
напишем первую программу - hello word;
установим инструменты разработчика (Clojure CLI, IDE) и научимся запускать код в REPL.

Домашние задания

Установка и настройка окружения для работы с Clojure. Запуск первой программы

Цель

Подготовить необходимое окружение на своём компьютере для дальнейшей работы с Clojure.
Научиться записывать простые выражение на Clojure и выполнять код в REPL.

Тема 2

Leiningen, создаём первый проект, углубляемся в синтаксис, работа со строками

Цель занятия

познакомиться с Leiningen, организацией Clojure проекта;
научиться работать с ветвлениями в коде;
научиться выполнять операции над строками;
начать работу над первым приложением.

Краткое содержание

познакомимся с инструментом Leiningen;
изучим его API и конфигурацию;
создадим первый проект с помощью Leiningen;
обсудим разделение кода на namespaces, определение точки входа приложения;
научимся делать "ветвление" в коде с помощью функций if, case, cond;
обсудим понятия identity и equality, функции сравнения (=, >, and, or и тд);
научимся и потренируемся выполнять основные операции со строками;
напишем первое приложение - URL Shortener.

Домашние задания

Установка и настройка Leiningen. Решение задачи "палиндром"

Цель

Установить Leiningen и попрактиковаться работать с ним.
Решить задачи и попрактиковаться работать со строками в Clojure.

Тема 3

Функциональное программирование в Clojure, функции высшего порядка, рекурсия и композиция

Цель занятия

научиться создавать собственные функции и использовать функции из стандартной библиотеки Clojure;
обсудить и использовать такие техники функционального программирования как рекурсия, композиция, функции высшего порядка.

Краткое содержание

поговорим про принципы функционального программирования;
способы создания функций (именованные и анонимные функции, литерал);
перегрузка аргументов функции (multi-arity);
обсудим несколько вариантов вызова функции (apply, partial, complement);
научимся использовать функции высшего порядка и устанавливать pre/post conditions;
разберём на практических примерах как использовать композицию функций, рекурсию, цикл loop и функции из стандартной библиотеки Clojure.

Тема 4

Персистентные структуры данных в Clojure, работа с коллекциями

Цель занятия

познакомиться с персистентными структурами данных, научиться работать с иммутабельными коллекциями.

Краткое содержание

поговорим о том, что такое персистентные структуры данных, как они устроены "под капотом", что такое shallow copy;
рассмотрим разные виды коллекций в Clojure, такие как списки, вектора, множества и хэш мапы;
обсудим их отличия и ситуации, где они наиболее применимы;
научимся использовать destructuring для доступа к элементам и использовать коллекции в качестве функций;
попрактикуем основные техники работы с коллекциями, таким как обход, фильтрация, сортировка и агрегация коллекций.

Домашние задания

Работа с Clojure коллекциями, циклами и рекурсией

Цель

Практика использования полученных знаний для решения

алгоритмических задач.
Тренировка в использовании функций стандартной библиотеки, рекурсии и коллекций.

Тема 5

Использование внешних Clojure библиотек, взаимодействие с Java классами

Цель занятия

научиться подключать и использовать внешние библиотеки в своём проекте;
познакомиться с импортом и использованием Java классов;
научиться перехватывать исключения.

Краткое содержание

научимся с помощью Leiningen управлять зависимостями проекта, а именно подключать внешние библиотеки из maven, создавать локальные версии библиотек для быстрого тестирования, используем плагин для подключения библиотек из S3;
научимся использовать профили для подключения библиотек в разных окружениях;
обсудим исключения в коде, как их обрабатывать и использовать для управления и выполнения программы;
научимся импортировать и использовать Java классы.

Тема 6

Взаимодействие с внешним миром, основные IO операции

Цель занятия

научиться использовать базовые IO операции для работы с файловой системой.

Краткое содержание

поговорим о том, что такое стандартный поток ввода/вывода в Clojure, как считывать и записывать информацию;
познакомимся со встроенным классом System и рассмотрим его API;
научимся создавать, читать и записывать информацию в файлы на диске, работать с директориями, обрабатывать файлы большого размера с использованием стримов;
научимся работать с таким форматом файлов как EDN.

Домашние задания

База данных покупок пользователей

Цель

Попрактиковаться работать с файловой системой в Clojure проекте.

Тема 7

Регулярные выражения, очереди и array comprehension

Цель занятия

познакомиться с регулярными выражениями, методом построения последовательностей array comprehension и семейством do функций.

Краткое содержание

научимся работать с регулярными выражениями в Clojure;
поработаем с такой встроенной структурой данных как очередь;
посмотрим на устройство array comprehension в Clojure;
рассмотрим функции семейства do (doseq, do-run, doall, dotimes).

Модуль 2. Продвинутые концепции

Тема 1

Метаданные в Clojure, тестирование Clojure кода

Цель занятия

познакомиться с новой концепцией в Clojure - метаданные;
рассмотреть тестовый фреймворк - clojure.test и попрактиковаться в написании тестов.

Краткое содержание

поговорим о том, что такое метаданные в Clojure, для чего они применяются и как их использовать в своих программах;
узнаем как использовать type-hints для обхода рефлексии;
познакомимся с clojure.test - фреймворком для тестирования кода;
попрактикуемся в написании тестов с использованием fixtures, match-combinatorics;
поговорим о разделении тестов с помощью test-selectors.

Домашние задания

Тесты для URL Shortener

Цель

Потренироваться в проектировании и написании тестов.

Тема 2

Полиморфизм в Clojure. Часть 1

Цель занятия

познакомиться с видами полиморфизма в Clojure;
попрактиковаться в использовании мультиметодов.

Краткое содержание

поговорим про виды полиморфизма в целом и в Clojure в частности;
рассмотрим примеры "закрытого полиморфизма" с использованием case, cond, паттерн матчинга, библиотеки clojure.core.match;
поговорим о наиболее важном "открытом полиморфизме" с использованием мультиметодов;
обсудим как использовать ad-hoc иерархии совместно с мультиметодами.

Тема 3

Полиморфизм в Clojure. Часть 2

Цель занятия

познакомиться со вторым методом использования "открытого" полиморфизма в Clojure с использованием объектов и протоколов.

Краткое содержание

поговорим про типы объектов в Clojure;
потренируемся создавать собственные типы объектов через deftype и defrecord;
обсудим их отличия и случаи их применения;
научимся добавлять поведение для собственных объектов и расширять поведение встроенных объектов с использованием протоколов;
научимся создавать анонимные объекты через reify, proxy;
поговорим как менять поведение объектов через метаданные.

Домашние задания

Получение информации из ID3v2-тегов в mp3-файлах

Цель

Попрактиковаться использовать полиморфизма для обобщения кода алгоритма и предоставления возможностей для расширения возможностей.

Тема 4

Ленивые коллекции и бесконечные последовательности

Цель занятия

познакомиться с ленивыми коллекциями;
познакомиться с transducers и попрактиковаться в их использовании.

Краткое содержание

поговорим о том, как устроена sequence abstraction;

поработаем с ленивыми коллекции, понятии отложенных вычислений, научимся создавать ленивые коллекции и использовать в работе с большими объёмами данных;
поговорим о бесконечных последовательностях, как о подвиде ленивых коллекций;
поговорим о том, что такое transducers и как их использовать с ленивыми коллекциями.

Тема 5

Опциональная типизация с помощью clojure.spec и дополнительные возможности библиотеки

Цель занятия

познакомиться с библиотекой clojure.spec;
попрактиковаться решать прикладные задачи валидации, трансформации и генерации данных с её помощью.

Краткое содержание

познакомимся с библиотекой clojure spec;
опишем с помощью spec структуры данных, научимся валидировать данные с помощью написанных спецификаций;
поговорим о рекурсивных спецификациях;
научимся трансформировать данные через конформеры;
научимся парсить данные на основе спецификаций;
научимся генерировать сэмпловые данные по спецификациям;
используем spec для генеративного тестирования.

Тема 6

HTTP запросы, библиотека clj-http

Цель занятия

научиться работать с протоколом HTTP в Clojure;
познакомиться с библиотекой clj-http.

Краткое содержание

подключим библиотеку clj-http;
посмотрим детально на её API;
потренируемся выполнять HTTP запросы;
начнем реализацию приложения WEB Scrapper с использованием библиотеки.

Тема 7

Разработка WEB сервисов на Clojure

Цель занятия

познакомиться с ring стэком;
разработать собственный WEB сервис.

Краткое содержание

познакомимся с ring стэком для разработки WEB сервисов;
поговорим об архитектуре ring стэка, понятиях адаптера и middleware;
используем jetty адаптер для запуска сервиса;
подключим библиотеку compojure для роутинга запросов в сервисе;
сделаем простой WEB сервис для приложения URL Shortener.

Домашние задания

WEB-сервис для генерации фейковых данных

Цель

Попрактиковаться с разработке WEB-сервисов.

Модуль 3. Параллельное и конкурентное программирование

Тема 1

Ссылочные типы данных, синхронизация состояния между потоками

Цель занятия

познакомиться с ссылочными типами данных, примитивами для работы с многопоточностью;
детальней изучить переменные и их API.

Краткое содержание

рассмотрим на примерах доступные ссылочные типы данных в Clojure: var, ref, atom;
поговорим о том, как они реализованы (через STM), как использовать watchers и validators;
изучим как можно изменять состояния ссылочных типов через reset! и swap! и синхронизировать состояние между потоками с помощью dosync;
научимся работать с динамическими переменными и использовать конструкцию binding;
посмотрим как переопределять указатели через with-redefs.

Тема 2

Многопоточное программирование в Clojure

Цель занятия

познакомиться с основными абстракциями многопоточного программирования в Clojure;
написать первое многопоточное приложение.

Краткое содержание

поговорим о том, как создаются потоки в Clojure, как ими управлять из основного потока выполнения (join, shutdown hook);
рассмотрим основные Clojure абстракции для работы с потоками: future, promise, agent;
рассмотрим пример координации выполнения потов с помощью locking;
начнём разработку многопоточного приложения - Redis.

Домашние задания

Многопоточный анализатор логов веб-сервера

Цель

Попрактиковаться в написании многопоточных приложений и синхронизации между потоками на Clojure.

Тема 3

Параллельное выполнение кода

Цель занятия

углубиться в тему многопоточного программирования;
научиться оптимизировать задачи, используя параллельные вычисления;
обсудить thread-pool.

Краткое содержание

поговорим о том, какие есть встроенные возможности для выполнения параллельных вычислений в Clojure: pmap, clojure.core.reducers;
поговорим о том, что такое thread pools, как они реализованы;
поработаем с библиотекой claypool в качестве практики;
рассмотрим встроенные JVM классы для многопоточного программирования, и такую концепцию как виртуальные потоки;
применим полученные знания для работы над приложением Redis.

Тема 4

Асинхронное выполнение кода

Цель занятия

рассмотреть библиотеку core.async;
доработать приложение WEB Scrapper с помощью core.async.

Краткое содержание

поговорим об отличиях параллельного и асинхронное выполнение кода, корутинах;
подключим библиотеку core.async, научимся использовать каналы, буферы и go блоки;
посмотрим как использовать пайплайны;
добавим асинхронные воркеры в приложение WEB Scrapper с помощью core.async.

Домашние задания

Работа в асинхронном стиле

Цель

Подобные API часто встречаются в реальных проектах и умение работать с такими сервисами эффективно — практически полезный навык.
Core.async является классическим инструментом, который используют в такого рода задачах, и данное задание будет хорошей практикой использования этой библиотеки.

Тема 5

Реактивное программирование в Clojure

Цель занятия

рассмотреть две дополнительные модели программирования;
попрактиковаться в их использовании.

Краткое содержание

обсудим еще одну модель программирования - реактивное программирование;
создадим собственную модель observable;
подключим библиотеку rx и научимся управлять потоками событий с её помощью;
поговорим об акторной модели и библиотеке AKKA;
обсудим в каких задачах стоит использовать различные модели программирования.

Тема 6

QA. Практическое занятие

Цель занятия

применить и структурировать полученные знания.

Краткое содержание

доработаем приложение Redis;
попрактикуемся использовать различные модели программирования.

Модуль 4. Data DSL и расширение возможностей Clojure

Тема 1

Работаем с деревьями и графами

Цель занятия

познакомиться с библиотеками clojure.walk и clojure.zip;
научиться работать с древовидными структурами данных.

Краткое содержание

поговорим в каких задачах используются графы и двоичные деревья;
познакомимся с инструментами для работы со сложными структурами данных, такими как clojure.walk и clojure.zip;
реализуем обход графа в глубину и обход в ширину;
научимся работать с виртуальными деревьями.

Домашние задания

Дерево директорий. Подсчет размера директорий

Цель

Попрактиковаться работать с древовидными структурами данных, строить и делать обход таких структур с помощью clojure.zip.

Тема 2

Основы Data DSL

Цель занятия

познакомиться с понятием Data DSL;
рассмотреть примеры использования.

Краткое содержание

познакомимся с понятием data DSL;
обсудим практические задачи, где применяется данный подход;
разберём примеры использования data DSL для описания доменной области;

поговорим про практическое применение в реальных проектах на примере compojure routing, hiccup, malli.

Тема 3

Data DSL практика

Цель занятия

попрактиковаться в использовании Data DSL;
написать свой комбинаторный парсер.

Краткое содержание

поговорим о том, что такое комбинаторные парсеры и context-free грамматики;
напишем собственный HTML парсер;
познакомимся с библиотекой instaparse;
посмотрим на реальные проекты основанные на комбинаторных парсерах;
добавим HTML парсер в приложение WEB Scrapper.

Тема 4

Основы datalog и “triplestore” базы данных

Цель занятия

рассмотреть еще один Data DSL;
познакомиться с альтернативными базами данных в Clojure.

Краткое содержание

познакомимся с распространенным в Clojure Data DSL - Datalog;
поговорим про альтернативные базы данных, т.н. triplestore на примере Datascript и Datomic;
научимся писать pull запросы, работать с графами сущностей, использовать рекурсивные правила.

Тема 5

Расширяем возможности языка Clojure, макросы

Цель занятия

познакомиться с макросами в Clojure.

Краткое содержание

познакомимся с методами расширения языка с помощью макросов;
разберём базовые принципы и синтаксис макросов (reader literals);
обсудим концепцию код = данные;
напишем несколько собственных макросов;
посмотрим на реализацию встроенных макросов в clojure.core;
поговорим про отладку макросов.

Тема 6

Введение в ClojureScript

Цель занятия

познакомиться с ClojureScript;
написать фронтэнд для приложения.

Краткое содержание

познакомимся с диалектом языка и компилятором ClojureScript;
настроим проект с помощью shadow-cljs;
посмотрим как компилировать код для разного окружения (браузер, Node.js, тесты);
обсудим дополнения и отличия от Clojure;
напишем фронтэнд для приложения URL Shortener с использованием библиотеки reagent.

Модуль 5. Работа с внешними системами

Тема 1

Работа с реляционными базами данных

Цель занятия

научиться работать с SQL базами данных используя Clojure библиотеки.

Краткое содержание

познакомимся с протоколом jdbc и подключим библиотеку next-jdbc;
подключимся к базе данных и научимся выполнять транзакции;
научимся использовать библиотеку honey-sql для работы с языком запросов SQL;
создадим базу данных для проекта URL Shortener, напишем SQL запросы для реализации бизнес логики приложения.

Домашние задания

Работа с данными по покемонам в базе Postgres. Конструктор аналитических запросов к базе данных

Цель

Научиться работать с распространёнными библиотеками для работы с реляционными базами данных.

Тема 2

Работа с брокерами сообщений

Цель занятия

научиться использовать брокер сообщений Kafka в Clojure проекте.

Краткое содержание

поговорим про использование Kafka в Clojure приложениях;
подключим библиотеку Jackdaw для работы с Kafka API;
создадим простое приложение с использованием producer и consumer;
научимся использовать Kafka Streams API;
расширим функционал приложения WEB Scrapper с использованием Kafka.

Тема 3

Архитектура больших Clojure проектов

Цель занятия

применить принципы гексагональной архитектуры в Clojure проекте;
познакомиться с библиотекой integrant и фреймворком duct.

Краткое содержание

поговорим про гексагональную архитектуру и её составные части, такие как сервисы, порты, адаптеры и доменный слой;
научимся разбивать проект на модули и слои;
научимся работать с библиотекой integrant и фреймворком duct;
попрактикуемся в использовании dependency injection;
научимся использовать гексагональную архитектуру для реорганизации кода приложения URL Shortener.

Тема 4

Запуск Clojure приложения в продакшен

Цель занятия

подготовить приложение для запуска в production.

Краткое содержание

поговорим о подготовке Clojure кода для сборки в production;
запакуем проект в uberjar для деплоя;
поговорим о конфигурации JVM при запуске приложения;
посмотрим на новые возможности сборки в бинарный файл с graalvm;
обсудим вопросы поддержки приложения в post production (логгирование, мониторинг).

Домашние задания

Применение принципов hexagonal architecture в проекте на выбор (WEB Scrapper || Pokemon app)

Цель

Научиться использовать принципы hexagonal architecture в реальном проекте.
Познакомиться с библиотекой integrant, фреймворком Duct.

Тема 5

Отладка Clojure приложений и оптимизация производительности

Цель занятия

научиться определять проблемные, в плане производительности, места в коде;
рассмотреть распространенные методы улучшения производительности.

Краткое содержание

поговорим про оптимизацию производительности приложений;
познакомимся с инструментом VisualVM для нахождения узких мест в производительности;
обсудим методы оптимизации кода: локальное использование transients, unboxing чисел для математических операций, использование java arrays.

Тема 6

Скрипты на Clojure, Babashka runtime

Цель занятия

рассмотреть проект Babashka;
научиться писать скрипты на Clojure.

Краткое содержание

познакомимся с проектом babashka для написания скриптов на Clojure;
научимся создавать command line interface для своих скриптов,
автоматизировать рутинные задачи с использованием babashka tasks;
используем babashka runtime для запуска кода на AWS Lamoda.

Тема 7

QA. Практическое занятие

Цель занятия

применить и структурировать полученные знания.

Краткое содержание

доработаем приложение WEB Scrapper.

Модуль 6. Проектный модуль

Тема 1

Выбор темы и организация проектной работы

Цель занятия

выбрать и обсудить тему проектной работы;
спланировать работу над проектом;
ознакомиться с регламентом работы над проектом.

Краткое содержание

правила работы над проектом и специфика проведения итоговой защиты;
требования к результату проекта и итоговой документации.

Домашние задания

Проектная работа

Цель

В этом ДЗ необходимо выбрать и утвердить в чате по ДЗ темы проекта, разработать и презентовать проект.

Тема 2

Консультация по проектам и домашним заданиям

Цель занятия

получить ответы на вопросы по проекту, ДЗ и по курсу.

Краткое содержание

вопросы по улучшению и оптимизации работы над проектом;
затруднения при выполнении ДЗ;
вопросы по программе.

Тема 3

Защита проектных работ

Цель занятия

защитить проект и получить рекомендации экспертов.

Краткое содержание

презентация проектов перед комиссией;
вопросы и комментарии по проектам.