🎬 СЦЕНАРИЙ ВИДЕО: "КОНСОЛЬНЫЙ TODO LIST НА JAVA С НУЛЯ"

ЧАСТЬ 1: ВВЕДЕНИЕ (3 минуты)

0:00–3:00

"Привет, друзья! Сегодня мы создадим с нуля консольный ToDo List на Java — простую, но невероятно полезную программу, которая поможет вам разобраться в работе с коллекциями, обработкой ввода и основами ООП."

"Этот проект — отличный способ прокачать навыки Java, научиться работать со списками задач, эффективно управлять состоянием приложения и создавать удобный пользовательский интерфейс прямо в консоли."

Наш ToDo List будет уметь:

- Добавлять новые задачи с автоматической нумерацией
- Показывать все задачи с их статусом выполнения
- Отмечать задачи как выполненные
- Удалять задачи из списка
- Работать в бесконечном цикле, позволяя управлять задачами без перезапуска программы

"Всё это мы напишем вручную, в трёх классах, чтобы вы поняли каждый шаг кода и принципы ООП!"

ЧАСТЬ 2: ТЕХНОЛОГИИ И ИНСТРУМЕНТЫ (3 минуты)

3:00–6:00

"Мы используем только стандартные возможности Java, без сторонних библиотек. Всё максимально просто и нативно."

Что применим:

- **ArrayList<Task>** — для хранения и управления списком задач
- **Scanner** — для взаимодействия с пользователем и получения команд
- **Принципы ООП** — инкапсуляция, композиция, разделение ответственности
- **Циклы while/for и switch** — для главного меню и обработки команд
- **Обработка исключений** — для корректной работы с пользовательским вводом

"Мы создадим три класса, каждый из которых будет отвечать за свою часть функциональности, что является отличной практикой правильного проектирования приложений."

ЧАСТЬ 3: ЧЕМУ ВЫ НАУЧИТЕСЬ (3 минуты)

6:00–9:00

"После просмотра видео вы получите не только готовый инструмент, но и глубокое понимание ключевых концепций Java и ООП."

Вы поймёте:

- Как правильно структурировать код с использованием ООП принципов
- Как работать с коллекциями ArrayList для хранения объектов
- Как создавать чистый и понятный пользовательский интерфейс в консоли
- Как обрабатывать пользовательский ввод и валидировать данные
- Как управлять состоянием приложения через отдельные классы

"Это идеальный мини-проект для практики логического мышления и правильного структурирования кода!"

ЧАСТЬ 4: ФУНКЦИОНАЛ TODO LIST (3 минуты)

9:00–12:00

"Таким образом, мы сделаем настоящий мини-органайзер, который работает прямо в консоли!"

Что будет у нашей программы:

- **Интуитивное меню** — простой выбор операций с задачами
- **Динамическое управление** — добавление, просмотр, выполнение и удаление задач
- **Визуальный статус** — отображение выполненных и невыполненных задач
- **Автоматическая нумерация** — каждая задача получает уникальный ID
- **Обработка ошибок** — защита от некорректного ввода
- **Непрерывная работа** — возможность работать с задачами без перезапуска

"Мы создадим систему, где каждая задача — это объект со своим состоянием, а весь список — это умная коллекция, которая умеет управлять этими объектами."

ЧАСТЬ 5: НАПИСАНИЕ КОДА (15 минут)

12:00–27:00

"Всё максимально понятно и структурировано. Мы не просто пишем код, мы разбираем его логику и архитектуру."

Блок 1: Класс Task — модель данных

- Создаём класс Task с приватными полями и публичными методами

- Реализуем принцип инкапсуляции — защищаем данные задачи
- Добавляем методы для изменения состояния задачи
- Создаём красивый toString() для отображения

Блок 2: Класс TodoList — бизнес-логика

- Создаём класс для управления коллекцией задач
- Реализуем методы добавления, отображения, выполнения и удаления
- Используем ArrayList для хранения объектов Task
- Добавляем автоматическую генерацию ID

Блок 3: Класс TodoListApp — пользовательский интерфейс

- Создаём главный класс с меню и обработкой ввода
- Реализуем бесконечный цикл программы
- Добавляем обработку исключений для стабильной работы
- Создаём удобный и понятный интерфейс

"Каждый блок мы будем писать и сразу же тестировать, чтобы вы видели результат каждого этапа!"

ЧАСТЬ 6: ЗАВЕРШЕНИЕ (3 минуты)

27:00–30:00

"Подводим итоги: Мы научились создавать структурированные приложения на Java, работать с коллекциями и объектами, реализовывать удобный пользовательский интерфейс."

- Реализовали полноценный, функциональный ToDo List с возможностью многократного использования
- На практике разобрали принципы ООП: инкапсуляцию, композицию, разделение ответственности
- Создали чистый и поддерживаемый код, который легко расширять

"Этот проект — отличная основа для дальнейшего развития. Вы можете добавить сохранение в файл, категории задач, сроки выполнения и многое другое!"

"Если тебе понравился проект, поставь лайк и подпишись, впереди будет ещё больше практических проектов на Java!"

"С вами был [твоё имя], до встречи в следующем видео, где мы создадим новый полезный инструмент на Java!"

🎯 **КЛЮЧЕВЫЕ АКЦЕНТЫ СЦЕНАРИЯ:**

Демонстрация ООП принципов:

- **Инкапсуляция** — приватные поля Task + публичные методы
- **Композиция** — TodoList содержит ArrayList<Task>, TodoListApp содержит TodoList
- **Разделение ответственности** — каждый класс решает свою задачу

Практическая польза:

- Реальный работающий инструмент для управления задачами
- Код, который можно сразу использовать и расширять
- Фундамент для более сложных проектов

Образовательная ценность:

- Понимание работы с коллекциями
- Навыки обработки пользовательского ввода
- Принципы архитектуры приложений
- Практика чистого кода и рефакторинга