🎬 **ОПИСАНИЕ СЦЕНАРИЯ ВИДЕО: "КОНСОЛЬНЫЙ СЕКУНДОМЕР НА JAVA С НУЛЯ"**

⏱️ **Общая продолжительность: ~20-25 минут**

🎯 **ЧАСТЬ 1: ВВЕДЕНИЕ (3 минуты)**

⏰ **0:00–3:00**

📝 **Сценарий:** «Привет, друзья! 👋

Сегодня мы с нуля создадим настоящий, функциональный **консольный секундомер** на Java! Это не просто учебное приложение, это отличный проект для понимания ключевых основ: работы со временем, управления состоянием и обработки пользовательского ввода.

Наш секундомер будет уметь:

- ▶ **Старт, Пауза и Сброс:** Корректно учитывать время даже после многократных пауз.
- ⏱️ **Фиксация Кругов (Lap Time):** Запоминать время прохождения каждого этапа.
- 📊 **Расчет Статистики:** Вычислять и показывать **лучшее** и **худшее** время круга.

Это отличный проект для начинающих Java-разработчиков — вы научитесь работать с системным временем, коллекциями и условными конструкциями, которые являются фундаментом любого сложного приложения.»

🔧 **ЧАСТЬ 2: ТЕХНОЛОГИИ И НАВЫКИ (3 минуты)**

⏰ **3:00–6:00**

🛠️ **Что используем:**

- 🎯 **JAVA CORE** — основа проекта
    - System.currentTimeMillis() — "пульс" программы, получение точного времени.
    - Scanner — для приема команд от пользователя (Старт, Пауза, Круг).
    - ArrayList<Long> — для хранения общего времени фиксации каждого круга.
    - LongStream — для быстрого расчета лучшего и худшего времени (Статистика).
    - **Условия if / else** — управление логикой Старт/Пауза/Сброс.
    - **Цикл while** — главный цикл работы программы, ждущий ввода команд.

🎙️ **Объяснение:** «System.currentTimeMillis() — это наш ключевой инструмент. Мы используем его, чтобы получить точный момент времени (в миллисекундах) и, вычитая его из lastStartTime, мы получаем продолжительность текущего отрезка. ArrayList будет нашей "памятью" для сохранения всех кругов. А Scanner и цикл while — это наши "глаза и уши" для взаимодействия с пользователем.»

🎓 **ЧАСТЬ 3: ЦЕЛИ ОБУЧЕНИЯ (3 минуты)**

⏰ **6:00–9:00**

Чему вы научитесь:

- 📘 **Управление состоянием:** Как правильно использовать флаги (running) и две переменные (totalElapsedMillis, lastStartTime) для корректной работы **Паузы**.
- 🧩 **Работа с коллекциями:** Как хранить последовательные результаты (laps) и вычислять разницу времени между ними.
- ⚙️ **Анализ данных:** Как использовать **Java Stream API** (LongStream.min() и max()) для мгновенного нахождения статистики кругов.
- 💡 **Форматирование:** Как преобразовать большие значения миллисекунд в читаемый формат **MM:CC.MMM**.

🧮 **ЧАСТЬ 4: ФУНКЦИОНАЛ СЕКУНДОМЕРА (3 минуты)**

⏰ **9:00–12:00**

Что умеет наш секундомер:

- ✅ **Основные команды (Меню):**
  - 1 (Старт): Запуск/Продолжение.
  - 2 (Пауза): Остановка с сохранением времени.
  - 3 (Сброс): Обнуление таймера и кругов.
  - 4 (Круг): Фиксация времени.
- ✅ **Отчеты и Статистика:**
  - Вывод времени в формате **MM:CC.MMM**.
  - Вычисление **длительности каждого круга** (getLapTime).
  - Отображение **лучшего** и **худшего** времени круга.

🎙️ **Переход:** «Теперь, когда мы точно знаем, как должна работать наша программа и какие технологии мы используем — давайте перейдём к самому интересному: напишем код с нуля и подробно разберём логику, которая позволяет секундомеру корректно считать время!»

💻 **ЧАСТЬ 5: НАПИСАНИЕ И РАЗБОР КОДА (7-9 минут)**

⏰ **12:00–~21:00**

(В этой части идет демонстрация кода: создание класса, переменных, метода main, реализация getCurrentTime, затем startTimer, pauseTimer, resetTimer и, наконец, addLap и showStats.)

🎤 **ЧАСТЬ 6: ЗАКЛЮЧЕНИЕ (3 минуты)**

⏰ **~21:00–24:00**

Текст: «Поздравляю! Вы только что с нуля создали полноценный консольный секундомер на Java.

Теперь вы умеете:

1. Управлять сложным состоянием программы с помощью флагов и двух переменных времени.
2. Корректно рассчитывать время, преодолевая сложности пауз и продолжений.
3. Использовать Java Streams для быстрого анализа данных кругов.

Это отличная база для будущих проектов, которые зависят от точного учета времени!»

Итоговый экран:

- "Исходный код в описании"
- "Подпишись, чтобы не пропустить новые уроки"
- "Поставь 👍, если хочешь продолжение"

Желаете, чтобы я написал **подробный скрипт** (текст, который нужно говорить) для одной из частей, например, для объяснения логики getCurrentTime()?