Чудесна идея! 🎯 Серия от малки, фокусирани **multithreading** проекти е отличен начин да затвърдиш знанията си и да покажеш опит. Ето ти **пълна поредица** от проектчета – по едно за всяка важна технология/концепция, която си споменавал:

**🔧 1. Basic Multithreading – Printer Sync**

**🔹 Технологии: Thread, Runnable, synchronized**

🔸 **Проект**: Класически Printer — няколко нишки отпечатват букви или думи последователно със синхронизация.  
🔸 Цел: Разбиране на synchronized, споделени ресурси, блокиране.

**🔁 2. Banking App – ReentrantLock**

**🔹 Технологии: ReentrantLock, tryLock(), finally**

🔸 **Проект**: Банкова система – потребители теглят и внасят пари в общ акаунт.  
🔸 Цел: Защита на споделени ресурси при състезание (race condition).

**🚗 3. Car Race Simulator – ExecutorService**

**🔹 Технологии: ExecutorService, FixedThreadPool, shutdown()**

🔸 **Проект**: Симулация на състезание с коли – всяка нишка е кола, която минава обиколки.  
🔸 Цел: Работа с ExecutorService, управление на нишки.

**🧮 4. Math Worker Pool – Callable & Future**

**🔹 Технологии: Callable, Future, submit(), get()**

🔸 **Проект**: Работници, които изчисляват факториели, прости числа и връщат резултат.  
🔸 Цел: Получаване на резултати от нишки.

**🕐 5. Task Scheduler – ScheduledExecutorService**

**🔹 Технологии: ScheduledExecutorService, scheduleAtFixedRate**

🔸 **Проект**: Програма, която симулира аларма/напомняне, което се изпълнява периодично.  
🔸 Цел: Работа с отложени задачи и повтарящи се задачи.

**🧱 6. Producer-Consumer – BlockingQueue**

**🔹 Технологии: BlockingQueue, ArrayBlockingQueue**

🔸 **Проект**: Един producer произвежда съобщения, няколко consumer-и ги консумират.  
🔸 Цел: Създаване на опашки и контрол на достъпа.

**🧠 7. Parallel Search – WorkStealingPool**

**🔹 Технологии: WorkStealingPool, ForkJoin, RecursiveTask**

🔸 **Проект**: Паралелно търсене на елемент в голям списък/множество.  
🔸 Цел: Работа с динамични и оптимизирани нишки.

**🗂️ 8. File Watcher – WatchService**

**🔹 Технологии: WatchService, Files, Path, ConcurrentLinkedQueue**

🔸 **Проект**: Следене на папка за нови файлове. Известие при нов .txt файл.  
🔸 Цел: Реакция в реално време, работа със събития.

**📊 9. Statistics App – Parallel Streams**

**🔹 Технологии: parallelStream, Collectors, Optional**

🔸 **Проект**: Статистика от списък с числа – средна стойност, медиана, мода.  
🔸 Цел: Функционален стил с паралелни потоци.

**🧪 10. Multithreaded Test System – JUnit + Mockito**

**🔹 Технологии: JUnit, Mockito, assert, Thread.sleep()**

🔸 **Проект**: Юнит тестове на горните класове, мокване на Logger, DatabaseService.  
🔸 Цел: Демонстрация на добри практики и покритие.

**💡 Как да го покажеш:**

* 🎥 Всяко едно мини-проектче с README.md, Demo, и Explanation.
* 📂 Организирай ги в GitHub репо java-concurrency-lab.
* 🧩 Добро въведение: „Това е лаборатория от 10 проекта, чрез които упражнявам най-често използваните подходи за многнишкова обработка в Java.“

Ако искаш — ще направим заедно първия още сега. Или мога да ти дам скелет и ти го завършваш сам. Как предпочиташ?