14. 다음으로

#0.강의/1.자바로드맵/5.자바-고급1편

1. 프로세스와 스레드 소개

- /멀티태스킹과 멀티프로세싱
- /프로세스와 스레드
- /스레드와 스케줄링
- /컨텍스트 스위칭

2. 스레드 생성과 실행

- /프로젝트 환경 구성
- /스레드 시작1
- /스레드 시작2
- /데몬 스레드
- /스레드 생성 - Runnable
- /로거 만들기
- /여러 스레드 만들기
- /Runnable을 만드는 다양한 방법
- /문제와 풀이
- /정리

3. 스레드 제어와 생명 주기1

- /스레드 기본 정보
- /스레드의 생명 주기 - 설명
- /스레드의 생명 주기 - 코드
- /체크 예외 재정의
- /join - 시작
- /join - 필요한 상황
- /join - sleep 사용
- /join - join 사용
- /join - 특정 시간 만큼만 대기
- /문제와 풀이

4. 스레드 제어와 생명 주기2

- /ReentrantLock - 활용
- /인터럽트 - 시작1
- /인터럽트 - 시작2
- /인터럽트 - 시작3
- /인터럽트 - 시작4
- /프린터 예제1 - 시작
- /프린터 예제2 - 인터럽트 도입
- /프린터 예제3 - 인터럽트 코드 개선
- /yield - 양보하기
- /프린터 예제4 - yield 도입
- /정리

5. 메모리 가시성

- /volatile, 메모리 가시성1
- /volatile, 메모리 가시성2
- /volatile, 메모리 가시성3
- /volatile, 메모리 가시성4
- /자바 메모리 모델(Java Memory Model)
- /정리

6. 동기화 - synchronized

- /출금 예제 - 시작
- /동시성 문제
- /임계 영역
- /synchronized 메서드
- /synchronized 코드 블럭
- /문제와 풀이
- /정리

7. 고급 동기화 - concurrent.Lock

- /LockSupport1
- /LockSupport2
- /ReentrantLock - 이론

- /ReentrantLock - 활용
- /ReentrantLock - 대기 중단
- /정리

8. 생산자 소비자 문제1

- /생산자 소비자 문제 - 소개
- /생산자 소비자 문제 - 예제1 코드
- /생산자 소비자 문제 - 예제1 분석 - 생산자 우선
- /생산자 소비자 문제 - 예제1 분석 - 소비자 우선
- /생산자 소비자 문제 - 예제2 코드
- /생산자 소비자 문제 - 예제2 분석
- /Object - wait, notify - 예제3 코드
- /Object - wait, notify - 예제3 분석 - 생산자 우선
- /Object - wait, notify - 예제3 분석 - 소비자 우선
- /Object - wait, notify - 한계
- /정리

9. 생산자 소비자 문제2

- /Lock Condition - 예제4
- /생산자 소비자 대기 공간 분리 - 예제5 코드
- /생산자 소비자 대기 공간 분리 - 예제5 분석
- /스레드의 대기
- /중간 정리 - 생산자 소비자 문제
- /BlockingQueue - 예제6
- /BlockingQueue - 기능 설명
- /BlockingQueue - 기능 확인
- /정리

10. CAS - 동기화와 원자적 연산

- /원자적 연산 - 소개
- /원자적 연산 - 시작
- /원자적 연산 - volatile, synchronized
- /원자적 연산 - AtomicInteger
- /원자적 연산 - 성능 테스트

- /CAS 연산1
- /CAS 연산2
- /CAS 연산3
- /CAS 락 구현1
- /CAS 락 구현2
- /정리

11. 동시성 컬렉션

- /동시성 컬렉션이 필요한 이유1 - 시작
- /동시성 컬렉션이 필요한 이유2 - 동시성 문제
- /동시성 컬렉션이 필요한 이유3 - 동기화
- /동시성 컬렉션이 필요한 이유4 - 프록시 도입
- /자바 동시성 컬렉션1 - synchronized
- /자바 동시성 컬렉션2 - 동시성 컬렉션
- /정리

12. 스레드 풀과 Executor 프레임워크1

- /스레드를 직접 사용할 때의 문제점
- /Executor 프레임워크 소개
- /ExecutorService 코드로 시작하기
- /Runnable의 불편함
- /Future1 - 소개
- /Future2 - 분석
- /Future3 - 활용
- /Future4 - 이유
- /Future5 - 정리
- /Future6 - 취소
- /Future7 - 예외
- /ExecutorService - 작업 컬렉션 처리
- /문제와 풀이
- /정리

13. 스레드 풀과 Executor 프레임워크2

- /ExecutorService 우아한 종료 - 소개

- /ExecutorService 우아한 종료 - 구현
- /Executor 스레드 풀 관리 - 코드
- /Executor 스레드 풀 관리 - 분석
- /Executor 전략 - 고정 풀 전략
- /Executor 전략 - 캐시 풀 전략
- /Executor 전략 - 사용자 정의 풀 전략
- /Executor 예외 정책
- /정리

로드맵 소개

실전 자바 로드맵

- 김영한의 자바 입문 - 코드로 시작하는 자바 첫걸음 (오픈)
- 김영한의 실전 자바 - 기본편 (오픈)
- 김영한의 실전 자바 - 중급 1편 (오픈)
- 김영한의 실전 자바 - 중급 2편 (오픈)
- 김영한의 실전 자바 - 고급 1편 - 멀티스레드와 동시성

김영한의 실전 자바 - 고급편 (예정)

- 고급 2편: IO, 네트워크, 리플렉션, 애노테이션
- 고급 3편: 람다, 스트림, 모던 자바

실전 데이터베이스 로드맵

- 자바 로드맵 이후

백엔드 개발자 로드맵 소개

김영한 백엔드 개발자 자바 스프링 JPA 실무 로드맵

백엔드 개발자 로드맵 소개 영상 링크: https://youtu.be/ZgtvcyH58ys

스프링 완전 정복 로드맵

- 스프링을 완전히 마스터 할 수 있는 로드맵
- URL: https://www.inflearn.com/roadmaps/373

스프링 부트와 JPA 실무 완전 정복 로드맵

- 최신 실무 기술로 웹 애플리케이션을 만들어보면서 학습
- URL: https://www.inflearn.com/roadmaps/149

하고 싶은 이야기

대나무 죽순 이야기

- 씨앗을 심으면 수 년간 싹이 보이지 않고, 수 년간 뿌리를 단단히 내림
- 싹이 나온 후에는 하루에 수십cm씩 쑥쑥 자람
- 6주 후에는 30m까지 성장
- 뿌리를 단단히 내려둔 덕분에 이후에 크게 성장

결국은 기본기

- 제가 빠르게 성장할 수 있었던 이유를 돌아보면 자바, DB 같은 기본기를 잘 다진 덕분
- 실무에서 사용하는 기술들 대부분이 기본기를 활용하는 기술들
- 자바 백엔드 웹 애플리케이션 = 객체 지향 + 멀티스레드 + 네트워크(HTTP)
- 실무에서 사용하는 기술들은, 우리가 기본기를 알 것이라고 가정하고 설명함
- 기본기가 약하면 활용 기술들은 이해 자체가 어려울 수 있음
- 기본기를 학습하는 시간이 돌아가는 것 같지만, 결과적으로 더 빠른 지름길일 수 있다.

- 기본기가 탄탄하게 되어 있으면, 이해도 성장도 빠르다.

하고 싶은 이야기 정리 링크

개발 인생 전반의 이야기

EO 인터뷰 영상

- 한국 개발자 최고 1타강사 김영한의 인생 [1부]: https://youtu.be/_HTj5b59Em0
- 한국 개발자 최고 1타강사 김영한의 인생 [2부]: https://youtu.be/MNyNRraMU8Y

개발바닥 - 시골 청년 개발왕 되다

- 1편: https://youtu.be/Pb69UQ6f8n0
- 2편: https://youtu.be/b4QP5RsuJts
- 3편: https://youtu.be/l0h1pQ96u2g

취업과 이직에 대한 고민

인프콘 - 어느 날 고민 많은 주니어 개발자가 찾아왔다, 성장과 취업, 이직 이야기

- https://youtu.be/QHlyr8soUDM

인프런 최초 20만 명 달성 기념 QA

- https://youtu.be/psXdWq008DA

인프런 최초 30만 명 달성 기념 QA

- https://inf.run/81ogv