**[Java]**

* 객체지향(oop)의 정의와 장단점
* 제너릭, 컬렉션, 배열
* Stack/heap 자료구조의 이해
* 오버라이드 오버라이딩
* 오버로드 오버로딩
* 클래스 상속 인터페이스
* 접근제한자
* 스레드 vs 프로세스
* Call by Reference / Call by Value
* Static 과 void
* 람다식

**[ Spring ]**

* 프레임 워크의 원리

-aop – psa – ioc/di

* Interface VS Abstract
* Web Application Server(WAS)와 WebServer의 차이
* Cors
* 아파치 VS 톰켓 (멀티쓰레드? 멀티 프로세스?)
* 서블릿(servlet)
* 디자인 패턴 (MVC/싱글톤)
* Parameter 처리
* 데이터 엑세스 층의 설계 구현

**[ DB ]**

* Join alter update
* DDL DML DCL DBMS
* 트리거
* 인덱스

\*이 키워드로 검색해서 찾으시면 웬만한 기술면 접은 다 뽀갤 수 있습니다ㅋㅋ 면접 파이팅

<https://whimsical.com/backend-TB9iYJtWPFrrAA2TEpmGsL> (pw 3981)