Summary of Questions

1. 자기 소개 (아래 시나리오대로 소개)

A. **다년 간** STB/OTT/Mobile Android 제품에 탑재되는 **Android UI / Service Application** 업무담당
B. **Android 개발 팀으로 시작**, **Git**, **Jenkins**, Opengrok, **Gerrit** 시스템 활용
C. 이슈 관리는 **Jira** / **Redmine** / **Confluence** 시스템 활용
D. Embedded Android, **AOSP 소스에 익숙**, App 포팅, 빌드 스크립트 작성
E. **Broadcom / Amlogic / Telechips Chipset 사 AOSP SDK 위에서 작업한 경험**, JNI 작업
F. STB/OTT/Mobile Platform에서 **Android UI 개발한 앱, Launcher, Wizard, Settings, VOD, Radio, SW Testing App** 개발
G. UI App 개발, **UX/UI 시나리오 팀 협의** 하에 진행
H. Service 앱으로는 **RCU** Service, 3rd Party 업체에 API 제공을 위한 **API** Service, **IPTV** Service 개발 (**Java Docs** / **Markdown** 포맷의 문서 제공)
I. **협업** 경험, **출장**, 국내 **3rd Party 업체**와의 업무 (RCU업체)
J. **외국어 (영어)** 사용 기회가 잦음, **Vietnam / Spain** 개발자와 Communication

2. Android Native App을 구현한 것인가?

A. 그렇다. Android Native App 구현 / 유지보수를 담당해왔다
B. Android Native UI App과 Service App을 개발해왔다

3. 개발 업무가 아닌 프로젝트 매니저 역할 (Leading)에 관심 있는가?

A. 관심 있다. 개발을 하면서 나중에는 그 경험을 발판 삼아 팀을 이끌어 나가고 싶다
B. 초기 개발부터 릴리즈까지 개발을 해왔으며 그 과정에서 쌓아온 안드로이드 지식을 기반으로 해서 적재적소에 사람을 배치하거나 개발 과정 중 필요한 것들이 무엇인지 알기 때문에 안드로이드 개발 매니저/팀장을 하고 싶다

4. Flutter / React Native 관심 있는가?

A. 관심 있다. 개발 환경 구축 정도는 해봤으며 개인 GitHub에 올려 놨다.

5. IOS 개발 경험은 있는가?

A. 없다. 기회가 된다면 경험해보고 싶다
B. IOS로 넘어가야 되는 상황이면 업무 외 시간을 활용해서라도 IOS 개발 관련해서 경험을 쌓겠다

6. OTT/STB Android 개발 경험이 주 경험인데 모바일 개발 관련해서 본인 어필을 할 만한 것이 있는가?

A. 기본적인 Android API, Tool 그리고 빌드 방법이 OTT/Mobile/STB 다 동일하다

B. 기본적인 안드로이드 아키텍쳐 자체는 동일하다.
C. 개인 프로젝트로 Mobile App 개발을 꾸준히 진행 중이다. 다른 것은 UX/UI 면이라 생각한다.
D. 모든 안드로이드 컴포넌트를 사용할 줄 안다
E. 초기 개발 단계부터 릴리즈 까지 가능

7. 결제 모듈 시스템 연동 경험 있는가?

A. 개념
  A. 결제 모듈
    i. **Google/Apple** : 디지털 **콘텐츠** 구매 (게임, 음악, 이모티콘 등)
    ii. **PG 사** : **실물** 결제 (음식, 온라인 쇼핑 등) – 카카오페이와 유사
      1. 카카오페이는 **TCC라고 하는 MSA에서 쓰이는 트랜잭션 방식**

  B. 결제 모듈 처리 개념
    i. **트라이 – 컨펌, 캔슬** (트라이 ➔ 성공 ➔ 컨펌 / 캔슬)
    ii. **사용자 입장**에서의 개념 : **Ready, Approve**
      1. 카카오 페이 **Ready 요청** 시도
      2. 카카오 페이는 결제 트라이, 해당 결제 건에 대한 **Unique ID** 발행
      3. 사용자에게 **Response** 제공 (**결제 Scheme**을 띄우는 응답)
      4. 사용자는 카카오톡 결제 Scheme이 Invoke되고 **결제를 시도**
      5. 결제 완료 시 카카오톡 자체 리다이렉트 Path로 진행
      6. 결제 완료되면 **실제 결제한 앱으로 다시 결과를 리다이렉트** 함
      7. 그 Response를 받은 결제한 앱에서 **Approve** 요청을 보냄
      8. **결제가 유효한지 체크**
      9. 결제 완료 그 뒤 결제 앱 **비즈니스 로직**을 타게 됨

  C. 결제 종류

➔ **단건 / 정기**

⇨ 정기결제에 대한 추가지식

- 결제할 때 **Sid** (Subscription id)를 받아와서 Ready/Approve가 아니라 **바로 결제 진행**함. (이유 : **사용자가 승인하는 Step이 없기 때문에**) // Sid를 받아오는 시점에 사용자 승인을 받고 그 id로 계속 결제하는 방식

8. APK 개발 ~ 배포 (플레이 스토어 등재) 까지의 Step 설명

A. 개발 착수 업무 (아이디어 구체화, 요구사항 정립)
B. UX/UI Design (개발팀과 디자인 프로토타입 결과물 공유)
C. 앱 구현 기술 설계
D. 개발 방식 채택 (OS Native / Cross Platform / Hybrid)
E. 개발 언어 선택 / 데이터 구조 정립

F. 앱 개발 시작 (계획 수립 / 개발 / 테스트 (기능, 성능, 엣지 케이스, 장치별, 사용성, 최종))
G. 앱 상용화 (업데이트 주기 계획 / 업데이트 / 개발자 인증 / 자동화 / 호스팅 환경 / 앱 정보 최신화 / 서버 시스템 / Apple or Google App Store)
    A. 구글 계정 생성
    B. 안드로이드 개발자는 만 18세 이상이어야 등록 가능
    C. 등록 수수로 발생 USD 25
    D. 구글 플레이 개발자 등록
H. 상용화 이후 모니터링
    A. 시간 측정
    B. Crash
    C. UI 응답 시간
    D. 앱스토어 관리
    E. 배터리 소모량 측정
    F. 모바일 데이터 소비량 측정

9. 스트레스를 많이 받는 타입인지?

A. 취미 생활로 해소, 기타 연주, 운동으로 해소
B. 스트레스 받는 그 순간이 힘들 수 있으나 항상 모든 것은 사소한 일이다 라고 생각하면서 마인드 컨트롤 하려고 노력한다.

10. 술/담배 하는지?

A. 안한다 / 한다 (술은 즐기나 흡연은 하지 않는다)

11. 영어는 어느 수준까지 하는가?

A. 메일 또는 구두로 개발자와 의사소통 가능한 정도

12. 이직/퇴사를 결심한 이유는 무엇인가?

A. **Mobile 앱 개발 영역을 특화** 시키기 위함
B. **최신 트렌드, 최근 모든 IT Service들의 창구가 Mobile로 집중**, STB/OTT에 이어 **트렌드를 쫓아가기** 위함
C. Mobile 앱 개발 뿐만 아니라 **Mobile 앱에서 DLNA/WiFi/BT 기술 이용하여 IoT 장비와의 연동 처리에 대해서도 관심**이 있다
D. 기존 **STB 시장에서** 8년간 일해오면서 **능력, 인내, 열정 만으로는 미래를 보장할 수 없다는 생각을 하게 되었다. 사양 산업**으로 변하게 되면서 **미래에 대한 불안**이 있었기 때문에 퇴사를 결심하였다.
E. (참고-게임업체의 경우) **최근 10년간 꾸준히 성장하고 있는 산업이 (모바일) 게임시장**
    i. 글로벌 게임시장을 보면 **2020년 기준 전년도 대비 9.3% 성장 추세**

ii. 전체 게임시장에서 **모바일 게임의 계속적인 성장 (전체의 48%비중)**
iii. **모바일 분야를 특화** 시키고자 하는 나의 의지와 이 모바일 게임 시장에서 그동안의 안드로이드 **개발 경험을 가지고 안드로이드 개발 부문에 공헌**할 수 있을 것 같아서 지원하였다.
iv. PC게임부터 시작해서 모바일까지 **전통적인 게임**의 강자 업체가 있고, **레트로 게임**을 기반으로 성장한 업체도 있다.
v. **유저들의 과금(현질) 유도보다는** 앞으로의 **게임 슈퍼 싸이클을 잘 타려면** 현재 출시된 게임 외에 **신작 게임의 출시가 필요**하다고 생각
vi. XXX 게임의 경우처럼 **내수시장에 이어 활발한 외수시장 진출 사업**도 하면 회사 발전에 좋을 것 같다.
vii. 게임 개발 외에 사용자와 개발자 모두를 위한 공통적인 부분에 대한 **플랫폼 SDK** 개발 업무도 존재
    1. 대부분의 게임에는 **결제, 로그인, 회원가입** 등 필수적으로 들어가는 기능 ← 이 기능들을 게임 앱에 적용하기 위해 SDK 사용
    2. 게임 개발자가 자체적으로 인증 / 애플, 구글의 결제 기능을 구현한다면 시스템 개발에 많은 시간 소요 ← **SDK 제공/연동**을 통해 이 부분 해소
    3. 안드로이드는 리소스 + Manifest 파일 포함 시킨 **AAR** 형태로 제공 / iOS는 **Framework** 파일로 제공

13. SDK란 무엇인가?

A. Software Development Kit 약자, 소프트웨어를 개발할 때 사용하는 도구 모음
B. API, GUI, 문서, 라이브러리, 코드샘플을 모은 패키지

14. 지원 동기는 무엇인가?

A. 최신 트렌드가 Mobile로 집중되는 상황에서 해당 기업의(대기업의) 모바일 개발 업무를 원해서 지원

15. 우리 회사 앱을 사용해 본 적이 있는가?

A. 설치해서 사용해 본적이 있다. UI/기능을 적당히 사용해보았다.

16. 사용해 본 앱에서 어떤 부분이 부족하다고 생각하는가?

A. 조금 더 직관적이고 안드로이드 Default UI가 아닌 디자인된 UI를 적용했으면 하는 아쉬움, 타 사 유사 앱과 비교
B. UI Component 배치를 유명 앱들 로그인/회원가입 화면처럼 구성했으면 한다.
C. 설정 관련해서 수동 설정, UI 통한 설정이 자유로웠으면 좋겠다
D. 기능에 대한 가이드가 부족한 것 같다. (도움말)
E. 설정하고 화면을 빠져나갈 때 최종적으로 확인하는 팝업이나 표시하는 화면이 있었으면 좋겠다. (최종 설정 확인을 위해)
F. 설정을 다 미친 후에는 최종 설정에 대한 내용을 SMS/메일/카톡으로 보내주는 기능

추가가 필요

G. 필요한 부품(IoT 장비 등)을 구매할 수 있는 Feature도 추가가 되면 좋을 것 같다

17. 이전 회사 개발팀 구성이 어떻게 되는가?

A. Android 파트 내 앱 / 프레임워크 / 미디어 플레이어 / 플랫폼 파트 / 하드웨어 팀으로 구성

18. 형상 관리 시스템 구축은 무엇을 해보았는가?

A. Jenkins 시스템 환경 구축

B. 빌드 이미지 관리 / Release를 위한 Bash / Python 스크립트 작성 경험

C. CI, CD Job 설정 / 관리

19. 우리 회사에 계속 다닐 생각인가?

A. 네 그렇습니다. 8년 간 한 회사에서 근면/성실하게 일해왔다.

20. 개발 직이면 네이버나 카카오 그룹이 더 적성에 맞지 않는가?

A. (해당 기업이 B2C 기업이라면) 조금 더 소비재에 가까운 앱을 개발할 수 있고 조금 더 사용자 입장에서 업무를 할 수 있어 재미와 보람이 있을 것 같아서 지원

B. 같은 B2B 기업이라면, 회사의 규모/Job Description/기사를 언급 / 모바일 분야 언급

21. 개발도 개발이지만 우리는 대외적인 커뮤니케이션, 매니징 역할도 원한다. 가능한가?

A. 가능하다. 현재는 개발 업무만을 하고 있지만 몇 년 후에는 개발 조직을 이끄는 기술 매니저가 되기를 희망한다.

B. 현재도 개발 업무만을 하는 것이 아니라 안드로이드 기술적으로 가이드를 위해 외부 업체와 커뮤니케이션을 진행 중이다.

C. 현재 맡은 모듈에 대해서는 개개인이 외부 업체와 능동적으로 커뮤니케이션 / 개발을 하고 있다.

22. 우리가 급할 때는 개발도 하고 외주 인력 관리도 하고 여러가지 일들을 해야 한다. 가능한가?

A. 가능하다. 지금 현재도 개발 업무만 하는 것이 아니라 협업하는 개발자와의 커뮤니케이션, 앱 포팅 및 빌드 환경 셋업, 타 업체에 개발 가이드 제공 업무 등을 동시에 같이 하고 있다. (바로 위 질문과 유사)

23. 현재 회사에서 B2C 비슷한 일을 해본 적이 있는가?

A. 거의 다 B2B 사업이었다.

B. 소비자에게 배치된 상품에 대한 피드백을 개발자들이 사내 품질 팀을 거쳐 이슈 관리 시스템을 통해 전달 받게 되고 우리는 그 이슈에 대한 재현 경로를 숙지하고 재현 후 수정을 하고 있다.

24. 하이브리드 앱 특성에 대해 설명해보아라.

A. 웹 표준 기술을 그대로 사용, 웹 앱(Html, Css, Js)을 개발한 후에 오픈 소스 크로스 프레임 워크를 이용하여 네이티브 앱으로 변환시켜 배포되는 앱 형식
B. 네이티브 앱 특성을 가지고 있음 (카메라, GPS 등의 센서에 접근 가능)
C. 개발 프레임워크
   i. PhoneGap : Cross Platform Mobile Application Framework
   ii. Titanium : Web App을 Native App으로 변환 (변환을 위해서는 Titanium에서 제공한 API만을 사용해야 함)
   iii. Appspresso : KTH에서 개발한 최신 프레임 워크
D. UI 프레임워크
   i. jQuery : 자바 스크립트 생산성 향상을 위한 라이브러리
   ii. Sencha Touch : 모바일 웹 앱 개발을 위한 자바 스크립트 프레임워크

25. 이력서 내용 중 Open API Service라는 것이 무엇이냐?

A. 팀 내 또는 **3rd Party 업체에 우리 API를 제공**하기 위한 서비스
B. Platform 의존성이 있는 설정 관련 API / DB 제어 API / 사내 서버 API / 프로퍼티 접근 등 제공
C. API 라이브러리와 API 사용 가이드를 배포
   i. **Java Docs / PDF / Markdown Document** 이용해 가이드 문서 작성/배포

26. 로그인 – 회원가입 관련 처리 경험이 있는가?

A. 앱에서 **사내 서버로의 로그인** 동작 처리 해보았다
B. OTT박스에서 로그인 후 사내 웹앱을 다운로드 받는 시스템이었다

27. AWS 서버 연동 경험이 있는가?

A. 없음. 사내 Cloud 서버 연동 경험은 있다 (로그인 / 다운로드 처리)

28. 본인의 부족한 부분을 어떻게 채울 것이냐?

A. 사내 교육 또는 업무 외 활동을 통해 채울 것이다. (동영상/현장 강의, 서적 등을 통한 스터디 활동)

29. 개발 툴을 무엇을 사용하는가?

A. 앱 개발 때는 Android Studio를 사용, Android Framework 소스 검토 시에는 Visual Code를 주로 사용한다. 버전 관리는 Git으로 진행, 코드 리뷰는 Gerrit Review System을 사용한다.

30. 앱 에이징(스트레스 테스트)는 주로 어떠한 식으로 하는가?

A. 로그 추출 프로그램 실행 이 후 특정 동작 반복을 위한 Shell Script 작성한 뒤 실행하여 테스트 진행

31. 안드로이드 버전 업그레이드에 대한 대처는 어떠한 식으로 하는지?

A. Android Developer 사이트 / Google I/O 등의 매체에서 변경사항 학습
B. **Target SDK 버전** 업데이트해서 빌드 테스트
C. **최신 OS의 구글 레퍼런스 (넥서스) 폰 또는 에뮬레이터**에 앱 설치/테스트
D. 앱 내 Android Framework 라이브러리를 직접적으로 가지지 않고 Java Reflector를 사용하여 Method를 사용하도록 처리
E. Android Version에 따라 Gradle 빌드 환경을 설정하여 앱 빌드

32. 카메라 연동 경험이 있는지 ?

A. 과거 해커톤이라는 프로젝트 참가하여 카메라 이용한 실시간 번역 앱을 만들어본 경험이 있다. DB, 단어장, 번역기능을 구현했었다
B. 1박 2일 동안 개발 진행 후 구현에 대한 발표를 진행했다

33. 안드로이드 모바일 개발과 STB/OTT 개발 간 차이는 무엇인가?

A. UX/UI 적인 부분은 다를 수 있어도 개발하는 것은 차이 없다.
B. 기본적인 안드로이드 아키텍쳐는 동일하다
C. 오히려 STB/OTT 개발자들이 더 강점이 있다. AOSP Source Code를 볼 수 밖에 없는 환경이라 앱만 개발하는 사람에 비해 안드로이드 시스템 전반적인 것에 대해 이해도가 높다

34. Google Android TV Spec 정합 작업이란 무엇인가?

A. 고객사 요구사항에 따라 Google 기본 설치 마법사의 Step 수정 작업 진행
B. Google 설치 마법사 맨 앞/뒤에 Vendor가 Customizing 할 수 있는 페이지 작업 (Bluetooth RCU Pairing, Launcher 진입 전 최종 설정 작업)

35. 라이브러리 배포 시 문서화와 배포는 어떠한 식으로 하였는가?

A. 메일/Jira Task 상으로 전달
   i. 라이브러리 제작 후 사내 Cloud에 업로드하여 링크 제공
   ii. 문서는 Markdown 형식으로 편집하여 전달 (또는 PDF / Java Docs 사용)

36. 안드로이드 버전 현재 어디까지 진행되었는가?

A. 현재는 Pie 버전에서 개발 진행 중이다.
B. 참고
   i. Android 12 (API Level 31) 개발 중
   ii. Android 11 (API Level 30) 지원 중
   iii. Android 10 (API Level 29)
   iv. Android 9 (API Level 28)

37. 입사하면 대충 어떤 일을 할 지에 대해 감이 오는가?

A. 인터넷 기사 / Job Description / 회사 사이트를 보고 내용 숙지해서 말한다

38. DLNA / WiFi Direct / Bluetooth 에 대해 설명해보아라

A. 기술

i. **DLNA**

1. Digital Living Network Alliance
2. SmartPhone, TV, 컴퓨터, NAS 등 기기 간의 미디어 컨텐츠 공유를 가능케 하는 기술 (제조사, 제품 종류에 관계없이 공유 가능)
3. 같은 IP 네트워크 망에서 상호 연동 가능

ii. **와이파이 다이렉트**

1. 무선 AP가 없어도 장치들을 쉽게 연결할 수 있는 와이파이 표준
2. 와이파이 다이렉트 장비들은 첫 연결이 이루어질 때 AP로 동작할 장비를 서로 간에 결정한다

iii. **블루투스**

1. 디지털 통신 기기를 위한 개인 근거리 무선 통신 산업 표준

39. 현재 팀장님이 나에게 조언 / 피드백을 준다면 어떠한 것들이 있을까?

A. **적극적인** 업무 활동 **유지**

B. **스트레스 관리** ➔ 범우주적인 관점에서 볼 때 모든 일들은 사소한 것이다라고 생각을 하려고 노력한다. 그 밖에 가벼운 음주, 운동, 취미생활로 해소하려고 노력한다.

C. 업무 관련 잦은 **피드백** (관리자 입장에서는 물어보지 않아도 현재 진행 상황을 알 수 있으면 좋기 때문에 Jira나 여러 툴을 사용해서 현 상황을 공유한다)

40. 혼자서 앱을 구현한 것인지?

A. UI앱의 경우 UX/UI 팀과 협업하면서 혼자서 구현

B. Android TV 데이터 처리 Service 구현 시에는 여러 명 이서 협업

41. 페어 코딩한 프로젝트가 있는지?

A. 내부 처리 모듈이 많은 앱의 경우 모듈 별로 나눠서 각자 모듈을 구현

42. FFmpeg Library 사용해서 음성, 비디오 등 여러가지 처리 해보았는지?

A. Internet Radio라는 앱 구현 시 Audio 재생을 위해 사용해 본 경험이 있다.

43. 업무 외 따로 개인적으로 공부를 한 것은 있는가?

A. 최근 React Native, Flutter, Kotlin 활용한 앱을 만들어 보고 있다.

44. 많은 유저가 사용하는 앱을 담당한 적이 있는가?

A. 많은 유저가 동시적으로 접속하는 앱은 만들어 본 적 없으나 유저에 의해 조작되는 앱은 많이 담당해보았다. (RCU 서비스, Launcher, Wizard)

45. UI/UX 없이 어떤 기능을 만들어라고 요구를 받으면 가능한지?

A. 가능하다. 다만 Designer가 없이 개발하는 것이라 UX/UI 관점에서 세련되지 않을 수 있다. 만약 UI 샘플이 있다면 최대한 비슷하게 제작 가능하다

46. JNI 관련 작업 해본 적 있는가?

A. 스크린 사이즈 조정 / HDMI 해상도 설정을 하는 Amlogic / Broadcom 의 특정 기능을 Java Layer 단에서 컨트롤 하기 위해서 JNI 작성 해본 적 있다.

47. Rest API 사용에 대한 개념이 있는가?

A. 개념

i. **REST : Representational State Transfer**

1. **리소스를 이름으로 구분하여 해당 자원의 정보를 주고 받는 모든 것**
2. HTTP 프로토콜을 그대로 활용하므로 웹의 장점을 최대한 활용 가능
3. Network 상에서 Client와 Server 사이의 통신 방식 중 하나
4. HTTP URI(Uniform Resource Identifier)를 통해 리소스를 명시하고 HTTP Method(POST, GET, PUT, DELETE)를 통해 해당 자원에 대한 CRUD Operation을 적용하는 것
5. 자원 기반의 구조(ROA, Resource Oriented Architecture) 설계의 중심에 리소스가 있고 HTTP Method를 통해 리소스를 처리하도록 설계된 아키텍쳐
   A. CRUD Operation (Create-Post, Read-Get, Update-Put, Delete, HEAD)
6. 의도하는 바 쉽게 파악 가능, HTTP 표준 프로토콜 따르는 모든 플랫폼에서 사용 가능
7. HTTP Method 형태가 제한적
8. 브라우저 버전에 따라 Method 사용 불가한 점
9. 필요한 이유
   A. 다양한 클라이언트 사용 가능
10. 구성 요소
    A. URI(리소스), HTTP Method(행위), JSON/XML/TEXT/RSS (표현)

ii. **REST API**

1. **REST 기반으로 서비스 API를 구현한 것**
2. 최근 구글맵, 공공 데이터 등과 같은 Open API 등을 제공하는 업체 대부분은 REST API를 제공

iii. **RESTful**

1. **REST API를 제공하는 웹 서비스를 RESTful하다고 할 수 있음**
2. REST 원리를 따르는 시스템

48. 네트워크 비동기 처리 경험이라고 작성되어 있는데 무엇인가?

A. OkHttp Library 사용한 **클라이언트 단** 구현 경험을 의미한 것이다

B. 고객사 서버로부터 추천 리스트 / 로그인 정보 / 등록된 단말기 정보를 요청하고 결과를 받기 위한 코드 구현을 하였다.

49. 구현 업무 시 디자인도 같이 하시는 것인지?

A. 디자인은 UX/UI 팀에서 진행하고 시나리오 문서와 GUI 이미지를 전달받으면 구현을 시작한다.

50. Jenkins 관리만 한 것인지, 구축까지 한 것인지?

A. 구축과 관리 둘 다 하였다. (CI/CD Job 설정하여 활용)

51. Jenkins에 테스트 자동화까지 연동해 보았나?

A. 빌드 테스트 자동화를 위해 **CI** Job 구성해 본 경험이 있다. 소스 코드 저장소에 Push가 발생했을 때 자동으로 빌드를 시키고 빌드 결과를 팀 전체 메일로 공유되도록 설정

52. UX/UI팀에서 무리한 요구를 할 때 주로 어떻게 상황을 대처하는가?

A. 개발 진행하면서 그러한 상황은 딱히 없었다.

B. 보통 UX/UI팀에서 시나리오 회의를 주최하면 개발자도 함께 참석

C. 참석한 회의에서 서로 의견을 조율, 서로의 영역에 대한 기초 지식 공유

53. UX/UI팀과 시나리오 협의는 자주 일어나는가?

A. 디자인팀에서 시나리오 회의 주최 시 개발자도 함께 참석하여 사전에 의견 조율하므로 빈번하게 일어나지는 않는다.

54. 최근 관심 가지고 있는 트렌드나 SW 관련 기술이 있는가?

A. React Native, Flutter, Kotlin, 안드로이드 아키텍쳐 패턴, RxJava 등

55. 출장은 보통 어떠한 경우에 가는 것인가?

A. 내수보다 외수가 대부분인 회사이므로 거의 대부분 출장 업무가 필요

B. **현지 개발 / 테스트**를 위함

C. **현지 개발자들과의 협업**을 위함

D. **일정 단축**을 위함

56. 방송 스트림에 대한 레코딩 업무라는 것에 대해 설명해달라.

A. Cable / IP로 들어오는 방송 데이터를 ts파일로 저장하는 작업을 의미

B. Launcher 앱에서 레코딩 입력이 들어왔을 때 이벤트를 내부 서비스로 전달하고 레코딩 작업이 개시되면 현재 레코딩 진행 정보를 Callback으로 전달 받아 UI를 업데이

트 하였다.

C. HDD, DLNA 저장소에 대해 녹화가 되도록 개발하였다. DLNA 의 경우 DLNA 라이브러리르 포팅하여 개발 진행하였다.

57. 이력서 상에 대 제목으로 경력을 기술해놓은 것은 본인이 생각하기에 큰 사건이나 구현을 기준으로 작성한 것인지?

A. 그렇다. 기간이 길고 **대규모 프로젝트를 중심으로** 기술하였다.

58. 자신이 가지고 있는 안드로이드에서의 장점은 무엇인가?

A. 특정 앱의 일부분 모듈 만을 다년간 개발한 것이 아닌 어떤 하나의 앱을 초기 개발 단계부터 최종 릴리즈 까지 개발이 가능하다는 것이 장점

59. REST API Server 구축이 가능한 것인지?

A. 가능하다. 과거 프로젝트에서 API Server 구현 소스를 검토해 본 경험이 있다

60. 어떤 컨텐츠에 대한 데이터 처리를 Launcher 앱에서 처리하였는지?

A. 추천 컨텐츠 리스트 구현을 한 예로 들 수 있다.

B. Launcher가 아닌 다른 서비스에서 고객사 Cloud 서버로 컨텐츠 정보를 요청하고 결과 값을 받으면 DB에 저장

C. Launcher에서 Home 버튼을 누르면 해당 메뉴 진입 시 DB를 읽어들여 컨텐츠 UI들을 표시

61. 로그 데이터 관리는 어떻게 하였는지?

A. 개발하는 앱 내에서 로그 관리하는 클래스를 따로 만들어서 사용

B. 안드로이드 Tombstone, Vendor log

62. 웹 개발로 전향 시 현업 투입 시까지 몇 주 몇 달 소요될 것 같은가?

A. 숙달되는 데에까지 3 ~ 6개월 예상

63. Jenkins 구축 시 시간이 많이 소요되는 부분은 어디인가?

A. 빌드를 위한 Manifest 구성 작업에서 시간이 많이 소요

64. Jenkins 빌드 / Publish 등에 사용되는 Script는 어떻게 만들었는가?

A. Shell / Python Script로 만들었다.

65. 회사는 어느 경로로 알게 되었는가?

A. 중소의 경우

i. 구직 사이트에서 먼저 보고 Job Description 과 회사 사업을 보고 적합하다 생각하여 지원했다

B. 대기업의 경우
   i. 기사나 회사 사업 내용을 보고 현재까지 쌓아온 스킬과 회사에서 요구하는 내용이 매치되고 모바일 분야 / 모바일을 통한 BT, WiFi, DLNA 통신에 관심이 있었는데 회사 사업도 그러한 분야라 서로의 Needs에 부합된다 생각하여 지원했다

66. HTTP Request 코드에 대해 아는 대로 설명해보아라

A. HTTP Response 상태 코드
   i. **2xx** : 성공
   ii. **3xx** : 리다이렉션
   iii. **4xx** : 클라이언트 오류
   iv. **5xx** : 서버 오류

67. 비동기 처리할 때 효과적인 방법은 무엇인가?

A. 이벤트 핸들링을 한 곳에서만 처리

B. 코루틴, RxJava 활용

68. 디자인패턴 / 아키텍쳐 패턴 사용?

A. 디자인 패턴
   i. 싱글턴 패턴
   ii. 팩토리 메소드
   iii. 추상 팩토리 패턴
   iv. 빌더 패턴
   v. 옵저버 패턴
   vi. 반복자 패턴
   vii. 데코레이터 패턴

B. 아키텍쳐 패턴
   i. **MVC** (전통적인 방법, 안드로이드에서는 C 가 Activity)
   ii. **MVP** (View – Presenter 1:1 매칭)
   iii. **MVVM** (View – ViewModel n:m 매칭)

69. 오픈소스 라이브러리 사용 시 선정 기준?

A. GitHub 나 Google 통해 Review 후 자주 쓰이는 라이브러리를 찾아서 사용했다

70. 테스트 자동화는 어떻게 하였는가?

A. Appium(앱피움) 등으로 테스트 자동화 가능한 것으로 알고 있다.

71. 로그데이터는 어떻게 처리하는가?

A. Splunk 활용, 기타 잡지식 이야기 한다.

72. 앱 Crash 발생 건은 어떻게 처리하는가?

A. Tombstone 로그나 품질팀으로부터 보고된 재현 경로를 통해 개발자가 직접 재현하고 이슈 수정 후 Jira / 기타 이슈 트래킹 페이지 이용하여 개발팀 매니저에게 Resolved 상태로 할당한다.

73. 로그 관리는 어떻게 하는가?

A. 사내에서 시스템적으로 Splunk 를 사용하고 있고 안드로이드 개발 앱 내에서 Splunk API 를 호출하도록 구현하고 있다.

74. Main Thread 와 Worker Thread 등을 구분하는 이유?

A. **ANR 발생을 막기 위해** 구분 (Application Not Responding)

B. Main(UI)스레드에서 5 초 이상 반응이 없으면 ANR 이 발생하므로 오래 걸리는 작업은 Worker Thread 에서, UI 업데이트는 Main 에서 작업

75. Main Thread 에서 반드시 동작해야 하는 함수가 있는지에 대해 설명해봐라

A. UI 업데이트 관련 함수는 Main Thread 에서 Call 되어져야 한다

76. 람다 함수와 고차함수(High Order 함수) 에 대한 설명을 해보아라

A. 모르겠다

77. 람다 사용에 익숙한가요?

A. 안드로이드 자바 개발 중 Anonymous 함수 구현 시에는 사용한다. (문법적인 요소)

78. 데이터 베이스는 사용해 보았는지?

A. 안드로이드 앱 개발 시 대부분 Database 를 사용하고 주로 SQLite 를 사용한다. mysql, sqlite 사용해보았다

79. DB Join 도 해 보았나요?

A. 두개 이상의 테이블을 연결 또는 결합하여 데이터를 출력하는 것을 의미

B. 테이블 연결하여 쿼리를 할 때 주로 사용했다

80. mysql, mssql 같은 데이터베이스 연동해 본 경험이 있는가?

A. 대학 전공 과목 중 데이터베이스 과목이 있었으며 그 때 MySQL, MSSQL 사용해보았다.

81. UI 테스트는 어떻게 하며 자동화는 어떻게 하고 있는가?

A. Appium 으로 테스트 자동화 가능

B. 실제 현업에서는 특정 UI 동작에 대한 Shell Script 작성 후 스크립트를 실행하여 테스트를 진행했다. 주로 에이징 테스트인 경우에 스크립트를 써서 테스트하였다

82. 현재 팀 내에서 진행 중인 업무 나열해보아라

A. Launcher 앱 유지보수

B. API Service 앱 유지보수

C. 이슈 연관 시 기타 앱 소스 코드 분석 / 수정 작업

83. 디자인 패턴인 싱글턴 (Singleton) 사용 이유 / 개념 / 용법

A. 사용이유

i. **고정된 메모리 영역을 얻으면서 한번의 new 로 인스턴스를 사용하기 때문에 메모리 낭비를 방지할 수 있음**

ii. 싱글톤으로 만들어진 클래스의 인스턴스는 **전역 인스턴스**이기 때문에 다른 클래스의 인스턴스들이 **데이터를 공유하기 쉽다.**

iii. **공통된 객체를 여러개 생성해서 사용해야하는 상황에서 많이 사용.**

iv. 안드로이드 앱 같은 경우 각 액티비티나 클래스 별로 주요 클래스들을 일일이 전달하기가 번거롭기 때문에 싱글톤 클래스를 만들어 어디서나 접근하도록 설계하는 것이 편하다

B. 개념

i. 애플리케이션이 시작될 때 어떤 클래스가 **최초 한번만** 메모리를 할당하고(Static) 그 메모리에 인스턴스를 만들어 사용하는 디자인패턴

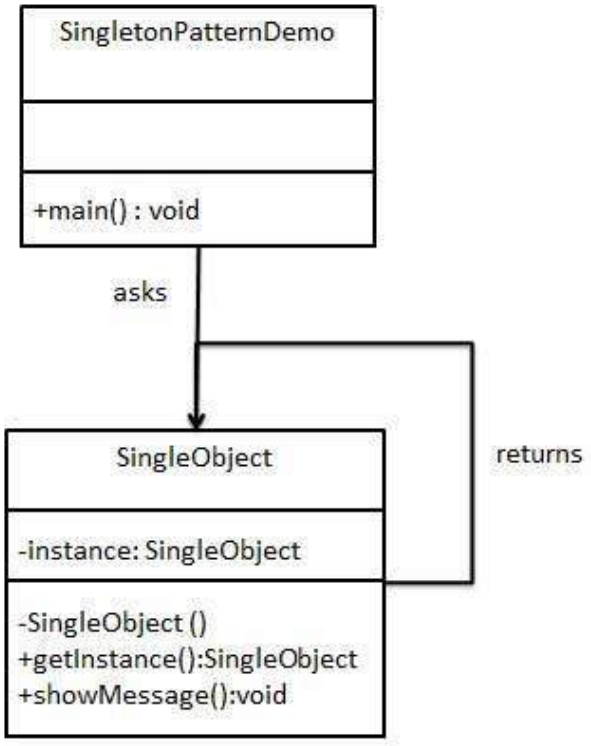
ii. 단 하나의 인스턴스를 생성해 사용하는 디자인 패턴

C. 용법

i. **Class 안에 Class(Holder)**를 둔다

ii. Holder 안에 선언된 instance 가 static 이기 때문에 **클래스 로딩 시점에 한번만 호출될 것이며** final 을 사용해 다시 값이 할당되지 않도록 만든다

D. UML



E. Code

```
public class ExampleClass {

    //private construct
    private ExampleClass() {}

    private static class InnerInstanceClazz() {
        private static final ExampleClass instance = new ExampleClass();
    }

    public static ExampleClass getInstance() {
        return InnerInstanceClazz.instance;
    }
}
```

84. Asynctask 에 대한 설명 / 사용 이유

A. 설명

i. 백그라운드에서 짧은 작업을 비동기적으로 실행할 수 있도록 Object 클래스를 확장하는 클래스

B. 사용 이유

i. **개발자가 스레드 및 핸들러를 조작하지 않고도 백그라운드 작업을 수행**하고 **UI 스레드에 결과 게시 가능**하기 때문에 사용

85. Final 키워드 위치에 대한 설명 (Class 앞 / 멤버변수 앞)

A. 변수 앞

i. **상수**

ii. 변수를 선언과 동시에 초기화하며 이후에 값 수정 불가

B. 함수 앞

i. **오버라이딩 불가능**

ii. 상속 받은 그대로 사용해야 함

C. 클래스 앞

i. **상속이 불가능**

ii. 서브 클래스 생성 불가능

86. Process <-> Thread 차이점

A. 프로세스

i. 컴퓨터에서 연속적으로 실행되고 있는 컴퓨터 프로그램

ii. 메모리에 올라와 실행되고 있는 프로그램의 인스턴스

iii. OS 로부터 시스템 자원을 할당받는 작업의 단위

1. CPU 시간, 주소공간, Code, Data, Stack, Heap 구조의 독립된 메모리 영역

iv. 실행된 프로그램

v. 기본적으로 최소 1 개의 스레드(메인 스레드)를 가지고 있다

vi. 각 프로세스는 다른 프로세스의 변수나 자료구조에 접근 불가능

vii. 접근을 하려면 IPC 를 사용

B. 스레드

i. 프로세스 내에서 실행되는 여러 흐름의 단위

ii. 프로세스의 특정한 수행 경로

iii. 프로세스가 할당받은 자원을 이용하는 실행의 단위

iv. 스레드는 프로세스 내에서 각각 Stack 만 따로 할당받고 Code, Data, Heap 영역은 공유

v. 프로세스 내 주소 공간 / 자원들을 서로 공유하면서 실행

vi. 한 스레드가 프로세스 자원을 변경하면 다른 스레드도 그 변경 결과를 즉시 볼 수 있다.

87. RxJava 사용해본 적 있는가?

A. 개념은 숙지하고 있다

B. 개념

i. 관찰 가능한 스트림을 사용하는 비동기 프로그래밍을 위한 API 를 Java 로 구현한 라이브러리

ii. 동시성 문제, 다중 이벤트 처리, 백그라운드 처리 들의 문제 처리

iii. Observable(발행), Subscriber (소비), subscribe, map, just

88. Mutable / Immutable Class 설명

A. Mutable Class

i. 인스턴스가 생성된 후에 값의 내용이 변할 수 있는 클래스, 주소는 못 바꿈

ii. StringBuilder (내부적으로 return this, 주소값 변경되지 않음)

B. Immutable Class

i. 클래스의 인스턴스가 생성된 후에는 인스턴스의 내용이 절대 변하지 않는 특징을 갖는 클래스

ii. Set 메소드가 없다.

iii. String, Boolean, Integer, Float, Long

89. Bitmap Class 사용함에 있어서의 유의사항 / 용법

A. BitmapFactory 클래스 사용하여 Bitmap 객체를 얻는다

B. 유의사항

i. 카메라로 촬영한 이미지 사이즈가 아주 크면?

1. 사이즈가 큰 이미지를 여러 번 decoding 하다 보면 OutOfMemoryError 발생

2. inSampleSize 옵션 (Decode 할 때 얼마만큼 줄여서 decoding 을 할 지 정하는 옵션, 1 ➔ 1, N ➔ 1/N, 2 의 지수만큼 비례할 때 가장 빠르다)

3. inSampleSize 사용하면 SkScaledBitmapSampler Object (Library Level) 를 생성 ➔ Object 를 만들 때 정해진 SampleSize 만큼 축소하여 Width 와 Height 를 정한 뒤에 생성 (애초에 축소된 사이즈로 이미지 Decoding)

90. Bitmap & JPEG 차이점 설명

A. Bitmap ➔ 압축 X, 화질 가장 좋음, 용량 큼

B. JPEG ➔ 손실 압축 방식, Bitmap 에 비해 화질 떨어짐, 용량 작음 (인터넷에 사진 업로드)

C. GIF ➔ 비 손실 압축 방식, 화질 좋으나 최대 256 색까지 지원, 특허 (인터넷에 사진 업로드)

D. PNG ➔ 비 손실 압축 방식, 트루 컬러 지원 (인터넷에 사진 업로드, 간단한 로고제작)

91. 리스트뷰 / 스크롤뷰 / 리싸이클러뷰 차이점

A. 리스트뷰
   i. ViewHolder 패턴이 강제가 아님
   ii. 세로 방향만 지원
   iii. 아이템 추가/제거 때 적용할 수 있는 애니메이션 없음
   iv. ArrayAdapter / CursorAdapter 등 다양한 소스에 대한 어댑터 존재
   v. Divider 속성 이용하여 구분선 설정 가능
   vi. OnItemClickListener 인터페이스 존재

B. 리싸이클러뷰
   i. ViewHolder 패턴 강제 구현
   ii. 가로/세로/지그재그 방향 지원
   iii. 아이템 애니메이션 처리 클래스 존재
   iv. 사용자 정의 Adapter 를 필수 구현
   v. ItemDecoration 을 사용하여 구분선 설정
   vi. Click 이벤트 처리는 직접 구현해야 한다

C. 스크롤뷰
   i. 단 하나의 Direct 자식만을 가질 수 있음
   ii. ListView, RecyclerView 를 가지면 안된다. 가지고 있다면 ScrollView 가 아닌 NestedScrollView 를 사용해야 한다. (이중 스크롤 이슈 방지를 위해)

92. 액티비티 생명주기 설명 (Activity Transparent 속성 포함해서 설명)

A. onCreate ➔ onStart ➔ onResume ➔ onPause ➔ onStop ➔ onDestroy

B. 다른 액티비티에 의해 가려지는 경우
   i. 일반적인 상황 ➔ onStop ➔ onDestroy

ii. Transparent 액티비티에 의해 가려진 상황 ➔ onPause

93. Thread / Service 구현 간의 차이 (Service 를 구현하는 이유)

A. Thread
   i. 앱이 사용자와 상호작용하는 과정에서 UI Thread 가 Block 되지 않기 위한 작업 등을 처리하기 위한 Foreground 작업에 적합
   ii. 앱이 구동 중에 시스템에 의해서 프로세스가 강제 종료되는 경우
      1. Android 시스템이 다시 복구시켜 주진 않음

B. Service
   i. 사용자와 상호작용하지 않아도 계속 수행되어야 하는 Background 작업에 적합
   ii. Background 에서 실행 시간이 긴 작업을 수행
   iii. 앱이 구동 중에 시스템에 의해서 프로세스가 강제 종료되는 경우
      1. onStartCommend 반환 값에 따라서 강제 종료된 Service 를 시스템이 다시 자동으로 시작하게 만듬

94. 비동기, 동기, Blocking, Non-Blocking 에 대해 설명해보아라.

A. 동기 / 비동기
   i. 호출되는 함수의 **작업 완료 여부를 누가 신경쓰느냐**가 중점
      1. 동기
         A. 작업 요청 시 요청의 결과값을 직접 받는 것
         B. 결과값이 함수의 return 값과 동일
         C. 호출한 함수가 스스로 작업 완료를 신경 쓴다
         D. 동기 + Blocking ➔ Read/Write
         E. 동기 + Non-Blocking ➔ Read/Write(Polling)
      2. 비동기
         A. 작업 요청 시 결과값을 간접적으로 받는 것
         B. 결과값이 return 값과 다를 수 있다
         C. 호출된 함수(Callback 함수)가 작업 완료를 신경 쓴다
         D. 비동기 + NonBlocking ➔ 비동기 I/O
         E. 비동기 + Blocking ➔ I/O Multiplexing (Select / Poll) – Node.js + MySQL

B. Blocking / Non-Blocking
   i. 호출되는 함수가 **바로 리턴하느냐 마느냐**가 중점
      1. Blocking

A. 요청한 작업을 마칠 때까지 계속 대기
B. 제어권을 넘겨주지 않고 대기

2. Non-Blocking
A. 요청한 작업을 즉시 마칠 수 없다면 즉시 return 한다
B. 호출한 함수가 다른 일을 할 수 있는 기회를 준다

95. onSaveInstanceState function 에 대한 설명

A. 이전 액티비티의 상태 저장 / 복원하는 콜백 메소드
B. Running 상태의 Activity 가 다른 Activity 를 호출한 경우, 즉 Activity 가 완전히 가렸다가 다시 Foreground 상태가 되는 경우에 호출됨
   i. onSaveInstanceState ➔ onPause ➔ onStop ➔ onRestart ➔ onStart ➔ onRestoreInstanceState ➔ onResume
C. 화면 회전 시
   i. onPause ➔ onSaveInstanceState ➔ onStop ➔ onDestroy ➔ onCreate ➔ onStart ➔ onResume
D. Pause 상태의 Activity 가 System 자원(메모리) 부족으로 System 에 의해 강제 종료되었다가 System 자원 여유가 생겨 다시 복구될 경우
   i. 강제종료 ➔ onCreate ➔ onStart ➔ onRestoreInstanceState ➔ onResume

96. Fragment 라이프 사이클은 어떻게 되는가?

A. 시작 시
   i. onAttach ➔ onCreate ➔ onCreateView ➔ onActivityCreated ➔ onStart ➔ onResume
B. 전환 시
   i. onPause ➔ onSaveInstanceState ➔ onStop ➔ onDestroyView ➔ onDestroy ➔ onDetach ➔ onAttach ➔ onCreate ➔ onCreateView ➔ onActivityCreated ➔ onState ➔ onResume

97. 패딩, 마진에 대한 설명

A. 패딩
   i. 위젯 테두리로부터 위젯 안에 내용 사이의 여백
B. 마진
   i. 위젯의 부모 레이아웃의 테두리로부터 여백

98. 핸들러-루퍼-스레드 상관관계

A. 핸들러
   i. 스레드의 Message Queue 에서 던져주는 Message 또는 Runnable 을 처리하는 역할

B. 루퍼
   i. 자신을 생성한 스레드의 Message Queue 에 들어오는 Message, Runnable 을 순서대로 Handler 에 전달하는 역할
   ii. Main 스레드는 Looper 가 기본적으로 생성되어 있음
   iii. 새로 생성한 Sub 스레드는 Looper 를 생성하지 않으므로 메시지나 Runnable 을 받으려면 Looper 를 생성해야 함 (prepare()로 생성 / loop()로 메시지큐 관리)

C. 스레드
   i. 스레드 안에 루퍼, 핸들러 존재

99. 왜 Main Thread 에서만 UI 작업이 가능하도록 설계되었나?

A. 병렬 처리로 돌아가고 있는 Main 과 Sub Thread 에서 같은 TextView 를 setText 했을 때 어떤 Thread 의 명령을 따라야 할지 결정할 수 없기 때문에 **(동기화 문제 처리을 위함)**

100. 자료구조 Set, Map, List, Stack, Queue 에 대해 설명하여라

A. Set
   i. 중복되지 않는 데이터를 구할 때 사용하면 유용
   ii. 순서 X, 중복 허용 X
   iii. 빠른 속도
   iv. 단순 집합 개념, 정렬하려면 별도의 처리 필요

B. Map
   i. 키와 값으로 나눠서 데이터 관리
   ii. 순서 X, 키 중복 X, 값 중복 O
   iii. 빠른 속도
   iv. 키의 검색 속도가 검색 속도를 좌우

C. List
   i. 저장공간이 필요에 의해 자동으로 늘어난다

ii. 순서 O, 중복 O
iii. 가변적인 배열
iv. 원하는 데이터가 뒤쪽에 위치하는 경우 속도의 문제 발생
v. ArrayList (객체 내부에 있는 배열에 데이터 저장), Vector(배열, 동기화 처리), LinkedList (양방향 포인터 구조, 삽입, 삭제가 빈번할 경우 빠른 성능 보장)

D. Stack
i. 후입선출, Last In First Out 방식
ii. Top 을 통해 삽입하는 연산 ➔ Push
iii. Top 을 통한 삭제하는 연산 ➔ Pop
iv. 가장 마지막에 삽입된 자료가 가장 먼저 삭제된다
v. 예 ➔ 역순 문자열 생성, 실행 취소, 웹브라우저 뒤로가기

E. Queue
i. 선입선출, First In First Out 방식
ii. 삭제 연산만 수행되는 곳을 Front
iii. 삽입 연산만 이루어지는 곳은 Rear
iv. 큐의 Rear 에서 이루어지는 삽입 연산 ➔ enqueuer
v. Front 에서 이루어지는 삭제 연산 ➔ dequeuer
vi. 예 ➔ 은행 업무, 프린터 인쇄 대기열

101. Weak vs Strong vs Soft vs Phantom Reference

A. Weak Reference
i. GC 가 발생하면 무조건 수거됨
ii. Weak Reference 가 사라지는 시점이 GC 의 실행 주기와 일치
iii. 짧은 주기에 자주 사용되는 객체를 캐시 할 때 유용

B. String Reference
i. 일반적으로 new 를 통해서 객체를 생성하게 되면 생기게 되는 참조
ii. 강한 참조를 통해 참조되고 있는 객체는 GC 대상에서 제외됨

C. Soft Reference
i. GC 에 의해 수거될 수도 있고 안 될 수도 있다
ii. 메모리에 충분한 여유가 있다면 GC 가 수행되더라도 수거되지 않음
iii. Out Of Memory 의 시점에 가깝다면 수거될 확률 높음

D. Phantom Reference

i. GC 가 돌기 전 (즉, '이것은 GC 대상이야' 라고 결정할 때, Finalize()호출 후) 메모리에서 정리됨
ii. 주로 GC 되기 전 Cleanup 해야 할 때 사용

102. TCP & UDP 차이점

A. TCP
i. Transmission Control Protocol, 전송을 제어하는 프로토콜
ii. 연속성보다 신뢰성있는 전송이 중요할 때 사용하는 프로토콜
iii. TCP 서버와 클라이언트는 1:1 연결
iv. 수신 여부 확인
v. 전송 순서 보장
vi. 연결형 서비스
vii. 가상 회선 방식

B. UDP
i. User Datagram Protocol, 사용자 데이터그램 프로토콜
ii. 신뢰성보다는 연속성, 성능이 중요한 서비스, 실시간 서비스(Streaming)에 자주 사용
iii. UDP 서버와 클라이언트는 1:1, 1:N, N:M 으로 연결 가능
iv. 수신 여부 확인하지 않음
v. 전송 순서 바뀔 수 있음
vi. 비연결형 서비스
vii. 데이터그램 방식

103. 안드로이드 TV Platform 과 모바일과의 차이는 무엇인가?

A. Android TV App 은 스마트폰 / 태블릿용 앱과 동일한 아키텍쳐 사용
B. Android 용 앱 개발에 관해 알고 있는 내용을 가지고 새로운 TV 앱을 빌드하거나 기존 앱을 확장하여 TV 기기에서도 실행 할 수 있음
C. UX/UI 적인 부분만 다르다

104. 현재 나의 스킬셋에 대해 설명해보아라.

A. 언어 ➔ Java, Kotlin, C/C++, Python
B. IDE & Tools ➔ Eclipse, Visual Code, Android Studio, SQLiteSpy
C. Database ➔ Android SQLite
D. OS ➔ Linux, Windows
E. Code Management ➔ SVN, GIT
F. App Development ➔ Bluetooth API, Framework API, 3rd Party Library (Glide, Picasso, OkHttp, Retrofit, RecyclerView, Gson
G. Issue Tracker ➔ Jira, Confluence, Redmine
H. Source Management ➔ Jenkins, Gerrit, Opengrok
I. Cloud ➔ 앱에서 사내 Cloud 연동 (로그인/컨텐츠 다운로드)

105. Launcher 구현 / 유지보수 업무에 대해 자세히 설명해보아라

A. User 에게 가장 맨 처음 보여지는 화면 구성
B. 방송 Video Play
C. 추천리스트, 녹화리스트, Home 화면, Netflix, 3rd Party VOD, 앱 리스트, Setting 메뉴 구성

106. 능숙한 것이 있는가?

A. 어떠한 앱이든 초기 개발 단계부터 릴리즈까지 가능하다

107. 영어 이름이 있는가?

A. 있다. Aiden

108. 앱만 제공하는 부서가 존재하는가? 3rd Party 는 주로 어느 업체인가?

A. 과거에는 존재하였으나 현재는 존재하지 않고 Android 팀에 개발자들 구성
B. RCU 업체, TV Program 편성표 앱, Launcher 앱 등 시나리오에 따라 다르다

109. 왜 자바만 그동안 사용했는가? 다른 언어 도입을 시도 해봤는가?

A. 이전 코드 구현 내용들이 자바로 구성되어 있었기 때문에 자바만 사용해왔다
B. 코틀린 언어에 대한 세미나 진행 했다

110. 경력 연차에 비해 그동안 전통적인 방법(MVC)로만 개발을 하신 것 같은데 신기술(패턴, 언어 등) 도입이 어려웠나요?

A. 어렵지는 않았으나 그 동안 개발을 익숙한 방향으로만 해왔다.

B. 여러 개발자들과 제품 개발을 목적으로 개발을 해왔으며 이 과정에서 신기술을 도입하려면 세미나부터 시작해서 많은 논의(회의)가 필요했기 때문에 기존 구현 방법을 유지했으며 대신 코드 구조 설계 시 최대한 비즈니스 로직과 UI 로직을 분리하고 파생모델/Feature 에 대해 고민하여 개발을 진행하였다.

111. 도입 실패 시 적극적으로 피력했는가?

A. 지식 공유를 위해 세미나 진행한 적은 있다

112. Bluetooth BLE 관련해서 Bluetooth RCU FW Upgrade 서비스 앱 개발 시 BLE 로 RCU 펌웨어 자체를 RCU 로 전송한 것인지? 아니면 Configuration 관련 데이터만 전송한 것인지?

A. RCU 펌웨어 자체를 전송했다

113. Internet Radio APP 개발은 혼자서 진행 하였는가? 어떻게 진행하였는가?

A. 그렇다. 혼자서 진행하였다

B. 앱 개발 시작 후 UX/UI 팀과 시나리오 / GUI 논의 진행

C. 내부 Jira 페이지 활용하여 Task / Sub Task TODO 항목 티켓 진행 순서대로 생성

D. 진행되는 티켓은 In Progress 상태로 State 변경하여 팀 내 공유

E. GUI 적용 / 개발자 테스트 이 후 SW 품질 인증 진행

F. 테스트 후 생성된 이슈 처리

G. 최종 인증 통과 이 후 사내 Portal Server 에 APK 업로드

114. 앱들은 혼자 다 작성한 것인가?

A. 그렇다. UI App 의 경우 주로 혼자서 진행하였으며 TV 방송 데이터 처리를 하는 서비스 구현 작업의 경우는 여러 명이 코딩하였다.

115. Jenkins 가 이력서 있는데 Jenkins 통한 유닛테스트도 한 것 인가? 아니면 구축만 한 것 인가?

A. 유닛 테스트는 진행하지 않았다. Daily Build Job, Code Push 가 일어날 때 마다 빌드하는 CI Job, Release 를 위한 Build Job, 사내 Cloud 에 빌드 이미지를 업로드 하는 Publish Job 구성하여 사용했다

116. 안드로이드 아키텍쳐 패턴은 주로 무엇을 사용했는가?

A. MVC 를 적용했다. 최대한 인터페이스를 사용해서 UI 와 기능 부분이 독립적으로 되게끔 노력해서 코딩했다

B. MVP, MVVM 의 Concept 인지하고 있으며 사용해 본 경험 있다

117. 안드로이드에서 MVC 를 적용했을 때 가장 큰 장점/단점이 무엇인가요?

A. 장점

i. Model, View 로 분리되고 Unit 테스트에서 쉽게 모델만 테스트 가능

ii. 누구나 쉽게 파악 가능, 개발기간이 짧아짐

B. 단점

i. Model 과 View 사이에 의존성 발생 (View 의 UI 갱신을 위해 Model 을 직/간접적으로 참조 ➔ 앱 자체가 커지고 로직이 복잡할수록 유지보수가 힘들어짐

ii. 시간이 지날수록 컨트롤러에 많은 코드가 쌓여 코드가 비대화 (티비티 소스코드의 비대화)

iii. Controller 가 안드로이드 API 에 깊게 종속되므로 유닛 테스트가 어려움

118. TvInput Service 는 혼자 구현한 것인가?

A. 아니다. 모듈 별로 담당자들이 존재했다. 나는 IPC 통신하는 서비스를 개발했다. 외부 3rd Party 앱 / 개발하는 다른 앱으로부터 전달 되어지는 메시지에 따라 해당하는 모듈로 연결해주는 작업을 하였다.

119. BLE Gatt Data 사용해서 처리 했다는게 RCU FW 데이터를 받았다는 것인가? FW Configuration 을 받은 것인가?

A. RCU FW 데이터를 BLE Gatt 를 이용하여 RCU 에 전달하였다

120. 출장이 잦은데 외국인과 소통할 기회가 많았나? 언어는 영어를 사용 했었나?

A. 항상 외국인이 있지는 않았다. 누구와 일하는 냐에 따라 다르다. 스페인 출장 때 현지 개발자들과 소통했으며 언어는 영어를 사용했다. 베트남 개발자들과 일할 때도 영어 사용했었고 주로 Slack, Mail 그리고 Phone Call 로 소통했다

121. 일과 관련해서 최종 목표는?

A. 팀을 이끄는 리더

B. Mobile - IoT 분야 ~ Server 까지 모든 SW 분야에 대해서 알고 할 줄 아는 것이 목표

122. 소프트웨어 아키텍쳐 사용 MVC, MVP, MVVM 에 대한 설명해보아라.

A. MVC
B. MVP
C. MVVM

123. 앞으로의 개발자로서 테크를 어떻게 하고 싶은가?

A. 개발자로서 경험을 더 쌓은 이후 매니저/팀 리더 역할을 해보고 싶다
B. 개발업무에서의 재미, 원활한 대인관계 그리고 세심한 성격을 가졌다고 생각한다. 그래서 매니저를 하면 잘할수 있을 것 같다. 매니저 역할의 일부분이긴 하지만 과거 혼자 해외 출장을 약 1 달간 가서 현지 개발자들과 커뮤니케이션하면서 다음 릴리즈 일정을 생각하고 개발도 개발이지만 개발 진행에 대한 계획을 하면서 재미와 보람을 많이 느꼈었기 때문에 매니저 테크를 생각한다.

124. 개발 할 때 힘들었던 점은 무엇인가?

A. 사람이 힘들었던 점은 무엇인가?
   i. 비협조적이거나 방어적인 사람과 일할 때 힘들었다
   ii. 업무가 미숙한 사람과는 힘들지 않았다. Jira 로 Task 를 나눠서 할 수 있는 부분들을 서로 논의하면서 진행해서 무리가 없었다
   iii. 일은 진행해야 하기 때문에 태도에 상관없이 계속 커뮤니케이션을 시도했다

B. 업무가 힘들었던 점은 무엇인가?
   i. 거리 상 멀리 있어 커뮤니케이션 핑퐁이 잦을 때 힘들었다
   ii. 담당자 부재로 인수 인계가 제대로 되지 않아 과거 시나리오 파악이 어려워 업무가 힘든 적이 있었다

.

125. 다른 언어나 기술 사용을 해보고 싶다면 무엇을 적용해서 해보고 싶은가?

A. 코틀린 언어의 완전 적용
   i. 소스 코드의 간결화, 가독성 증가 목표
B. Android 아키텍쳐 패턴 재 정리
   i. 불필요한 패턴 정리, 최대한 UI / 기능적인 부분 분리

126. 진행했던 코틀린 세미나 내용 중 인상 깊었던 내용은 무엇이고 세미나는 몇 차례 정도 진행하였는가?

A. 세미나는 1 회 진행하였다

B. 특징 (기억나는 특징 나열)

i. Nullable, Non-Nullable 타입으로 Null 안정성 증가

ii. Coroutines (동시성 프로그래밍) – 루틴을 진행하는 중간에 멈추어서, 즉, Return 없이 특정 위치로 돌아갔다가 다시 원래 위치로 돌아와 나머지 루틴을 수행 가능

iii. Data 클래스

iv. 자바보다 간결한 문법

v. 컴파일 속도가 자바보다 빠르고 런타임 사이즈가 작다

vi. 자바와 100% 상호 호환, JVM 에서 동작

vii. 안드로이드 공식 사이트에 첫번째 예제가 코틀린으로 되어 있다

127. 회사에서 영어 쓸 일이 많은가요?

A. 그렇다. 고객사가 외국에 있어서 주로 업무 때 영어 메일로 커뮤니케이션 많이 한다.

128. 외국인과 특히 베트남 엔지니어들과 협업할 때 총괄 매니징은 누가 하였는지? 아니면 본인이 직접 하였는가?

A. 전체적인 프로젝트를 매니징 하는 사람이 팀 내 존재, 그리고 특정 모듈 담당자들의 개발 일정을 매니징하기 위한 매니져가 따로 존재하였다

B. 매일 같은 시각, 공간에서 스크럼을 진행하였다

C. 개발 중에 일어나는 커뮤니케이션은 개발자들이 해당 담당자들과 직접 했다

129. 크로스 플랫폼에 대해 어떻게 생각하나요?

A. 크로스 플랫폼 프레임워크로는 Flutter 와 React Native 가 있다

B. 효율 보장은 확실하다.

C. UX 가 얼마나 Native 에 가깝느냐에 따라 유지할 지 Native 로 갈 지를 정한다. Native 로 앱을 만들게 되면 조금 더 OS 특성에 가까운 UI Component 를 사용할 수 있다.

D. React Native 는 웹을 경험한 개발자가 쉽게 접근 가능

E. Flutter 는 Dart 언어 학습 필요

130. iOS 개발은 관심 없나요?

A. 관심 없지는 않으나 Android 업무만 주로 해와서 개발 경험 해본 적이 없다

B. 기회가 주어진다면 업무 외 시간을 활용하여 iOS 개발 영역을 공부해서 개발 업무를 해보고 싶다

131. 플랫폼 쪽 일도 경험 있나요? Amlogic, Broadcom 칩셋 관련 작업을 해보았나요?

A. AmLogic / Broadcom Module 의 특정 기능을 Java Layer 에서 사용하기 위해 JNI 를 연결해 본적이 있고 이슈 디버깅 시 주로 전체 Android 로그를 보기 때문에 소스도 많이 봐왔다.

132. 회사에 품질 팀이 있는가? 프로세스가 어떻게 되는지 설명해달라

A. 프로세스
   i. Software 개발 완료 후 품질 팀에 빌드 이미지 인증 요청
   ii. 품질 테스트 Stage1 ~ 3 까지 진행
   iii. 품질 팀에서 이슈를 Jira 이슈 티켓으로 발행
   iv. TM 에게 이슈 할당
   v. TM 이 각 담당자들에게 이슈 재 분배
   vi. 담당자들이 이슈 처리

133. Android Version Upgrade 될 때 API 변화에 어떻게 대응하였는가?

A. Android Version 호환성 대비
   i. 예를 최신 버전이 Android 11 이라 하면 각 앱들의 TargetSdkVersion 을 11(30)으로 올리지 않으면 11 에 적용되는 변경 사항들이 적용되지 않게 되므로 이 시간을 두고 호환성 대비를 할 수 있다.

134. 구글 AAC Jetpack 사용해보았는가?

A. Android Architecture Component
B. 기본 Android Platform 에 속하지 않는 별도의 라이브러리로서 제공
C. 모듈화된 코딩을 위해 DataBinding, LiveData, ViewModel, Room 등의 라이브러리 제공
D. JetPack 구성요소
   i. WorkManager
   ii. Navigation
   iii. Paging
   iv. Slices
   v. Android KTX (확장 프로그램)

vi. Data Binding

vii. LifeCycle

viii. LiveData(+ViewModel)

1. Observable 가능한 DataHolder Class
2. 뷰에서 LiveData 를 관찰하게 되면 데이터가 변경될 때 내부적으로 자동적으로 알려주게 됨
3. Activity 나 Fragment 의 생명주기를 인지
4. Activity 가 화면 위에 활성화 되어 있을 때만 UI 변경 등의 기능 동작을 하므로 Memory leak 발생을 줄여준다 (Destroy 상태에서는 동작하지 않음)

ix. Room

1. SQLite 코드를 직접 작성하는 경우, 직접 테이블 생성 및 쿼리문을 일일이 변수에 저장해야 했지만 Room 을 사용하면 직관적이고 편리하게 사용 가능

135. 로그 관리는 어떻게 하는가?

A. 사내 Splunk 시스템 활용 / Splunk API 호출을 위해 각 앱에서 해당 부분 구현

136. 테스트 자동화, UI 테스트는 어떻게 진행 하였는가?

A. UI 테스트는 특정 이슈 발생할 시나리오에 대한 스크립트 생성 후 에이징 테스트 진행

137. 모든 위젯 사용해보았는가?

A. 그렇다. 안드로이드에서 제공하는 거의 모든 UI 위젯들을 사용해 보았다.

138. 외부라이브러리를 포팅해서 사용해보았는가?

A. 그렇다. 3rd Party 업체에서 제공하는 라이브러리나 같이 협업하는 팀에서 제공하는 라이브러리를 사용하고 있다

139. REST API 서버 구축해 본 적 있는가?

A. 과거 프로젝트 시 팀원이 API 서버를 구축하였고 이슈 검토를 위해 소스를 본 적은 있다

140. 깃허브나 플레이 스토어에 업로드 해 본 경험이 있나요?

A. GitHub 에는 꾸준하게 소스를 올리려고 노력하고 있다

B. 플레이 스토어에 업로드 해 본 경험은 없다. VOD/Radio 앱 개발을 한 적이 있는데 이 앱 개발 완료 후 사내 서버에 올려본 적은 있다.

C. 플레이 스토어에 개발 완료된 APK 를 업로드하는 방법은 숙지하고 있다

141. 안드로이드 빌드에 대해서 설명하라

A. Ant, IntelliJ

142. 웹앱, 하이브리드앱에 대해 설명해보아라

A. 웹 앱
   i. 웹 기술로 구현하는 앱
   ii. HTML, CSS, JS, PHP, ASP 사용해 만들어진 앱
   iii. 기본적으로 브라우저 주소창이 존재 (숨길수 있음)
   iv. 업데이트 빠르다, 앱 스토어 배포할 필요가 없다
   v. 센서, 카메라 등의 기능 활용이 어렵다
   vi. 스마트폰 자체 운영체제에서 제공하는 API 는 사용할 수 없다

B. 하이브리드 앱
   i. 네이티브 + 웹앱의 기술을 함께 사용하는 앱
   ii. 콘텐츠 영역은 HTML 기반의 웹앱으로 개발하되 패키징 처리만 각 아이폰, 안드로이드 플랫폼 안에서 함으로써 앱 배포를 가능케 한다
   iii. React Native, Flutter, PhoneGap, Cordova, Ionic
   iv. 웹 앱과 다르게 모바일 API 사용 가능, 하드웨어 제어 가능
   v. 네이티브 앱만큼 기능을 제공하지 않는다
   vi. 앱스토어 배포가 필수적, 업데이트 반영이 느리다

143. React Native, Flutter 에 대해 설명해 보아라

A. React Native
   i. 페이스북이 개발한 오픈 소스 모바일 앱 프레임워크
   ii. React(Javascript)의 문법만 익혀서 개발하면 된다
   iii. 핫 리로드, 라이브 리로드 가능
   iv. Code-Push (고객에게 문제점 발생 시 빠른 대응 가능)
   v. 패턴 지원, Redux, MobX

B. Flutter
   i. 구글이 개발한 오픈 소스 모바일 앱 개발 프레임워크
   ii. Dart 언어 사용

144. Lambda 에 대해서 아는가?

A. 익명 함수 (Anonymous Functions)

B. 코드 간결성

C. 매개변수, ->(화살표), 함수 몸체로 구성

145. Steam 이란 무엇인가?

A. 다양한 데이터를 표준화 된 방법으로 다루기 위한 라이브러리

B. Java 8 부터 추가됨

C. 데이터를 변경하지 않는다

D. 구성

i. Stream() ➔ 스트림 생성

ii. Filter ➔ 중간 연산 (스트림 변환) ➔ 연속해서 수행 가능

iii. Count ➔ 최종 연산 (스트림 사용) ➔ 마지막에 단 한 번만 사용 가능

1. 예 ➔ example.stream().filter(x -> x < 2).count
2. 예 ➔ IntStream.range(1, 11).filter(i -> i%2 == 0).forEach(System.out::println);

146. 프레임워크, 라이브러리 둘 중 개발자 입장에서 무엇이 더 좋은가?

A. 프레임워크

i. 특정 프로그램을 개발하기 위한 여러 요소들과 매뉴얼인 룰을 제공하는 프로그램

ii. 전체적인 흐름을 자체적으로 가지고 있다

iii. 프로그램을 쉽게 만들 수 있게 하는 데에 목적

iv. 자유도 낮음, 꼭 써야 되는 것과 지켜야 되는 룰이 있다

B. 라이브러리

i. 소프트웨어를 개발하기 쉽게 어떤 기능을 제공하는 도구들

ii. 도구의 모음

iii. 프로그래머가 전체적인 흐름을 가지고 있다

iv. 프로그램을 쉽게 만들 수 있게 하는 데에 목적

v. 자유도 높다, 개발자 마음

147. Android TV 에서 달라진 점

A. UX/UI 적인 면 외에는 모바일과 거의 유사하다

B. Leanback UI 적용 (DPAD 에 적합한 위젯과 템플릿 프래그먼트를 사용하여 Android TV 기기용 앱 개발)

C. Leanback Support Library

i. BrowseFragment (검색 버튼, 메뉴, 컨텐츠 리스트)

ii. SearchFragment (검색 버튼 눌렀을 때, 검색어 입력 / 결과 표시하는 화면)

iii. DetailsFragment (항목 상세 정보 표시)

iv. VerticalGridFragment (항목 수가 많을 때 필수적인 프래그먼트)

148. Kotlin 사용해 본적 있는가?

A. 사용해 본 적 있다. 현재 일부 프로젝트는 코틀린으로 변경 / 시작하고 있다. 그리고 과거 팀 내 세미나 진행했었고 간단한 앱 제작하여 GitHub 에 올리는 중이다

149. 현재 Kotlin 을 사용해서 어떤 앱을 현업에서 만들고 있나요?

A. Launcher 에서 추천리스트 Feature 구현을 코틀린을 사용해서 구현 중에 있다.

B. 추천 리스트는 고객사 OP Server 로부터 컨텐츠 정보를 JSON 형태로 받아온 다음 DB 에 저장하고 Launcher 에서 Home 키나 해당 메뉴 진입 시 DB Query 를 하여 메뉴를 그리고 있다.

150. Git 써 본적 있어요?

A. 네, 안드로이드 업무할 때 계속 Git 을 사용해왔다

B. 앱 수정 내용 Commit 시 / Repository 구성 / Amlogic Patch 적용 작업 때 사용한다

151. Git-flow 에 대해서 설명해보라

A. SW 의 소스코드를 관리하고 출시하기 위한 브랜치 관리 전략

B. 5 가지 브랜치

i. Master ➔ 제품 출시용

ii. Develop ➔ 다음 출시 버전 개발용

iii. Feature ➔ 기능 개발용

iv. Release ➔ 이번 출시 버전 준비용

v. Hotfix ➔ 출시 버전에서 발생한 버그 수정용

C. 개발 흐름

i. Master & Develop 브랜치 생성

ii. 새로운 추가 작업이 있는 경우 Develop 에서 Feature 브랜치를 생성

iii. 기능 추가 작업 완료 시 Feature 는 Develop 브랜치로 Merge
iv. QA 를 위해 Develop 에서 Release 브랜치를 생성
v. QA 를 진행하면서 발생한 버그들은 Release 에 수정
vi. QA 끝나면 Release 브랜치를 Develop 과 Master 로 각각 Merge
vii. Hotfix 는 Master 와 Develop 브랜치로 각각 Merge

152. 코드 리뷰는 어떠한 식으로 진행하는가?

A. Gerrit Review 시스템 또는 오프라인 회의를 개최한다.
   i. 기능 수정 / 추가 작업 완료 후 소스 코드를 Gerrit 에 Push
   ii. 관련 담당자 / 매니저를 Reviewer 로 등록
   iii. Reviewer 로부터 리뷰가 완료되면 담당자 직접 최종 Push 를 한다

153. 개발하고 나서 테스트는 어떤 식으로 진행하는가?

A. API 테스트의 경우 따로 Test 용 UI 앱을 만들어 결과를 UI 상에서 보면서 테스트
B. UI 테스트의 경우는 수동으로 하거나 에이징 테스트 진행
   i. 에이징 테스트의 경우 스크립트 작성하여 진행

154. 외부라이브러리는 어떤 것들을 사용해 봤나?

A. UI 작업 시 이미지 로딩 때는 Glide, Picasso, Leanback, RecyclerView
B. Network 작업 시 OkHttp
C. 시스템 보안 체크 시 RootBeer
D. JSON Object → Java Object 역직렬화, 직렬화 작업 시 Gson

155. 여기 말고도 다른 곳도 지원 하셨나요?

A. 이직 관련해서 신중하게 생각을 해야 되고 무슨 일을 할 때 신중한 편이라 현재는 여기만 지원한 상태이다

156. 굳이 OTT/STB 에서 모바일로 변경 하려고 하는 이유는?

A. OTT/STB/Mobile 개발을 다 해보았다
B. 요즘 모든 서비스들이 모바일로 집중되고 있고 이러한 트렌드를 쫓아가기 위해서 모바일 쪽을 특화시키고자 한다

157. 모바일 관련해서 해본 프로젝트가 있는가?

A. Mobile RCU 앱 유지보수 업무를 했었다
B. DLNA 를 사용해서 OTT 와 연결했다
C. 개인적으로 모바일 앱을 만들어서 GitHub 에 업로드 중이다
D. 과거 해커톤 프로젝트 참가하여 카메라를 이용한 번역 앱을 만든 경험이 있다.
E. 음악 리스트를 보여주는 앱을 구현해 본 경험이 있다. (OkHttp + RecyclerView + Database 연동)

158. 혼자서 일을 오랫동안 하신 것 같은데 같이 일할 때와 혼자 일할 때 둘 중 뭐가 편하냐?

A. 서로 담당하는 모듈이나 앱이 다르더라도 같이 일하는 것이 더 편한 것 같다
B. 서로 봉착 해있는 문제에 대해 쉽게 논의가 가능하고 서로의 코드를 리뷰해줄 수 있어서 좋은 것 같다
C. 더 좋은 코드를 생산해 낼 수 있어서 같이 일하는 것이 더 효과적이라고 생각한다

159. 같이 일할 때 애로사항?

A. 같이 일할 때 애로사항은 크게 없었다.
B. 서로 구조나 코드에 대해 논의할 수 있어서 좋았다.
C. 모든 것은 대화를 많이 하면 해결해 나갈 수 있는 것 같다

160. CI/CD 관련해서 한 일?

A. CI ➔ 지속적 통합 (Continuous Integration)
   i. Jenkins CI Job 을 생성하여 코드 Push 가 일어날 때마다 소스 빌드를 하여 버그를 찾아내는 데에 도움을 주었다
B. CD ➔ 지속적 제공 (Continuous Delivery / Continuous Deployment)
   i. Daily Image / Release Image / 사내 Cloud 에 Publish 하는 Job 구성하였다

161. 8 년차이면 전문가가 될 테크를 타신 것 같은데 왜 갑자기 이직하려고 하나요?

A. OTT/STB/Mobile 업무를 다 진행해봤었다. 최근 모든 IT 서비스들이 모바일에서 이루어지고 있고 이러한 시대의 트렌드에 맞추기 위해 모바일 업무를 조금 더 특화시키고 싶었고 BT/DLNA/WiFi 이용한 Mobile 중심의 IoT 기기 연동에도 관심이 많아서 이직하려고 한다.
B. 현재 다니고 있는 회사보다 큰 기업에서 모바일 업무를 수행해보고 싶어서 이직하려고 한다

162. Leanback UI 라이브러리 내용 중 무엇을 사용해 봤나요?

A. VOD 앱 개발 시 BrowseFragment, SearchFragment 등을 사용해보았다

163. Activity 안에 굳이 프래그먼트로 나눠서 설계하는 이유는 무엇입니까?

A. Activity 를 변경하지 않아도 쉽게 View 를 변경할 수 있어서 많이 사용

B. 화면 사이에 데이터 전달에 용이하다 (분리된 Activity 간에는 데이터 전달이 어렵고 한정적인 Class 만 쓸 수 있으나 Fragment 는 하나의 Activity 의 컨텍스트 안에 있기 때문에 데이터를 쉽게 공유 가능)

C. 디바이스 차이를 다루기 위해 (태블릿, 폰 간의 차이)

   i. CustomView 로도 가능하다 같은 액티비티에서 넣었다 뺐다 하는 작업이 번거롭고 이슈를 만들 가능성이 높다

D. 다른 액티비티에서도 사용 할 수 있어 재사용성이 뛰어남

E. ViewPager, FragmentPagerAdapter 를 이용하여 Swipe 스타일 구성이 용이

164. 상용화된 제품이 있는가? (상용화된 제품 안에 당신이 만든 앱이 있나요?)

A. 네. 출시된 OTT 제품에 앱이 탑재되어 있다.

B. Radio / Settings / Wizard / Launcher / RcuService 등이 있다

165. 버전 업그레이드로 인한 사용이 불가했던 특정 Class 나 함수 이름이 무엇인지 기억하는가?

A. BluetoothInputDevice 관련 Class 였었는데 기억이 나지 않는다.

166. Java 리플렉션 사용시 이슈 발생한 적이 있는가?

A. 리플렉션 사용할 때 invoke 하기를 원하는 클래스의 package 명과 Class 이름을 명시해줘야 하는데 이 Package 명 또는 Class 명이 안드로이드 버전 업그레이드로 인해 변경되어져서 해당 API 가 동작하지 않았던 경우가 있었다

167. RxJava 사용해 본적 있는가?

A. 관련해서 예제 코드 학습을 하였었고 개념은 숙지하고 있다

168. 상용화된 제품에서 발생한 Crash 같은 이슈는 어떻게 대처하는가?

A. 상용화된 제품에서 Crash 발생 시 유저가 CS 팀에 이슈를 제기

B. 품질팀에서 해당 이슈 재현 방법에 대해 Jira 나 메일에 작성/발행 후 개발팀 매니저에게 할당

C. 개발팀 매니저는 전달받은 이슈를 담당자에게 다시 할당

D. 담당자는 품질팀에서 가이드 준 재현 방법대로 재현하여 이슈 수정 시도

E. 수정이 완료되면 이슈 티켓을 개발팀 매니저에게 할당
F. 개발팀 매니저는 최종 확인 후 이슈 티켓을 다시 품질팀에게 할당
G. 품질팀에서 다시 재현/이슈 검토 후 이상 유무 확인
H. 이상 없으면 APK 릴리즈 / Firmware 릴리즈로 해당 문제 대처

169. 과거 상용화된 프로젝트에서 발생했던 Crash 가 어떠한 Crash 였는지 기억하는가?

A. NullPointerException Crash 였던 것으로 기억한다.
B. 상용화된 박스에서 로그인 시도 후 결과 토큰 값이 Null 상태라서 참조할 때 NullPointerException 이 발생했다

170. TDD 해본 적 있는가?

A. TDD
   i. Test Driven Development, 테스트 주도 개발 (테스트가 개발을 이끌어 나간다)
      ➔ 매우 짧은 개발 사이클을 반복하는 소프트웨어 개발 프로세스 중 하나
   ii. 테스트를 먼저 만들고 테스트를 통과하기 위한 것을 짜는 것
   iii. 불확실성이 높을 때 TDD 수행한다 (누가 유지보수 할 지, 코드 변경이 많을 때, 요구조건이 바뀔 가능성이 높을 때, 처음 개발하는 프로그램)
   iv. 피드백 증가
   v. 개발 시간이 증가한다. (개발 시간 정의가 애매, 본인이 다했다고 생각할 때까지? 기능 구현이 끝났을 때 까지?)
   vi. 버그 감소
   vii. 코드 복잡도 감소
   viii. 일하는 방식, 협력 피드백에 중점을 둔 개발 방식
   ix. 애자일 방법론 중 하나인 eXtream Programming(XP)의 Test-First 개념에 기반을 둔 단순한 설계를 중요시
   x. 개발 중 후반에 수정되는 내용에 대해서 깨지는 테스트 케이스를 계속 유지하면서 가져가기가 어렵다
B. BDD
   i. Behavior Driven Development, 행동 중심 개발
   ii. 시나리오를 기반으로 테스트 케이스를 작성, 함수 단위 테스트 권장하지 않는다
   iii. TDD 와 유사하나 개발자가 아닌 일반 사용자가 봐도 이해할 수 있는 자연어에 가깝게 작성한다

171. 애자일 개발 프로세스를 따르는가?

A. 그렇다. 매일 정해진 시간에 정해진 장소에서 모듈 담당자들 끼리 짧은 스크럼을 진행한다

B. 소스 코드 코딩 시에도 테스트 코드 작성을 통해 최대한 TDD 를 따르려고 한다.

i. 참고

1. 애자일 개발 방법론 ➔ 일정한 주기를 가지고 끊임없이 프로토타입을 만들어내며 그때 그때 필요한 요구를 더하고 수정하면서 SW 를 만들어 가는 Adaptive Style

172. 만약 입사를 했는데 모바일이 아닌 다시 Android TV 개발을 해라고 한다면 어떻게 하실 겁니까?

A. 원하던 바다. 지원이 필요하다면 개발 지원 가능하다

B. 모바일로 지원하였기 때문에 모바일 쪽 업무를 중심으로 하고 싶으나 다양한 것을 경험해보고 싶기 때문에 지원 가능하다

C. 회사라는 곳이 하고 싶은 일만 하면서 지낼 수는 없다고 생각한다. 저의 Android TV 개발 경험을 바탕으로 지원을 해야 되는 상황이 있다고 하면 적극적으로 지원할 생각이 있다

173. 자신의 장점

A. 기술적 장점

i. 앱 개발 외에도 Broadcom / Amlogic AOSP SDK 에 대한 패치 작업, Repository 구성 그리고 App 포팅 업무도 가능

ii. 앱 개발 외 Android OTT Box Google 인증 프로세스 경험

iii. 앱 개발의 경우 어떠한 앱이든 처음부터 끝까지 개발이 가능

B. 성격적 장점

i. 긍정적임, 최악의 상황에서도 좋은 면을 보려고 한다

ii. 성실한 스타일, 끈기가 있음

174. 마지막으로 어필하고자 하는 것은 무엇입니까?

A. 8 년 동안 안드로이드 개발 경험을 바탕으로 초기 개발부터 끝까지 개발이 가능하다.

B. 늘 근면 성실한 태도로 회사 생활에 임했다.

C. 새로운 것을 받아들이는 데에 거부감이 없으며 앞으로 더 노력하고 잘할 자신이 있다

D. 현재도 계속 그동안 해왔던 것들에 대해 복기하고 새로운 것을 공부 중이다. 많은 것들을 경험해보고 싶다.

175. 업무 우선순위는 어떻게 정해서 진행하나요?

A. 팀 내 기술 매니저와 이슈 티켓 우선 순위에 대해서 논의한 뒤에 정한다

176. 학부 때 따로 프로젝트한 경험은?

A. 해커톤 프로젝트에 참가해서 카메라를 이용한 외국어 번역 앱을 만든 경험이 있다

B. 1 박 2 일 동안 앱 개발하여 최종 발표를 했다

177. 나이가 많은 상태에서 학교 입학 했는데 트러블 없었나

A. 없었다. 오히려 나이가 많아서 서로 잘 지낸 면도 있는 것 같다.

178. 취미는 있나

A. 있다.

179. 집 위치는 어디입니까?

A. 서울이다

180. 주식 펀드 해 보았나요?

A. 주식앱 통해서 몇번 해보았고 펀드는 은행에서 적립식 펀드 가입해본적이 있다.

181. 젠킨스 구축 사항이 이력서에 적혀있는데 왜 이 업무를 했었나요?

A. Android 신사업 개발팀에 첫 입사 했을 때 Android Build / 형상관리 관련한 시스템이 갖춰지지 않은 상태였다. 그래서 이 업무를 하게 되었다

182. 앱 만들면서 3rd Party 라이브러리 포팅 경험이 있나요?

A. 네. 이때까지 제작했던 앱들 거의 대부분이 3rd Party 라이브러리를 참조하고 있다

183. 금융상품에 대한 이해가 있나요?

A. 개발 관련 지식은 있지만 금융 상품에 대한 지식은 부족하다. 개발 공부와 더불어 같이 공부하면서 따라가도록 하겠다. 업무 외 시간을 활용해서라도 이해도를 높이기 위해 노력하겠다.

184. 부족한 금융 지식에 대한 것을 어떻게 메꿀 것인가?

A. 업무 외 시간 때 서적/인터넷/동영상 강의를 통해 지식을 쌓도록 하겠다

185. 현재 연봉과 희망 연봉은 어떻게 됩니까?

A. 현재 연봉은 XXXX 만원 이며, 희망 연봉은 XXXX 입니다.

i. 계약 연봉으로 XXXX 를 희망한다

186. 현재 진행중인 프로젝트 및 본인이 하는 일은 무엇입니까?

A. 일본 최대 규모 Cable TV 사업자인 JCOM 의 셋탑박스 제품 개발 프로젝트 진행 중이다.

B. Launcher App, DLNA 녹화 서비스, 3rd Party 에 API 전달을 위한 서비스 개발/유지보수 중이고 이슈 발생 시 APK 구분 없이 전체적으로 로그를 분석하면서 처리 중이다

187. 팀 내 개발자 구성 및 인원 수는 어떻게 되는가?

A. Android 팀 내 App / Framework / Media / Platform 파트로 구성

B. 각 파트마다 3~4 명 정도 배치, 전체 인원 수는 대략 12 명 정도이다

188. 앱 개발 업무를 어떤 식으로 맡아서 하는가? (혼자? 다같이?)

A. 단순 UI 앱 / 서비스의 경우 혼자서 진행, 규모에 따라 다르다

B. 방송 데이터 처리를 위한 서비스 개발의 경우 모듈 별로 개발자가 할당된다

189. 출시 경험이 있나요?

A. 플레이 스토어에 TestApp 을 올려본 경험이 있다

190. 업무가 기존에 하던 업무에서 많이 바뀌는데 걱정스러운 것들이 있나요?

A. Android UX/UI 적인 부분은 다르지만 기본 구조는 같기 때문에 특별히 걱정되는 부분은 없다

B. 대부분의 안드로이드 컴포넌트를 사용해보았기 때문에 구현적으로는 크게 걱정되는 것이 없다.

191. 금융 앱들을 봤을 때 개선이 되어야 할 점이 있어 보이나요?

A. 대부분 앱들이 사용하기에는 큰 불편함이 없어 보인다

B. GUI 가 조금 더 세련되었으면 하는 생각이 있다

192. 입사를 하게 되면 인수인계 시간 얼마나 필요한가요?

A. 1 달 정도의 시간을 주실 수 있다면 1 달 이내에 인수인계하고 정리 가능하다

193. 업무 하면서 가장 보람 있었던 것이나 기술적으로 보람 있었던 것이 있는가?

A. Spain 텔레콤 사 Orange STB OTT 프로젝트 때 보람을 느꼈다
   i. 현지 개발자와 영어로 성공적인 의사소통
   ii. IPTV 방송 데이터 처리 서비스 개발 완료
   iii. IPTV 컨텐츠 재생이 되는 것 현지에서 실제로 확인

194. BLE (저전력 블루투스) 에 대해 설명하세요

A. Bluetooth Classic 과의 큰 차이는 훨씬 더 적은 전력을 이용하여 Classic 과 비슷한 수준의 무선 통신을 할 수 있다
B. 통신과정 : 스캔 ➔ 연결 ➔ 통신
C. 연산종류 : Read / Write / Notification / Indication
D. Bluetooth GATT 구조 : 여러 개의 BluetoothService 를 가지고 각 BluetoothService 는 여러 개의 BluetoothCharacteristic 을 가지고 각 BluetoothCharacteristic 은 여러 개의 BluetoothDescriptor 를 가진다
E. 기본 통신은 바이트 배열을 보내고 받는다
F. BLE 통신 시 최대 송신 가능한 바이트는 20 바이트

195. 아쉬움이 남는 프로젝트가 있나요?

A. 과거 프로젝트에서 RCU 업체와 협업할 때 거리상 문제로 커뮤니케이션이 어려워서 힘들었던 적이 있다. HW 문제가 있을 때 수정된 HW 를 받는 데까지 2~3 일 정도가 소요되었었고 메일이나 전화로 업무를 주로 해야해서 힘든 면이 있었다. 하지만 납기일은 지켰고 결국 H5/D'live OTT 박스 출시까지 해낼 수가 있었다.

196. 자바에서 다중 상속이 왜 안 되는지 아십니까?

A. 다이아몬드 문제 발생
   i. 최상위 부모 클래스 GrandParent 가 A 라는 메소드를 가지고 있을 때 Parent1, Parent2 클래스가 각각 A 메소드를 오버라이딩해서 구현하였다면, Parent1, Parent2 를 모두 상속받은 Child 클래스 입장에서는 어떤 부모의 A 메소드를 사용해야 할 지를 몰라 충돌이 생기게 됨
   ii. C++에서는 이 문제를 개발자에게 일임
   iii. Java 는 내부적으로 구현이 불가하도록 막아놓음
   iv. 인터페이스는 다중 상속 가능

1. 기능에 대한 선언만 있으므로 가능
2. Java 8 에서는 default method 가 추가되었기 때문에 내부적으로 코드 구현이 가능 ➔ 이 경우는 Class 의 다중 상속처럼 컴파일 진행되지 않음. 메소드를 새롭게 오버라이딩하면 충돌 회피 가능

197. 학부시절 때 성적이 상당히 좋은데 상대 평가입니까? 절대 평가입니까?

A. 절대 평가인 과목도 있었으나 거의 대부분 상대 평가입니다

198. 무슨 과목이 가장 좋았습니까?

A. Java, 알고리즘, 자료구조, 컴퓨터 구조, 논리회로, 리눅스, 데이터베이스, 웹프로그래밍, 안드로이드 프로그래밍, C, C++, C#, MFC

199. 대규모 프로젝트 진행한 것에 대해서 설명해달라

A. 일본 최대 통신사 JCOM STB 제품 개발 경험
   i. Launcher / 3rd Party 에 API 제공을 위한 서비스 개발
B. 스페인 텔레콤 사 Orange OTT STB 제품 개발 경험
   i. 3rd Party 업체와 같이 개발 (출장)
   ii. 방송 데이터 처리 서비스 구현/유지보수

200. 프로젝트 리딩을 해본 경험이 있습니까?

A. 개발하면서 Vietnam / 타 업체와 같이 일 해본 경험은 많다.
B. 타 업체에게 Android App 포팅/구현 관련해서 가이드 창구 역할을 해본 적 있다

201. 모바일, Android TV 둘 다 진행하라고 하면 어떻게 할 것이냐?

A. 업무 우선 순위 조정을 요청할 것이다
B. 우선 순위 조정이 힘들어 한꺼번에 두 개를 진행해야 된다면 오전/오후로 시간을 쪼개서 하는 방향으로 이야기를 해볼 것 같다

202. 고등학교 졸업 후 공백기간 5 년인데 설명해 줄 수 있나요?

A. 졸업 후 재수를 1 년 하고 나서 군입대를 하였다
B. 그리고 나서 다시 수능 준비를 해서 대학을 들어가게 되었다

203. 대학 4 학년 2 학기때는 수강 과목이 없는데 설명 가능합니까?

A. 1~3 학년 동안 수업을 미리 많이 들어서 4 학년 2 학기 때는 졸업 작품 수업에만 들어갔다

204. 한 회사에서 7~8 년 다닌 것이냐?

A. 그렇다. 첫 회사이다. Android 개발 팀에서 8 년 동안 STB/OTT/Mobile 에 탑재되는 Android Application 개발 업무를 주로 했다

205. Java 외 다른 언어 가능하나요?

A. Kotlin, C/C++, Python, Shell Script 가능하다

206. 안드로이드 플랫폼 소스를 보거나 업무를 해 본적 있습니까?

A. 네 이슈 수정을 위해 플랫폼 Framework 소스 검토 / 수정을 해 본적이 있다

B. Custom Screen Saver 를 적용하거나 Ethernet / Wifi 우선순위 변경, 새로운 API 추가를 위해 작업해본 적이 있다

207. 모바일 부서로 지원했는데 모바일 쪽 프로젝트 해 본적 있습니까?

A. 네. Mobile RCU 앱, Mobile DLNA 앱 디버깅/유지보수 한 적 있다

208. 대규모 앱이나 서비스 구조 설계가 가능합니까?

A. 가능하다. 안드로이드 앱 개발 관련해서는 서비스 – 앱 – 서버 간의 동작 흐름에 대해 설계 / 논의가 가능하다

209. 컴퓨터 아키텍쳐 32 비트, 64 비트가 무엇인지에 대해 설명해보아라

A. 32 비트

i. 메모리를 4GB(4,294,967,296)까지 밖에 인식하지 못함

ii. 최대 64GB 까지 RAM 을 인식할 수 있는 PAE(Physical Address Extension) 툴도 있긴 하다 (윈도우 서버 에디션에서만 사용 가능)

iii. 2 의 32 승 데이터를 한번에 처리 가능

iv. 32 비트 CPU 는 한번에 처리하는 레지스터 크기가 32Bit
(CPU 칩 내부에 레지스터라는 메모리가 존재, 명령어 해석이나 연산 시에 사용되는 데이터를 순간적으로 저장하는 용도)

v. 64 비트 앱 실행 불가능 (앱 시작하는 명령어가 64 비트 만큼 필요한데 32 비트 만큼만 있기 때문에 실행 오류 발생)

B. 64 비트
   i. 4GB 이상의 RAM 을 인식
   ii. 2 의 64 승 데이터를 한번에 처리 가능
   iii. 64 비트 CPU 는 한번에 처리하는 레지스터 크기가 64Bit
   iv. 32 비트 앱 실행 가능

210. 우리가 경력을 뽑는다. OTT/STB 개발을 위주로 했는데 경쟁력(어필 할만한 것)이 무엇이 있나요?

A. Android TV 프로젝트를 중심으로 하였으나 UX/UI 적인 부분을 제외하고 나머지 Android 자체 구조적인 부분은 동일하다
B. Android 앱 개발을 약 8 년간 계속 해왔고 어떠한 UI 앱이든 서비스앱이든 초기 개발부터 끝까지 제작이 가능하다

211. 안드로이드 어플리케이션 클래스에 대해 설명하세요

A. Activity 간 어떠한 클래스 공유
B. Application 객체 멤버는 프로세스 어디에서나 참조 가능
C. 공통으로 전역 변수를 사용하고 싶을 때 Application 클래스를 상속받아 사용 가능
D. onCreate 는 액티비티나 서비스보다 항상 먼저 호출됨
E. Custom Application 클래스를 등록하려면 AndroidManifest.xml 에서 name 속성으로 등록을 해야 한다

212. Context 에 대해 설명하세요

A. Application / Activity 에 관련된 정보를 얻을 수 있다. (리소스, 데이터베이스, Preferences 등에 대한 접근을 제공)
B. Application / Activity 의 부모 클래스
C. 앱에 관하여 시스템이 관리하고 있는 정보에 접근하고 안드로이드 시스템 서비스에서 제공하는 API 를 호출 할 수 있는 기능을 수행하는 Abstract 클래스 (getPackageName, getResource / startActivity, bindService 등의 함수들이 대표적인 함수)
D. Application Context

i. 싱글턴 인스턴스, getApplicationContext()를 통해 접근
ii. App 자체 라이프 사이클과 연결되어 있다
iii. 액티비티에서 라이브러리를 초기화해야 하는 경우, Application Context 를 전달해야 한다
iv. 오래 유지되는 Context 가 필요할 때에만 getApplicationContext() 사용

E. Activity Context
i. Activity 마다 존재
ii. 액티비티의 라이프 사이클과 연결되어 있다
iii. GUI 와 관련된 것은 액티비티 Context 를 사용해야 한다 (Dialog, Toast 등 UI 작업에 필요한 Context 는 Activity Context 사용)

213. 액티비티 라이프 사이클 호출 과정 구체적으로 설명하여라

A. 액티비티
i. **투명 액티비티**가 위에 표시될 때
1. 액티비티(1) ➔ onCreate(1) ➔ onStart(1) ➔ onResume(1)
➔ 투명 액티비티(2) Start ➔ onPause(1)
2. 투명 액티비티(2) ➔ onCreate(2) ➔ onStart(2) ➔ onResume(2)
3. 투명 액티비티(2) 종료 ➔ onPause(2) ➔ onResume(1) ➔ onStop(1) ➔ onDestroy(1)

ii. **노말 액티비티**가 위에 표시될 때
1. 액티비티(1) ➔ onCreate(1) ➔ onStart(1) ➔ onResume(1) ➔ 액티비티(2) Start ➔ onPause(1) ➔ onCreate(2) ➔ onStart(2) ➔ onResume(2) ➔ onStop(1)
2. 액티비티(2) 종료 ➔ onPause(2) ➔ onRestart(1) ➔ onStart(1) ➔ onResume(1) ➔ onStop(2) ➔ onDestroy(2)

B. 참고
i. 서비스
1. StartService
A. onCreate ➔ onStartCommand ➔ onDestroy
2. BindService
A. onCreate ➔ onBind ➔ onUnbind ➔ onDestroy

ii. 프래그먼트

1. onAttach ➔ onCreate ➔ onCreateView ➔ onActivityCreated ➔ onStart ➔ onResume ➔ onPause ➔ onStop ➔ onDestroyView ➔ onDestroy ➔ onDetach

iii. 어플리케이션

1. onCreate ➔ onTerminate

214. Dagger 사용해 봤나요? DI 개념에 대해서 설명해보세요 (Dagger)

A. 개념 숙지한 상태이다

B. Dagger 는 DI framework, Dependency Injection

C. 사용 이유

i. 코드의 재사용

ii. 리팩토링 용이

iii. 테스트 용이

iv. 보일러플레이트 코드 감소

D. 5 가지 필수 개념

i. Inject ➔ 객체 주입

ii. Component ➔ 연결자

iii. SubComponent

iv. Module ➔ 공급자

v. Scope

215. 자바의 추상 클래스, 인터페이스 간의 차이점에 대해서 설명해라

A. 추상클래스

i. Abstract 키워드 사용

ii. Extends 키워드 이용하여 자식 클래스에서 상속받아 구현

iii. New 를 통해 객체를 직접 생성할 수 없다

iv. Abstract 로 선언한 메소드를 자식 클래스에서 반드시 구현해야 한다 (오버라이딩)

v. 관련성이 높은 클래스 간에 코드 공유 하고 싶은 경우 사용

vi. Public 이외의 접근자 선언이 필요한 경우

B. 인터페이스

i. Interface 키워드 사용

ii. Implements 키워드 이용하여 자식 클래스에서 구현

iii. 일반 메소드 / 멤버 변수를 가질 수 없음

iv. 모든 멤버 변수는 public static final, 생략 가능
v. 모든 메소드는 public abstract, 생략 가능 (JDK1.8 부터 static, default 메소드 사용 가능)
vi. 인터페이스는 인터페이스로부터만 상속 가능, 다중 상속도 가능
vii. 서로 관련성 없는 클래스들이 인터페이스를 구현하게 되는 경우 사용(예 ➔ Comparable, Cloneable interface)
viii. 다중 상속 허용하고자 하는 경우 사용

216. 4 대 컴포넌트를 나열하고 각각의 컴포넌트 들에 대해서 설명해라

A. 액티비티

i. 사용자가 앱과 상호작용하는 단일화면 의미
ii. 사용자와 상호작용을 담당하는 인터페이스
iii. 두 개 이상의 액티비티를 동시에 Display 불가능
iv. 1 개 이상의 View / ViewGroup 을 포함

B. 서비스

i. 백그라운드에서 어떠한 작업을 처리
ii. 메인 스레드(UI 스레드)에서 동작하므로 서비스 내에서 별도의 스레드를 생성하여 작업을 처리해야 함
iii. 별도의 UI 를 가지지 않음

C. 브로드캐스트 리시버 (방송 수신자)

i. 안드로이드 OS 로부터 발생하는 각종 이벤트와 정보를 받아와 핸들링
ii. 디바이스의 특수한 상황에 대응하기 위해 사용
iii. 특정한 상황을 제외하고는 시스템에서 시작됨

D. 컨텐트 프로바이더 (콘텐트 제공자)

i. 데이터를 관리하고 다른 앱의 데이터를 제공하는 데 사용되는 컴포넌트
ii. 앱 간의 데이터 공유를 위해 표준화된 인터페이스를 제공
iii. 앱과 앱 저장소 사이에서 데이터 접근을 쉽게 하도록 관리해주는 클래스
iv. 데이터를 캡슐화하여 단일 ContentResolver 인터페이스를 통해 앱에 제공
v. 여러 응용 프로그램 간에 데이터를 공유해야하는 경우에만 필요
vi. 6 개 메소드
    1. onCreate
    2. insert
    3. query
    4. update

5. delete
6. getType

217. 커스텀뷰와 프래그먼트 차이점은 무엇인가?

A. 예를 들어 이미 스마트폰용으로 출시된 앱이 태블릿까지 지원하게 하려면 다른 속성의 화면이 두 개가 필요한데 이런 구현을 쉽게 도와주는 것이 Fragment 이다.
B. CustomView 를 제작해서 거기에 로직을 담으면 요구사항 충족은 되겠으나 각 View 들을 제한된 공간, Activity 안에 보였다 안보였다 하는 식으로 관리를 해주려면 구현적으로 피곤하다. ➔ 이러한 일들을 FragmentManager 가 해준다

218. View.GONE 과 View.INVISIBLE 의 차이

A. GONE ➔ 보이지도 않고 영역도 차지하지 않음
B. INVISIBLE ➔ 보이지는 않지만 영역은 차지하고 있음

219. 프래그먼트 사이의 커뮤니케이션은 어떤 방식으로 하는가?

A. 프래그먼트끼리의 통신은 반드시 호스트 액티비티(프래그먼트들이 바인딩 되어 있는 액티비티)를 거쳐서 이루어져야 한다.
B. 프래그먼트에 콜백함수 정의 후 액티비티에 해당 콜백 함수를 구현함으로써 프래그먼트 사이 통신이 가능하다

220. SurfaceView 란 무엇인가?

A. View 를 상속받는 클래스
B. 일반 View 는 onDraw 메소드를 시스템에서 자동으로 호출해줌으로써 화면을 그림
C. SurfaceView 는 스레드를 이용해 강제로 화면에 그림
D. 원하는 시점에 바로 화면에 그리는 것이 가능 (시스템에 독립적)
E. 더블버퍼링 기능 (다음 장 화면을 미리 그려두고 그리기가 끝나면 바로 전환)

221. RelativeLayout, ConstraintLayout, LinearLayout, Framelayout 에 대해 설명하시오

A. Linear
   i. 뷰가 가로 / 세로로 순차적으로 나열

ii. Orientation 속성으로 가로 또는 세로 설정 가능
iii. 기본값은 가로

B. Relative
i. 따로 위치 지정 안하면 뷰가 0,0 위치에 계속 쌓임
ii. 최상위 부모 레이아웃 상대위치 또는 다른 뷰의 아이디를 참조해 상대적인 위치 지정이 가능

C. Constraint
i. 형제 View 들과의 관계 정의, RelativeLayout 과 유사
ii. RelativeLayout 에서 불가능했던 형제 자식 뷰 간의 상호관계 정의 가능
iii. LinearLayout 을 써야만 했던 뷰 비율 조정도 가능
iv. Chain 사용으로 다른 레이아웃 없이 요소들의 그룹화 가능

D. FrameLayout
i. 여러 레이아웃, 위젯을 하나의 레이아웃(FrameLayout) 안에, 겹쳐서 사용할 수 있다
ii. 위에서 부터 아래로 코딩되었을 때, 아래 코딩된 레이아웃, 위젯이 위에 보여진다.

222. 따로 업무 외 토이 프로젝트 진행하고 있는 것이 있나요?

A. 개인적으로 정리한 문서나 자주 쓰이는 템플릿 코드들은 GitHub 에 올리고 있다.
B. 코틀린 활용한 모바일 앱을 구현하여 GitHub 에 업로드 중이다

223. 코틀린은 사용하는가?

A. 일부 앱들 코드들의 코틀린화 및 신규 앱 개발 시에는 코틀린을 사용한다.

224. String Pool 스트링풀에 대해서 설명해보아라

A. 사전지식 (스트링은 두가지 생성 방식 존재)
i. New 연산자 ➔ Heap 영역에 존재
ii. 리터럴 이용 ➔ String Constant Pool 이라는 영역에 존재

B. 속도적인 면을 향상시키고 싶다면 equals 대신 intern() 함수를 사용
C. Java 6 까지는 Perm 영역 / Java 7 부터는 Heap 영역으로 변경됨 (Perm 영역은 고정된 사이즈, Runtime 에 사이즈 확장 안됨, OutOfMemoryException 발생 가능성)

225. 액티비티에서 홈키 누르는 것과 백키 누르는 것의 동작 상 차이는?

A. 홈키 ➔ onUserLeaveHint ➔ onPause ➔ onStop

i. 홈키 시 Activity 종료시키는 방법

1. AndroidManifest.xml 에서 launchMode = "singleTask", clearTaskOnLaunch="true"를 액티비티 태그에 삽입

226. 코틀린에서 Var 와 Val 의 차이 설명해보아라

A. Var

i. 초기화 후 값을 변경이 가능하다

B. Val

i. 자바에서 Final 과 같다

ii. 초기에 값을 할당되면 나중에 값을 변경할 수 없으며 값을 변경하게 되면 컴파일 에러가 발생

227. 코틀린으로 어느 정도까지 작업이 가능하나요?

A. 자바와 마찬가지로 앱 개발 작업에 코틀린 사용 가능하다.

i. 클라이언트 단 서버 통신, 레이아웃 구성, 기타 비즈니스 로직 구현 등 가능하다

228. 엘비스 연산자에 대해 설명해보세요

A. Null 대신 사용할 디폴트 값을 지정할 때 사용하는 연산자

B. Val value : String = s ?: "" ← s 가 Null 이면 결과는 "", 그렇지 않으면 s 를 변수 value 에 저장한다

229. IPTV 서비스 구현 건에 대해 자세히 설명해주세요

A. 커뮤니케이션 측면

i. 유럽 텔레콤 사인 Orange 사로 출장, 현지 외국인 개발자들과 협업

ii. IPTV 데이터는 Orange 서버팀과, Launcher 는 다른 현지 개발자와, 저는 IPTV Data 처리를 위한 서비스 구현을 담당해서 서로 커뮤니케이션

B. 기술적인 측면

i. TV Contents 를 표시하는 TvView 와 관련

ii. Launcher, LiveTv 또는 기타 앱에서 TvView 를 통해 내려오는 방송 이벤트를 처리하는 서비스

iii. IPTV Content 를 Tune 하면 IPTV 서비스로 해당 정보가 내려오게 되고 Database 에 저장된 IPTV Data 정보를 읽어들여 Tune 을 시도하게 된다. TV Contents 에는 일반 Cable TV, IPTV 등으로 분류가 되고 Vendor 사에서는 이 두가지 Custom 한 InputService 들을 Android TvInputService 를 상속받아 추가 구현을 해줘야 한다

iv. Orange Server 로부터 전달받은 IPTV 데이터를 Database 에 저장/갱신

v. Audio Track / Subtitle 정보 처리

230. 이력서 상 작성해 놓은 앱 포팅 이란게 무엇인지 설명해달라

A. AOSP 소스 상에서 빌드를 하면 이미지가 만들어 지는데 개발한 앱을 빌드 리스트에 추가하거나 3rd Party 앱 / 필요한 라이브러리 등을 추가 / 빌드 스크립트를 작성하는 작업이다

231. 본인의 단점은 무엇인가요?

A. 앱 개발 외에 다양한 업무를 바쁘게 하는 편이라 스트레스 관리가 필요

B. 체력 관리, 업무에 집중하다보면 체력이 고갈된 줄도 모른체 하다가 번아웃되는 경우가 있다.

C. 스트레스 관리를 위해 운동, 취미 생활로 최대한 풀려고 노력 중이다

232. 3rd Party 업체에 라이브러리 제공했다 라고 적혀져 있는데 구체적으로 설명해주세요

A. 프로그램 편성표 앱

i. DB 접근

ii. 스크린 사이즈 변경

iii. 메타 데이터 얻기

B. Launcher 앱

i. Player 관련 API 제공

ii. IPTV 메타 데이터 얻기 / 서버로부터 받은 세션 정보

233. 일을 할 때 주로 어떤 피드백을 받았나요? 인사고과는 어떠셨나요?

A. 근면, 성실, 능동적

B. 앱 개발자인데도 불구하고 앱 개발 외에도 빌드 서버 환경 구축, Git, Jenkins, 안드로이드 시스템에 대해서도 익숙하다는 평을 받았다

234. 자신의 5 년후(10 년 후)의 모습은 어떠할 것 같나요?

A. 5 년 동안 안드로이드 경험을 바탕으로 모바일 개발팀에서 기술 매니저가 되어 있을 것 같다

B. 이 회사에서 프로젝트 자체를 이끄는 리더가 되어 팀원 관리도 겸하면서 프로젝트를 성공적으로 이끌어 나가고 싶다.

235. 프로젝트 매니저, 리딩을 위해 준비하는 것이 있나요?

A. 인력관리, PMP, 애자일 방법론 적용

B. Mobile 개발 / Android TV 개발의 초기 단계부터 릴리즈까지 단계를 잘 알고 있다.

C. 중간중간 과정에서 무엇이 필요하며 어떠한 식으로 대처하는 지에 대해서 잘 알고 있으므로 매니저/리딩 업무를 할 수 있다.

236. 현재 받고 있는 연봉에 대해 어떻게 생각하시나요?

A. 만족한다.

B. 자신의 업무적 가치가 높아진다면 자연스럽게 연봉도 오를거라고 생각한다. 가치를 높이기 위해 계속 공부하고 업무에 충실하며 노력할 것이다

237. Git, SVN 차이점에 대해 말해보세요

A. SVN
   i. 직관적
   ii. 모든 사람이 중앙 서버에 있는 같은 자료를 받아온다
   iii. Commit 을 하는 순간 모든 사람에게 공유 (두 사람이 하나의 파일을 동시에 수정하고 커밋하였을 때 충돌이 일어날 확률이 높다)
   iv. 중앙 저장소에 에러가 생기면 모든 작업이 마비

B. Git
   i. 러닝 커브 큼
   ii. 작업 내용을 Commit 하여 로컬저장소에 반영한 후 원격 저장소에서 Fetch 로 로컬 저장소로 마스터 파일을 Merge 한 뒤 로컬 저장소의 내용을 Push
   iii. 모든 작업이 로컬에서 이루어지고 네트워크 사용은 원격 저장소로 저장할 때 한번 이루어지므로 개발 시 처리 속도가 빠름
   iv. 원격 저장소의 내용이 모든 협업자들의 로컬 저장소에 저장이 되므로 중앙 저장소에 에러가 생겨도 복구가 용이
   v. 히스토리 관리 용이

238. 소스코드 리팩토링 경험이 있나요?

A. Launcher 앱에서 클래스 간의 결합도가 높아서 서로 분리하는 작업을 했다

239. Unit Test 해본 적 있나요?

A. Unit Test 에 대해 알고 있다.

B. 코드의 유닛단위 (메소드, 클래스, 컴포넌트)의 기능을 실행하는 방식

C. 관련 툴

i. Junit / Mockito / PowerMock

240. Ui Test 해본 적 있나요?

A. 사용자 인터랙션(버튼 클릭, 텍스트 입력 등)을 평가

B. 관련 툴

i. Espresso / UIAutomator / Robotium / Appium / Calabash / Roboletric

241. Clean Code / Clean Architecture 설계 경험이 있나요?

A. Clean Architecture

i. Boundary 를 나누는 기술

B. Clean Code

i. 모든 테스트 실행

1. 테스트 케이스를 만들려면 결합도가 높으면 안된다
2. 한 클래스의 단일 책임 원칙을 지킬 수 있다
3. 다른 클래스와 결합도가 높으면 테스트가 어려워지므로 자연스럽게 의존성 주입을 사용하게 된다

A. 결론 ➔ 낮은 결합도, 높은 응집력이라는 객체 지향 방법론을 자연스럽게 달성

ii. 리팩토링 (중복 제거)

iii. 표현 방법

1. 좋은 함수 이름
2. 함수, 클래스 크기 줄이기
3. 표준 명칭 사용 (패턴 적용한 경우, COMMAND, VISITOR 를 클래스 이름 뒤에 붙인다)
4. 단위 테스트 코드 작성

242. 코드 품질 / 성능을 위한 노력을 한 경험이 있나요?

A. 같은 클래스 내에서 Getter, Setter 사용 지양
B. 필드 참조들을 캐시
C. 상수는 Final 키워드로 선언
D. 반복문은 For-Each 문을 사용
E. 열거형 피하기

243. Git Change ID 가 무엇인지 설명해보아라

A. Gerrit 이 변경사항을 구분하는 고유 식별값
B. Commit-msg 훅 스크립트가 설치되어 있으면 프로젝트 Commit 했을 때 자동으로 Change-id 를 생성한다

244. onStart() 와 onResume() 함수가 구분되어 있는데 둘의 차이점은 무엇인가요?

A. onResume ➔ Focus 를 다시 얻었을 때 호출됨
B. onStart ➔ Activity 화면이 다시 돌아 올 때 호출됨

245. Custom View 를 사용해 본 적이 있는지?

A. 있다. 원하는 스타일의 아이템 리스트를 만들기 위해 만들어 본 적이 있다.

246. 메인(UI)스레드에서 네트워크 통신을 하면 어떤 일이 생길까요?

A. AndroidRuntime 에러 발생 (Network IO 를 Main 에서 수행해서)

247. ListView 에 ViewHolder 를 넣을 수 있을까요?

A. 가능하다.
B. 스크롤 시 안 보이던 아이템은 매번 getView 호출됨
C. ViewHolder 로 한번 생성된 View 는 findViewById 재호출을 막는다

248. Activity 가 화면 가로세로 전환 또는 메모리 부족 등으로 종료되었을 때 데이터를 유지하는 방법이 무엇인가요?

A. onSaveInstanceState 함수 활용한다

249. BLE 개발 사례에 대해 자세히 소개해주세요

A. BLE Gatt Data 를 활용해서 RCU Firmware Upgrade Service, Force Touch RCU Service 를 만든 경험이 있다.

B. BLE Gatt 로 들어온 Custom Raw Data 를 가공/처리하여 안드로이드 키 이벤트를 생성했다

250. 최근 다녀온 개발자 컨퍼런스에서 기억에 남는 기술을 소개해주세요.

A. 최근 다녀온 개발자 컨퍼런스가 없고 과거 자바 개발자 컨퍼런스에 참석한 경험은 있다. 오래되어서 기억이 나질 않는다

251. 안드로이드는 다른 플랫폼에 비해 어떤 장점이 있는가?

A. 오픈소스 프로젝트

B. 자바 언어 사용

C. 스마트폰/기타 플랫폼을 위한 완벽한 컴포넌트 제공

252. 안드로이드 프로젝트 구성요소에 대해서 설명하시오.

A. App ➔ Manifest, 소스코드, 리소스(레이아웃, 아이콘, 그림 등)

B. Gradle Scripts ➔ build.gradle

253. 코딩하면서 좋았던 경험은 무엇인가?

A. Vietnam 개발자와 협업하여 Feature 구현 건이 완료되었을 때

B. 출장 업무, 외국 개발자와 Communication 하여 IPTV 서비스 개발 완료하였을 때

C. Radio 앱 혼자서 처음부터 릴리즈까지 완료하고 최종 품질 이슈 0 였을 때

254. 코딩하면서 힘들었던 일이 있었는가?

A. 인수인계가 제대로 되지 않은 프로젝트 이슈를 수정할 때

255. 10 년뒤에 어떠한 모습이 될 것 같은가?

A. 10 년 뒤에는 팀장이 되어 있을 것 같다

B. 개발 팀을 이끌면서 프로젝트를 성공적으로 수행해 나가고 싶다

C. 개발 관련해서 원활한 커뮤니케이션이 가능하도록 계속적으로 노력

256. Zygote 에 대해서 말해보세요.

A. 자바로 작성되는 안드로이드 앱의 실행 속도를 빠르게 하기 위해서
   i. 앱이 실행되기 전에 가상 머신의 코드 / 메모리 정보를 공유함으로서 앱 실행을 단축
   ii. 안드로이드 프레임워크에서 필요로 하는 클래스와 자원을 미리 메모리에 로딩, 연결 정보를 구성

257. JVM 과 Dalvik 가상 머신에 대해서 설명해보세요

A. 공통 ➔ 자바의 경우 한가지 CPU 의 아키텍쳐나 환경에 맞추는 것이 아닌 **바이트코드**라는 것으로 컴파일되며 이를 실행하기 위해 자바 가상 머신이 필요 (자바는 바이트코드만으로 여러가지 아키텍쳐나 플랫폼에서 작동할 수 있도록 하는 것이 목표)

B. **JVM** (**Java** Virtual Machine)
   i. 자바 바이트 코드를 실행할 수 있는 주체
   (자바 바이트 코드는 플랫폼에 독립적, JVM 에 의존적으로 실행된다)
   ii. 스택 기반의 가상 머신
   iii. 포인터를 지원하되 C 와 같이 주소 값을 임의로 조작이 가능한 포인터 연산이 불가능
   iv. 가비지 컬렉션(GC) 사용

C. **DVM** (**Dalvik** Virtual Machine)
   i. 32 비트만 지원
   ii. 실행할 때마다 컴파일
   iii. CPU, Memory 사용 높음
   iv. 배터리 소모가 크다 (실행 시 마다 컴파일, 하드웨어 전체적으로 상당한 부하)
   v. JVM 을 사용할 수 있지만 라이선스 문제로 **구글에서 Dalvik VM 을 따로 개발하여 안드로이드에 탑재** (JVM 은 무료이지만 GPL 라이센스)
   vi. 자바 코드를 최적화시켜 소형 기기에서도 잘 작동할 수 있도록 해줌
   vii. 자바 바이트 코드를 변환해서 확장자가 **dex**(Dalvik Executable)인 바이트 코드를 생성한다. (**Java ➔ Class ➔ Dex** (**여러 개의 Class 파일이 하나의 Dex 파일로 변경**된) ➔ **APK** (**Resource + Dex**)
   viii. 안드로이드 앱을 실행할 수 있는 가상 머신

ix. 모바일 기기 환경에 맞게 최적화 되어 나온 가상머신 (배터리 수명, 컴퓨팅 파워 / 메모리가 데스크탑 환경에 비해 열악하므로)
x. 스택 기반 모델이 아닌 레지스터 기반 모델 (적은 메모리 사용 요구, 빠르게 동작)
xi. **JIT** (Just In Time) 컴파일러 사용 (Android 2.2(Froyo) 버전부터 적용된 DVM 내부 컴파일러)
xii. 앱 설치 공간이 적게 든다
xiii. CMD 알고리즘 사용
xiv. Dex, ODEX 실행
xv. 여러 개의 VM 인스턴스가 한번에 실행 가능

D. **ART** (**Android** RunTime)
i. 32, **64** 비트 지원
ii. 안드로이드 KitKat 버전부터 생김
iii. 안드로이드 **Lolipop** 버전 이후는 **AOT 컴파일러가 기본으로 적용**
iv. 안드로이드 **Nougat** 버전부터 **JIT 와 AOT 를 모두 탑재**, 최초 설치시에는 무조건 JIT 사용하여 설치시간, 용량 적게 소모한 뒤, 기기 사용하지 않을 때나 충전 중일 경우 컴파일을 조금씩 수행하여, 자주 사용되는 앱을 AOT 방식으로 전환하는 것으로 바뀜.
v. CPU, Memory 사용 낮음
vi. 안드로이드 앱 런타임 환경으로 새로운 디버깅 기능과 고수준의 앱 프로파일링 기능 제공
vii. 퍼모먼스 개선
viii. **AOT**(Ahead On Time) 는 ART 내부 컴파일러, **앱이 설치되는 시점에 앱 전체 바이트 코드를 기계어로 번역한다**
ix. **앱 설치 시간이 오래 걸리나 런타임에서 바이트 코드를 해석하는 시간을 제거했기 때문에 퍼포먼스가 좋고 배터리 수명이 향상**된다
x. 앱 설치 공간이 많이 필요
xi. Customized CMD 알고리즘 (기존보다 2 배 정도 빠름)
xii. OAT (ART) 실행

258. SharedPreferences 에서 commit() 과 apply() 의 다른점에 대해 말해주세요.

A. Commit
   i. 성공하면 Boolean 타입인 true 값 반환
   ii. 동기 함수

B. Apply
   i. 반환값 없음
   ii. 비동기 함수
   iii. Apply 사용 시 저장 속도가 약 10~30 배 까지 빨라진다

259. 안드로이드에서 메모리 누수를 줄일 수 있는 방법에 대해 말해주세요.

A. 메모리 누수 발생 원인
   i. 정적 뷰에 Activity 를 참조
   ii. 작업 스레드에 대한 Activity 누수
   iii. 스레드 자체 누수

B. 메모리 누수 줄이는 방법
   i. onDestroy 에서 정적 변수 참조를 끊어야 한다 (Null 로 초기화)
   ii. Inner Class 를 만들 때 가능한 한 정적으로 생성한다

260. 안드로이드 APK 파일의 크기를 줄일 수 있는 방법들을 말해주세요.

A. 사용하지 않는 리소스 삭제
B. 불필요한 종속성 제거
C. 여러 개의 APK 를 화면 밀도에 맞춰 생성
   i. Density 지정
D. 특정 ABI(Application Binary Interface)를 지원하는 하나의 APK 생성
   i. ARM, API, MIPS, NVS64
E. 사용하지 않는 선택적 리소스 삭제
   i. resConfig 'en', 'th' ← resConfig 사용하여 필요한 언어만 포함
F. 리소스 축소 사용
   i. minifyEnabled (소스코드를 난독화해서 보안을 강화하거나 앱의 크기를 줄이고 싶으면 true 로 설정)
   ii. shrinkResources 속성을 true 로 하게 되면 minify 작업 후 사용하지 않는 리소스를 삭제
   iii. 쉐이프 드로어블 (Shape Drawable) 사용
      1. 비트맵보다 용량이 작으면서 XML 로 작성 가능

iv. WebP 사용

v. VectorDrawable 사용

261. 안드로이드 어플리케이션의 빌드 시간을 줄일 수 있는 방법들을 말해주세요.

A. 최신 Gradle 플러그인 사용

B. 레거시 Multidex 사용 자제

C. 개발 시 여러 개의 APK 생성 설정 비활성화

i. Splits.abi.enable=false / splits.density.enable=false

D. 개발 빌드에서 패키징 리소스 최소화

i. resConfigs('en', 'xhdpi')

E. PNG 크런칭 비활성화

i. AAPT(Android Asset Packaging Tool)는 기본적으로 PNG 를 Crunch 함. APK 를 릴리즈할 때는 좋지만 작업에 시간이 소요됨

1. aaptOptions.cruncherEnabled=false

F. Instant Run 사용

i. Run 은 코드 스와핑 일어나면서 앱이 재시작됨

ii. 시스템은 라이브 프로세스로 즉시 변경사항을 푸쉬, 재시작 필요 없음

G. Crashlytics 의 Build ID 업데이트 비활성화

i. Crashlytics (오류 보고 솔루션)

ii. 각 빌드 별로 고유한 Build ID 를 생성

iii. False 로 설정하면 시간 단축 가능

1. Ext.alwaysUpdateBuildId = false

H. 종속성 동적 버전 사용 자제

i. 참고

1. Gradle 은 맨 뒤에 '+'를 추가하면 자동으로 최신 버전으로 업데이트

A. Implementation 'com.xxx:testcompat-v1:+'

I. Gradle.properties 설정

J. R8 사용

262. 바이트 코드를 안드로이드에서 바로 실행할 수 있나요?

A. 바로 실행 불가능

B. 실행하려면 Dalvik VM 필요

263. 채팅 기능 구현 경험이 있는가?

A. 과거 대학 학부 시절 Java 를 이용하여 제작해 본 경험이 있다 (AWT, Swing, SWT 사용)

B. 자바 소켓 프로그래밍을 활용하여 개발하였다

264. Annotation 이란?

A. 메타데이터(Metadata)

i. 앱이 처리하는 데이터가 아님, **컴파일, 런타임 과정에서 코드를 어떻게 컴파일하고 처리할 것인지를 알려주는 정보**

B. 세가지 용도

i. 컴파일러에게 코드 문법 에러를 체크하도록 정보를 제공

1. @Override

ii. 소프트웨어 개발 툴이 빌드나 배치 시 코드를 자동으로 생성할 수 있도록 정보를 제공

iii. 실행 시(런타임 시) 특정 기능을 실행하도록 정보를 제공

265. AsyncTask Deprecated 된 이유는 무엇인가?

A. Context Leak 이나 콜백을 빼먹거나 설정 변화의 충돌로 많은 이슈 야기

B. AsyncTask 가 동작 중일 때 UI 의 변화, 즉, **완료된 시점에 UI 가 존재하지 않을 때에 대한 문제가 있음 (NPE 발생 등)**

i. 항상 UI 처리에 리스너 장착

ii. onPostExecute 에서 항상 View 를 체크

iii. AsyncTask 를 Customizing 해서 View 관리

266. JAR, AAR, DEX, APK 에 대해 설명해보아라?

A. **JAR** (Java Archive)

i. Java 응용 프로그램을 배포하기 위해 고안된 패키지 파일 형식

ii. 컴파일된 Java Class 파일, 매니페스트와 같은 파일들이 포함

iii. 기본적으로 ZIP 압축 형태

B. **AAR** (Android Archive)

i. Android 라이브러리 프로젝트의 바이너리 배포판

ii. 클래스 파일 + **리소스** 파일들도 포함

C. **DEX** (Dalvik Excutable)

i. DVM 을 위한 실행 파일

ii. JVM 을 위한 .class 파일과 같은 역할
iii. Android SDK 의 Dex 컴파일러에 의해 JVM 바이트코드를 DVM 바이트코드로 변환하고 모든 클래스 파일들을 Dex 파일에 넣습니다
iv. 바이너리 파일 형식으로 컴파일 된다

D. **APK** (Android Application Package)
i. Android 플랫폼에 배포할 수 있도록 설계된 파일 형식
ii. 컴파일된 클래스를 Dex 파일 형태로 포함시키고 AndroidManifext.xml 등 리소스 파일들도 포함

267. 프래그먼트는 기본 생성자를 왜 사용해야 할까?

A. 프래그먼트 소스 코드 중 instantiate 를 보면 새로운 인스턴스 생성 시 인자가 없는 생성자를 사용하여 초기화한다
B. 생성 시 파라미터를 전달하고 싶으면 Bundle 에 담아 **setArgument** 함수를 호출하는 것이 일반적이다

268. Gradle / Ant / Maven 이 무엇인가?

A. Gradle
i. Groovy 기반으로 한 빌드 도구
ii. Ant 와 Maven 같은 이전 세대 빌드 도구의 단점을 보완, 장점 취합하여 만든 오픈소스로 공개된 빌드 도구
iii. XML 이 아닌 JVM 에서 동작하는 Groovy 기반의 DSL(Domain Specific Language)를 사용
iv. Groovy 는 자바 문법과 유사, 자바 개발자에 한해서 러닝커브 낮음
v. Gradle wrapper 를 시용하면 Gradle 이 설치되지 않은 시스템에서도 프로젝트 빌드 가능

B. Ant
i. XML 기반으로 빌드 스크립트를 작성
ii. 자유로운 빌드 단위 지정 가능
iii. 간단, 사용 용이
iv. 유연하나 프로젝트 방대해지는 경우 스크립트 관리나 빌드 과정이 복잡
v. 생명주기를 갖지 않아 각각의 결과물에 대한 의존관계 등을 정의해야 한다

C. Maven
i. XML 기반으로 작성
ii. 생명주기, 프로젝트 객체 모델 (POM, Project Object Model) 개념 도입

iii. 빌드 스크립트 개선
iv. Pom.xml 에 필요한 라이브러리 선언하면 자동으로 해당 프로젝트로 불러옴
v. 러닝 커브 높음
vi. 라이브러리가 서로 의존하는 경우 복잡해질 수 있다

269. String vs StringBuffer vs StringBuilder 에 대해 설명해보아라?

A. String

i. 불변 (Immutable)의 속성
ii. 변하지 않는 문자열을 자주 읽어들이는 경우 String 사용 적합, 좋은 성능
iii. 문자열 추가, 수정, 삭제 등의 연산이 빈번하게 발생하는 곳에 String 클래스를 사용하면 Heap 에 많은 임시 Garbage 가 생성 ➔ Heap 메모리 부족, 성능에 악영향
iv. 새로운 문자열 참조 시 값 자체가 변경되는 것이 아니라 주소가 변경됨
    1. 새로운 메모리 영역을 가리키게 변경
    2. 사용하지 않는 메모리 영역은 Garbage 로 남아 있다가 GC 에 의해 소멸
    3. 문자열을 수정하는 시점에 새로운 String 인스턴스가 생성됨
v. 불변성을 가지므로 멀티쓰레드 환경에서 안전 (Thread-Safe)

B. StringBuffer

i. 가변(Mutable) 속성
ii. Append, delete 등의 API 사용하여 동일 객체 내에서 값 변경 가능
iii. 문자열 추가, 수정, 삭제 빈번한 경우 사용
iv. 동기화 키워드 지원, 멀티쓰레드 환경에서 안전 (Thread-Safe)

C. StringBuilder

i. 가변(Mutable) 속성
ii. Append, delete 등의 API 사용하여 동일 객체 내에서 값 변경 가능
iii. 문자열 추가, 수정, 삭제 빈번한 경우 사용
iv. 동기화 지원하지 않음, 단일 쓰레드에서의 성능은 StringBuffer 보다 뛰어남

270. 직렬화 vs 역직렬화 개념에 대해 설명해보아라?

A. 직렬화 (Serialization)

i. 객체를 직렬화하여 전송 가능한 형태로 만드는 것 (객체 ➔ 바이트 스트림)
ii. 객체들의 데이터를 **연속적인 데이터로 변형**하여 Stream 을 통해 데이터를 읽도록 해준다

B. 역직렬화 (Deserialization)
   i. 직렬화된 파일 등을 역으로 직렬화하여 **다시 객체의 형태로 만드는 것** (바이트 스트림 ➔ 객체)
   ii. 저장된 파일을 읽거나 전송된 스트림 데이터를 읽어 원래 객체의 형태로 복원

271. Parcelable, Serializable 차이점은 무엇입니까?

A. Parcelable
   i. 자바가 아닌 안드로이드 SDK 내에 포함
   ii. 자바 Reflection 사용하지 않음 (대신 보일러플레이트 코드 생김)
   iii. IPC 를 이용하기 때문에 속도 빠름 (커널 메모리를 통해 데이터를 다른 프로세스로 전달하는 통로를 만들어 줌)

B. Serializable
   i. 자바 표준 인터페이스, 안드로이드 SDK 에 포함되어 있지 않음
   ii. 자바의 Reflection 발생, 많은 Object 생성, GC 발생 ➔ 앱 성능 저하, 배터리 소모 심화
   iii. POJO 클래스에 Serializable 인터페이스 구현
   iv. Parcelable 처럼 writeObject 와 readObject 를 구현하면 Parcelable 보다 더 빠르다는 내용도 있음

272. STB/OTT 개발 하실 때 Board Chipset 은 어떤 것을 사용하였나요?

A. Broadcom / Amlogic / Telechips

273. 칩셋사로부터 AOSP SDK 소스 전달 받고 그 위에서 작업을 보통 하시나요?

A. 그렇다

274. 서버 통신 관련 코딩도 많이 해 보았나요?

A. 그렇다. 안드로이드 클라이언트 앱에서 HTTP 통신 관련 코딩 많이 해봤다

275. TSL, SSL 도 해 보았나요? (Http 통신 관련 질문)

A. 개념 숙지하고 있다

B. **TSL** (Transport Layer Security)
   i. 전송 계층 보안, 가장 최신 기술로 더 강력한 버전의 SSL
   ii. Digicert 와 관련

C. **SSL** (Secure Socket Layers)
   i. 보안 소켓 계층, 웹사이트와 브라우저(또는 두 서버) 사이에 전송된 데이터를 암호화하여 인터넷 연결을 보안을 유지하는 표준 기술

276. 앱 개발 때 어떤 식으로 분담해서 구현하는가?

A. 보통 APK 별로 담당자 할당이 되어서 구현을 하게 되고 규모가 있는 앱의 경우 2~3 명 씩 세부 모듈 별로 맡아서 진행한다

277. 가장 최근에 소스 빌드를 해서 이슈 검토를 해 본 경험이 언제 입니까?

A. 지난주에 이슈가 있어서 개발한 앱을 Full Source 에 적용한 뒤 검토를 했었다.

278. 일 년에 신규 앱은 보통 몇 개 정도 제작합니까?

A. 반기당 4~5 개씩 제작하는 것 같다. 기존에 제작했던 앱과 유사하면 템플릿을 가져와서 구현하지만 대부분 신규 시나리오 적용을 하게 되면 앱을 처음부터 다시 만들어야 했었다

279. 앱 개발 때는 주로 어떻게 개발 하시나요? (어떤 툴, 환경에서)

A. 단순 앱 개발은 Android Studio 툴에서 작업했다
B. 프레임워크 소스 검토 시에는 터미널에서 보거나 Visual Studio Code 를 사용했다
C. AOSP 소스 상에서 빌드 확인도 해야 하므로 Android.mk 파일을 만들어 리눅스 빌드 환경에서도 확인을 했다

280. 프레임 워크, 앱 파트로 구분된 이유가 있나요?

A. 프레임워크 파트에서는 주로 안드로이드 프레임워크와 밀접한 관계가 있는 방송 데이터 서비스를 구축하므로 앱과 프레임워크로 분리되어 있다

281. Hardware Driver 쪽 관련해서 업무를 해본 경험이 있는가?

A. 없다.

282. Media 파트가 있으면 주로 Media 파트와 부딪힐 일이 많았겠네요. Media Player 자체 내부 로직에 대해서도 검토를 해 본 경험이 있나요?

A. 검토해 본 경험은 없으나 앱 개발 시 이슈 검토할 때 커뮤니케이션은 잦다.

283. 현재 회사는 어디에 위치해 있나요?

A. XXX 에 위치해있다.

284. Launcher 의 경우 내부 Framework API 를 많이 사용하는 것으로 알고 있는데 어떻게 구현하셨나요?

A. Android Framework Library 를 직접적으로 사용하진 않고 자바의 Reflection 을 사용하여 내부 함수를 Call 한다.

B. 3rd Party 업체의 라이브러리도 사용한다

285. Framework 소스도 검토 해 보신 적 있나요?

A. 보통 경계선 없이 앱 이슈 검토할 때 Framework 까지 검토한다

286. 코틀린은 많이 사용해 보셨나요?

A. 현재 프로젝트 중 일부를 코틀린으로 변경하고 있는 상태이며 신규 프로젝트는 코틀린으로 진행 중에 있다.

B. 코틀린 사용해서 만든 모바일 앱을 구현 중이고 수정될 때 마다 GitHub 에 개인적으로 업로드 중이다.

287. 현재 진행 중인 프로젝트는 안드로이드 버전 몇을 사용 중이신가요?

A. 현재 진행 중인 일본향 STB 는 **Pie** 버전을 사용하고 있습니다.

288. 실제 개발 진행하여 OTT/STB 에 탑재하는 앱은 몇 개 정도 인가요?

A. 10 개 이상 된다 (Launcher / Wizard / Settings / VOD / File Browser / LiveTv / ETC Service)

289. 모바일 앱 개발을 따로 해 본 적이 있나요?

A. 네. 회사에서 Mobile RCU 앱을 유지보수 해 본 경험이 있고 따로 혼자서도 앱을 개발하고 있다.

290. Database 는 외부 Database 를 사용하나요? 아니면 STB/OTT 내장 DB 를 사용하나요?

A. Android **SQLite** 데이터베이스를 사용한다

291. 안드로이드 바인더에 대해 설명해보세요

A. 독립된 프로세서들을 연결해 주는 역할

B. **IPC (Inter Process Communication)도구,** 다른 프로세스에 있는 함수를 마치 현재 프로세스에 존재하는 함수처럼 사용할 수 있게 해주는 RPC(Remote Procedure Call)를 지원하는 데 주로 이용

C. 사용자 공간에서 접근할 수 없는 공간인 커널 공간을 이용해 테이터를 주고 받기 때문에 IPC 간의 보안 문제도 동시에 해결

D. 리눅스 커널 구조

i. Display Driver
ii. Camera Driver
iii. Bluetooth Driver
iv. Shared Memory Drivers
v. Binder (IPC) Driver
vi. USB Driver
vii. Keypad Driver
viii. WiFi Driver
ix. Audio Drivers
x. Power Management

292. 프로세스 제어 블록, PCB (Process Control Block)

A. 특정한 프로세스를 관리할 필요가 있는 정보를 포함하는 OS 커널의 자료구조

B. PCB 는 운영체제가 프로세스를 표현한 것

C. OS 가 프로세스 스케쥴링을 위해 프로세스에 관한 모든 정보를 가지고 있는 데이터베이스를 PCB

D. 각 프로세스가 생성될 때 마다 고유의 PCB 가 생성되고 프로세스가 완료되면 PCB 는 제거됨

E. PCB 구성 정보

i. 프로세스 식별자
ii. 프로세스 상태 (생성, 준비, 실행, 대기, 완료)
iii. 프로그램 계수기 (Program Counter)
    1. 프로세스가 다음에 실행할 명령어의 주소를 가리킴
iv. CPU 레지스터 / 일반 레지스터
v. CPU 스케쥴링 정보
    1. 우선순위, 최종 실행시각, CPU 점유시간

vi. 메모리 관리 정보 (해당 프로세스 주소 공간)

vii. 프로세스 계정 정보

1. 페이지 테이블, 스케쥴링 큐 포인터, 소유자, 부모

viii. 입출력 상태 정보

1. 프로세스에 할당된 입출력장치 목록, 열린 파일 목록

ix. 포인터

1. 부모프로세스, 자식 프로세스에 대한 포인터, 프로세스가 위치한 메모리 주소에 대한 포인터, 할당된 자원에 대한 포인터

293. 자바 Volatile 키워드에 대해 설명해보아라

A. 자바 코드의 변수를 **메인 메모리에 저장 할 것을 명시**하기 위해 사용된다.

B. 컴퓨터의 메인 메모리로부터 읽히고 쓰기 작업은 메인 메모리로 직접 이루어진다 (CPU 캐쉬가 사용되지 않는다)

294. 자바의 Reflection 에 대해서 설명해라

A. 구체적인 클래스 타입을 알지 못해서 그 클래스의 메소드와 타입 그리고 변수들을 접근할 수 있도록 해주는 자바 API

B. 객체를 통해 클래스의 정보를 분석해 내는 프로그램 기법

295. Anko 란 무엇인지에 대해서 설명해라

A. Kotlin 으로 작성된 DSL (Domain-Specific-Language)

B. 가독성 증대, XML 파싱에 대한 오버헤드 감소

i. UI 생성 시 XML 을 사용하게 되면 Java 코드로 변환되는 작업을 거치면서 CPU 나 배터리 소모를 하게 되고 재활용이 불편하다는 단점이 있다.

296. 1 차 면접에 대한 평이 적극적이다 라는 평이 있었다. 여기서 다시 한번 의지를 보여주실 수 있나요?

A. 안드로이드 앱 개발 시 모듈만 맡은 것이 아닌 전체적인 그림 그리기가 가능하고 초기 설계 단계부터 최종 릴리즈까지 경험이 있기 때문에 앱 개발 업무를 잘할 수 있다.

B. 같이 일하는 동료들과 마찰없이 업무를 수행해왔다.

C. 모르는 분야가 있으면 업무 외 시간을 활용해서라도 반드시 따라잡겠다.

D. 근면, 성실을 항상 기본으로 삼고 업무에 임하겠다.

297. 이직할 때 본인이 생각하기에 중요한 점은 무엇입니까? (연봉/업무/복지/사람/회사 비전 등)

A. 미래에도 할 수 있는 일 인지가 가장 중요하다. 생계와 관련이 있기 때문에.

B. 연봉이 중요하다. 좋은 보수는 더 업무에 집중할 수 있게 만들고 적극적으로 일하게 만든다고 생각한다.

C. 같이 일하는 사람이 중요하다. 협업이 잘 되어야 업무가 잘 되고 더 나아가 회사 차원에서도 좋을 것 같다.

298. 졸업하자마자 바로 취직을 한 것이냐?

A. 그렇다. 졸업 예정자로 취직을 했다.

299. 서로 다른 분야의 팀/사람과 커뮤니케이션 할 때 갈등이 많은데 만약 그러한 갈등이 생긴다면 어떻게 할 것인가?

A. 무조건 서로 간의 대화가 중요하다고 생각한다.

B. 다른 영역의 동료들과 초기 프로젝트 셋업이나 개발 시나리오 등 논의 시 다 함께 참여해서 초기에 이야기를 많이 한다.

300. 금융 쪽 일은 해 본 적이 있냐? 금융 쪽 지식을 어떻게 잘 쌓을 것인가?

A. 없다. 업무 외 시간 활용해서 인터넷/서적 등을 활용해서 최대한 빠른 시일 내에 업무에 지장이 없도록 최선을 다하겠다.

301. 현재 다니고 있는 회사의 업무 강도는 어떠한 가요?

A. 상황마다 다르다.

302. 대학 친구들 중에 금융권 일을 하는 사람이 있나?

A. 그렇다/없다

303. 서버 통신 관련해서도 구현을 해 본 적이 있는가?

A. 그렇다. 고객사 서버에 로그인을 하거나 로그인 이 후 데이터를 서버로부터 가져오거나 하는 코드를 구현해 본 적이 있으며 이러한 구현 업무는 거의 대부분 프로젝트마다 있었다. 그래서 해당 구현 업무에 익숙하다.

304. 안드로이드 Repository 패턴에 대해서 설명해보세요

A. DataSource 를 캡슐화
   i. 도메인과 연관된 모델을 가져오기 위해 필요한 DataSource 가 무엇인지 Presenter 계층에서는 알 필요가 없다. ➔ DataSource 를 새롭게 추가하는 것이 부담이 없음
   ii. DataSource 변경이 일어나도 다른 계층에 영향 없음
   iii. Client 는 Repository 인터페이스에 의존하기 때문에 테스트 용이
   iv. Presenter 계층과 Data 계층의 Coupling 을 느슨하게 해준다.

305. 구현해왔던 앱들에 대해서, STB/OTT 모델에 따라 다 구현이 달랐나요?

A. 전체적인 큰 비즈니스 로직은 유사
B. 고객사가 항상 달랐으며 고객사 시나리오에 따라 구현 내용이 항상 달랐다.

306. 생산 검증용 SW 관련 업무에 대해 말씀해주세요

A. 생산 라인에 있는 비개발자가 쉽게 테스트 결과를 볼 수 있도록 구현한 SW.
B. CPU 온도, Wifi 모듈, Bluetooth 모듈, Video Play, HDMI 해상도 변경 등의 테스트를 앱이 시작함과 동시에 자동으로 진단하고 UI 상에 결과를 표시해준다.
C. 앱이 들어있는 USB 를 삽입하면 자동으로 시료에 설치 / 테스트가 진행되는 방식

307. 본인의 현재 가지고 있는 스킬셋에 대한 장단점에 대해 말해보세요.

A. 자바, 안드로이드 앱 개발을 8 년간 계속 해왔고 앱 개발을 처음부터 릴리즈 단계까지 가능하다는 것은 장점이다.
B. 제품 개발 위주의 개발을 주로 해서 언어적으로 코틀린이나 다른 개발 패러다임과 같은 신기술 사용 경험이 상대적으로 적다는 것은 단점이다.

308. 이력서 상에 부족한 구조 설계로 인해 힘들었었고 이를 극복하기 위해 노력한다고 되어 있는데 현재 구조 설계는 어떻게 하시나요?

A. 새로운 기술을 도입하거나 사용하기 보다는 최대한 비즈니스 로직과 UI 로직을 적절한 인터페이스를 사용하여 분리하는 데에 중점을 둔다
B. 초기 구조 설계 시 기술 매니저와 항상 논의를 진행하고 있고 코드 리뷰를 받는다.

309. 제플린 툴 많이 사용해봤나요?

A. 최근 2 년간 UI 앱 개발 시에는 계속 사용해왔다.

310. C/C++ 로 어플리케이션 구현 할 줄 아는가?

A. 실제로 이 언어들을 사용해서 앱을 개발한 적은 없으나 이슈 발생 시 소스 코드 검토할 정도는 된다.

B. 자바 레이어에서 JNI 사용해서 C/C++ 단 코드를 사용해본 경험이 있다. (Amlogic, Broadcom, Telechips 와 일할 때 그 업체들로부터 AOSP SDK 를 전달받고 그 위에서 작업을 주로 했었다.)

311. 서버팀과 클라이언트 개발 팀 간의 갈등 같은 것이 있었는가?

A. 과거 출장가서 현지 고객사 서버 개발자와 커뮤니케이션 하면서 어려웠던 적이 있다.

B. 당시 클라이언트 단에서 점검해줘야 하는 사항과 서버 단에서 점검해줘야 하는 사항, 즉, 서로의 레이어에서 정확한 검증을 위한 Test Case 가 모호했다. 그래서 그 테스트 케이스를 정립하고 테스트를 하는 데에 시간이 많이 소요되었다.

312. 레이아웃 구성 관련해서 주로 어떠한 레이아웃을 사용해보았는가?

A. Linear / Relative / Constraint / Frame 등을 사용해보았다.)

313. Constraint 레이아웃은 Relative 레이아웃과 비교했을 때 무엇이 차이가 있는가?

A. Relative
   i. 레이아웃 겹치기 불가능
   ii. 배치의 자유도 높음
   iii. 레이아웃 비율계산 적용 불가능
   iv. 복잡도가 약간 높음

B. Linear
   i. 레이아웃 겹치기 불가능
   ii. 배치의 자유도 낮음
   iii. 레이아웃 비율계산 적용 가능
   iv. 복잡도가 낮음

C. Constraint
   i. 부모, 자식 뷰 관계 속성 도입
   ii. 레이아웃 겹치기 가능
   iii. 배치의 자유도 높음
   iv. 레이아웃 비율계산 적용 가능
   v. 복잡도가 약간 높음

vi. Linear + Relative = Constraint 정도의 느낌

314. 회사에 멘토가 있었는가? 누구였는가?

A. 있었다. XXX 책임이었다

315. 안드로이드 플랫폼에서 개발의 장단점은 무엇이라고 생각하는가?

A. 장점
   i. 오픈소스 ➔ 관련 자료가 많다. 디버깅 용이
   ii. 자바 언어 사용
   iii. 스마트폰을 위한 완벽한 컴포넌트를 제공
   iv. 타 앱 연동이 수월하다
   v. ART 런타임 탑재

B. 단점
   i. SDK 의 잦은 업그레이드
   ii. 제품 출시의 경우 인증 절차 복잡하다

316. 앱 기획도 해봤는지?

A. 해보지 않았다. UX/UI 팀과 개발팀과 초기 개발 단계 때 시나리오 회의를 가지긴 하나 직접 기획에 참여하진 않는다.

317. 리눅스 마스터 2 급은 회사 다닐 때 취득한 것인가?

A. 그렇다. 동기 부여를 하기 위해 취득했다.

318. 본인이 생각하는 성장이란 무엇이고, 어떤 성장을 꿈꾸시나요?

A. 나 자신만 만족해서는 안되고 대외적인 평가와 나 스스로의 만족이 결합된 것이 성장이라고 생각한다.

B. 개발 업무 전반에서의 방대한 지식, 아래 사람들을 아우를 수 있는 팀 리더로서의 성장을 꿈꾼다.

319. 동료 또는 상사가 평가하는 나는 어떠한 사람인지?

A. 행동가, 유쾌한 사람, 열심히 하는 사람, 노력하는 사람

B. 같이 일하고 싶은 동료로 평가 받았다. 방어적인 태도를 취하지 않고 적극적인 태도로 업무에 임했기 때문이다.

320. 현재 및 전 직장에서 하신 업무를 소개해주세요.

A. 현재는 Launcher Application 유지보수를 맡고 있다.
B. 그 이전에는 설치마법사, 메뉴 트리 UI, 팀 내 / 외부 업체 공유를 위한 라이브러리 서비스 개발, IPTV 서비스 등을 개발해왔다.

321. 지난/현재 회사에서는 어떻게 성과를 내었는지요? (이전에 했던 프로젝트 성과 등)

A. 출장 업무, 외국 개발자와 협업/커뮤니케이션
B. IPTV Content 재생되는 것 확인
C. RCU 업체와의 협업, 제품 출시에 이바지

322. 개발 프로젝트에 대해 왜 개발했고, 어떻게 개발했는지?

A. 고객사 요구사항, 시나리오에 따른 앱 개발 업무
B. 단순 UI 앱은 APK 별로 담당자 한명, 코드 리뷰는 기술 매니저에게 받았다.
C. 방송 데이터 처리 부분은 3~4 명의 개발자가 모듈 별로 담당
D. 앱 개발 시 Jira 이용하여 Task/SubTask 정의하여 업무 진행

323. 본인의 배경이 다른 지원자들과 비교해서 이 직무에 얼마나 도움이 될 것 같은지?

A. 안드로이드 앱을 초기 개발 단계부터 릴리즈까지 담당 가능
B. 안드로이드 앱 개발 경력 8 년차, 앱 개발 시 UX/UI 팀과 협업
C. 안드로이드 UX/UI 에 익숙하다.
D. Zeplin 툴 사용에 익숙하며, UX/UI 팀과 항상 협업을 진행하였기 때문에 디자인 가이드 라인에 익숙하다.

324. 최근 하신 일 중 가장 뿌듯했던 일(자랑하고 싶은 일)이 있다면 말씀주세요

A. 히스토리가 복잡했던 앱의 이슈를 처리한 경험
B. 치명적인 이슈 발생 없이 앱 구현을 마친 경험 (Radio / 3rd Party 라이브러리 제공을 위한 서비스 앱 개발)
C. 과거 출장지에서 외국인 개발자와 협업해서 결국 IPTV Content Play 가 되는 것 까지 보고 온 것

325. 반대로 뼈아프게 생각하는 실수담이 있다면, 이 사례를 통해 배운 점은 무엇인지요?

A. 실수 라기보다 과거 많은 담당자들을 거치면서 히스토리 관리가 되지 않은 앱을 담당하게 되어 어려움을 겪은 적이 있다.
B. 개발을 할 때 초기의 컨셉 / 개발 방향 / 목표를 확실하게 잡는 것이 중요하구나를 깨달았다.

C. 서로 간의 개발의 방향성을 동일하게 가져 가야 한다. (생각의 동일시)

326. 다른 사람과 의견 차이가 있습니다. 본인의 생각이 더 좋은 것으로 보이는데 어떻게 해결하겠습니까?

A. 모든 사안에 가장 큰 문제는 서로 간의 커뮤니케이션 방식이라 생각한다.
B. 충분한 시간이 있다면 상대방을 배려하겠지만, 그렇지 않다면, 만약 내의견이 100% 맞다 라는 확신이 있고 상황이 긴급하다면 최대한 설득해서 내 방식으로 유도하겠다.
C. 또는 만약에 내 의견이 조금이라도 받아들이기에 부족한 부분이 있다고 판단되면 최대한 상대방 생각을 맞춰주겠다.
D. 타 부서 / 타인과 불화가 발생하는 상황 역시 커뮤니케이션 방식 문제라 생각한다.
E. 일단은 상대방의 말을 듣고 나서 그 다음 내 입장을 나중에 설명할 것이다.

327. 본인의 업무 스타일은?

A. 미션이 주어지면 바로 행동을 한다.
B. 예를 들어 구현 과제가 주어지면 요구 사항 / 해야할 일을 빠르게 정리 후 Jira Task / SubTask 를 만들어 순서대로 진행한다.
C. 이슈 관리 툴을 적극 활용하는 편이다. 자신에게 미션을 주는 느낌이라 동기부여가 되는 것 같다

328. 안드로이드 개발자로서 중요한 역량은 무엇인지요?

A. 집념, 인내, 끈기, 최신 안드로이드 기술 동향 파악 등과 같은 진부한 대답 보다는 다음과 같이 생각한다.
B. 암기력이라 생각한다. 그 이유는 아래와 같다
C. 전형화된 컴포넌트 사용 방법들이 존재
D. 안드로이드에서 개발하다 보면 자주 사용되는 컴포넌트들이 존재하고 이 컴포넌트 사용하여 개발 중에 발생하는 Exception 에 대한 대처 방법을 숙지해 놓는 것이 개발 속도를 향상시킬 수 있는 방법이라 생각한다.

329. 같이 일하기 싫은 유형의 사람은?

A. 대안 없이 상황에 대한 불만 많은 사람
B. 업무에 있어서 방어적인 사람

330. 업무 진행 시 이것만은 참을 수 없다 하는 것은?

A. 그런 것은 없다. 모든 것은 이야기 풀어나갈 수 있다고 생각한다.
B. 다만 업무 진행 시 지나치게 방어적이거나 상황에 대해 불만 많은 사람들이 있으면 힘들었다

331. 업무 외적으로나 혼자 있을 때 주로 무엇을 하는지?

A. 드라이브
B. 기타연주 (취미)
C. 친구들과 여행/운동

332. 존경하는 사람은 누구인지?

A. 아버지 / 기술 매니저 등
   i. 힘든 상황에서도 공과 사를 구분
   ii. 짜증을 내지 않고 한결 같이 사람을 대하는 사람들을 보면 존경스럽다.

333. 인생에서의 도전적인 경험이 있다면?

A. 삼수를 했던 경험

334. 리더십, 팀장급 에서 필요한 본인이 생각하기에 가장 필요한 요소는 무엇이라 생각하는가

A. 업무에 있어서는 당연히 전반을 아우를 수 있어야 한다.
B. 개인의 감정을 드러내지 않고 한결 같이 감정을 유지 하는 것이 중요하다
C. 공과 사를 구분하는 사람
D. 실수를 하거나 감정이 상하는 일이 있어도 다음날 털어낼 수 있는 대범한 사람

335. 만약에 스펙 문서중에 어느 한 Feature 를 누락해서 일이 커졌다. 이런 사태를 본인이라면 어떻게 처리할것인가?

A. 솔직하게 현 사태에 대해서 이야기하고 최대한 Waiver 를 이끌어 내겠다.
B. 다음 페이지 2 로 해당 Feature 를 최대한 넘기고 해당 Feature 와 더불어 더 많은 기능을 제공할 것을 약속할 것이다. 단 두번째 약속은 무슨 일이 생겨도 지켜야 한다.
C. 워커라운드로 접근하겠다

336. 리더십 요소 중에 본인이 가지고 있는 요소는 무엇인가?

A. 공과 사를 구분

B. 실수나 갈등이 있어도 다음날까지 영향 받지 않는 성향

337. 우리는 당신의 이전 회사와 같은 산업군은 아니다. 같은 업계로 안 가고 우리 쪽으로 온 이유가 있나?

A. 퇴직 사유와 같이 모바일 쪽 분야를 특화시키고자 해서 퇴사를 결심하였다.

B. 모바일 쪽 개발도 진행하고 있기 때문에 기존에 했던 업무에 더불어 더 많은 것들을 해 볼 수 있을 것 같아 지원했다.

338. 언제부터 이직을 결심하셨나요?

A. 작년 10 월달에 막연히 생각했었고 그때 이력서를 써보기 시작했다.

B. 본격적으로 구직활동을 시작한 것은 2021 년 5 월달부터이다.

339. 자바, 코틀린 사용 비중이 어떻게 되셨나요?

A. 자바 90%, 코틀린 10% 정도 였던 것 같다.

B. 기존 프로젝트 내 앱들은 자바로 구현되어 있고 최근 진행하는 프로젝트에서 구현하는 앱들은 코틀린을 사용해서 구현 중이다.

340. 네이티브 쪽 C, C++ 개발도 해봤냐?

A. 자바와 코틀린 사용해서 앱 만든 것 처럼 현업에서 프로그램을 만들어 보진 않았다.

B. Amlogic / Broadcom 칩셋을 사용했었고 특정 기능(HDMI 해상도 / 스크린 사이즈 조정)을 사용하기 위해 JNI 작업은 해보았다.

341. 기술적으로 Trouble Shooting 해 보신 경험이 있나요?

A. Handler Leak 발생 이슈 처리, Android Studio Profiler 사용하여 메모리 점유율 체크

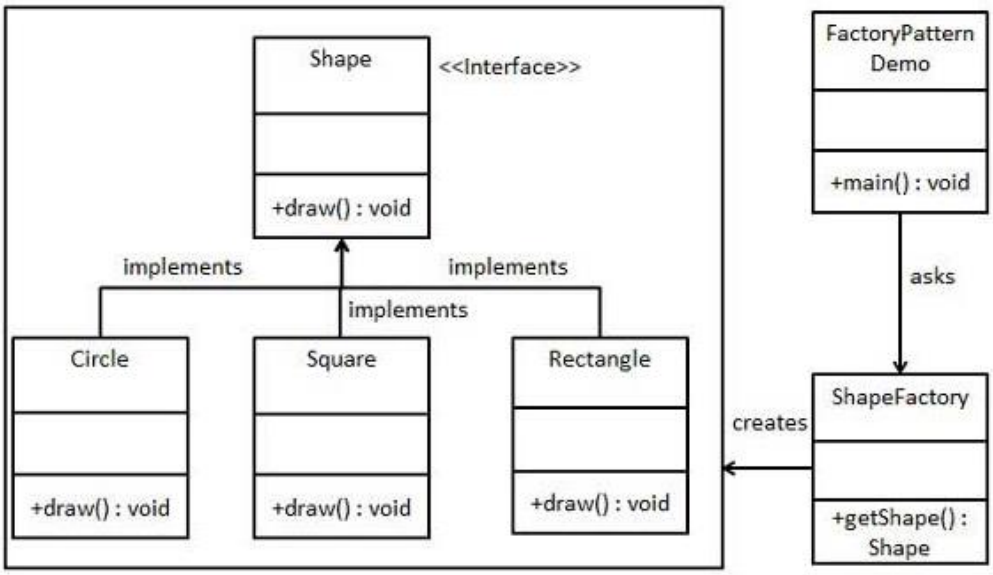
342. 디자인 패턴은 사용해 본 적이 있나요?

A. 그렇다. 파생되는 Feature 가 많은 곳은 Factory 패턴 적용

B. 싱글턴, 옵저버, 빌더 패턴 사용

343. 팩토리 패턴에 대해서 한번 화이트보드에 UML 로 그려보세요

A. UML



B. Code

Shape.java

```
public interface Shape {
   void draw();
}
```

Rectangle.java

```
public class Rectangle implements Shape {

   @Override
   public void draw() {
      System.out.println("Inside Rectangle::draw() method.");
   }
}
```

Square.java

```
public class Square implements Shape {

   @Override
   public void draw() {
      System.out.println("Inside Square::draw() method.");
   }
}
```

ShapeFactory.java

```
public class ShapeFactory {

   //use getShape method to get object of type shape
   public Shape getShape(String shapeType){
      if(shapeType == null){
         return null;
      }
      if(shapeType.equalsIgnoreCase("CIRCLE")){
         return new Circle();

      } else if(shapeType.equalsIgnoreCase("RECTANGLE")){
         return new Rectangle();

      } else if(shapeType.equalsIgnoreCase("SQUARE")){
         return new Square();
      }

      return null;
   }
}
```

FactoryPatternDemo.java

```
public class FactoryPatternDemo {

   public static void main(String[] args) {
      ShapeFactory shapeFactory = new ShapeFactory();

      //get an object of Circle and call its draw method.
      Shape shape1 = shapeFactory.getShape("CIRCLE");

      //call draw method of Circle
      shape1.draw();

      //get an object of Rectangle and call its draw method.
      Shape shape2 = shapeFactory.getShape("RECTANGLE");

      //call draw method of Rectangle
      shape2.draw();

      //get an object of Square and call its draw method.
      Shape shape3 = shapeFactory.getShape("SQUARE");

      //call draw method of square
      shape3.draw();
   }
}
```

344. 안드로이드 메뉴 트리 UI 앱은 커스텀한 Settings 앱인가요?

A. 그렇다. 고객사 시나리오에 따라 UI 를 새로 구현하였고, 기능은 Android Default Settings 와 거의 동일하다.

345. Settings 앱은 단순 리스트로 구성될 것 같은데 맞나요?

A. 그렇다. ListView 나 RecyclerView 를 사용해서 구현하였다.

346. RecylerView 의 뷰 재활용 과정에 대해 설명해달라.

A. RecyclerView 는 뷰 홀더를 필수적으로 구현해줘야 한다.

B. onCreateViewHolder 에서 뷰 홀더를 생성하고 뷰를 붙여준다.

C. onBindViewHolder 에서 재활용 되는 뷰가 호출되고 실행된다. 그리고 이 함수를 통해 리스트에 있는 position 값도 전달받는다.

347. 비동기 처리는 주로 어떻게 처리했는가?

A. Handler, Callback Interface 사용하여 처리

B. Message Handling 하는 부분을 최대한 한 곳으로 모은다.

C. 코루틴, Rx 기법 등 여러가지가 있는 것으로 안다.

348. Background, UI 처리 시 어떤 라이브러리 사용해서 처리했는가?

A. 특별한 라이브러리를 사용하지는 않았다.

B. AsyncTask 를 주로 사용했었다. Deprecated 된 이후로는 직접적으로 AsyncTask 를 사용하지 않고 AsyncTask 와 유사한 Custom Class 를 만들어서 사용하였다.

349. Launcher 앱에서 Background 처리는 어떻게 했나요?

A. 주로 AsyncTask 나 HandlerThread 의 Looper 를 가지는 Handler 를 사용해서 처리하였다.

B. Launcher 내 UI 핸들러, Work 핸들러 두가지로 구분해서 사용했다

350. 새로운 기술을 전파하기 위해 어떠한 노력을 했었나요?

A. 제품 개발 업무다 보니 업무 자체가 항상 타이트해서 적극적으로 행동은 하지 않았다. 장점에 대해 알리기 위해 세미나 진행하고 관련 문서를 공유하였다.

351. 빌드스크립트는 어떤 걸로 작업했나요?

A. Bash Shell Script 로 작업하였다.

B. 주로 PDK 빌드를 위한 App / 라이브러리 포팅, Android.mk 작성, 브링업 시 Configuration 작업, AOSP SDK 에 대한 패치 작업 등을 수행했다. (Bootloader 를 제외한 나머지 부분들을 수행했다.)

352. 다른 사람의 코드를 리뷰해 준 적이 있는가요?

A. 내가 과거에 담당했던 앱을 다른 사람이 수정을 한 경우
B. 경력 신입 브링업을 위해 Jira 로 업무 분할해서 코드 적용한 경우
C. 내가 하고 있는 모듈과 연관이 있는 경우

353. 메모리 릭 처리 시 그럼 프로파일러 사용해서 처리했나요? 자세한 과정을 알려주세요.

A. Handler 사용 이 후 Handler Looper 를 종료시켜 주지 않아 발생했었다.
B. 코드 적용 전 자주 재현되는 재현 스텝에서 프로파일러 작동 후 메모리 증가하는지 확인
C. 코드 적용 후 다시 확인
D. 코드 적용

354. 끝까지 메모리 릭 처리를 마무리 지었어야 되지 않나요? 왜 중간에 그만 두었나요?

A. 당시 여러가지 앱을 담당하고 있었고 담당할 수 있는 개발자도 부족하고 일정이 타이트했기 때문에 어느 정도 선에서 마무리 짓고 다른 앱 유지보수 작업으로 넘어갔었다.

355. WorkManager 사용해봤나요?

A. 사용해보지 않았다.
B. 구성
    i. WorkManager ➔ 처리해야 하는 작업을 자신의 큐에 넣고 관리
    ii. Worker ➔ 추상 클래스. 처리해야 하는 백그라운드 작업의 처리 코드를 이 클래스를 상속받아 doWork()를 오버라이드하여 작성
        1. 리턴값
            A. SUCCESS
            B. FAILURE
            C. RETRY
    iii. WorkRequest ➔ WorkManager 를 통해 실제 요청하게 될 개별 작업
        1. 처리해야 할 작업인 Worker 와 작업 반복 여부 및 작업 실행 조건, 제약 사항 등 이 작업을 어떻게 처리할 것인지에 대한 정보가 담겨져 있다.
        2. 반복 여부에 따라 2 가지로 나뉘어 짐
            A. OneTimeWorkRequest
            B. PeriodicWorkRequest

iv. WorkState ➔ WorkRequest 의 아이디와 해당 WorkRequest 의 현재 상태를 담는 클래스

356. 만약 경력신입으로 입사하게 되면 어떤 부분에 일조를 할 것인가요?

A. 상대적으로 최신 라이브러리 / 클래스를 사용한 경험이 적다.
B. 나와 비슷한 처지의 개발자가 앞으로도 들어올 가능성이 있을 것 같다
C. 미래의 후배들을 위해 최대한 간편하게 배우고 적용할 수 있는 문서와 코드를 작성할 것 이다.

357. 다른 사람 코드 리뷰를 해줄 때 주로 어떤 부분을 중점으로 봐줬나요?

A. 아래 내용을 체크했다.
   i. 한 Commit 에 하나의 주제만 있는지
   ii. Null Check 확실하게 했는지
   iii. 중복된 코드가 있는지
   iv. 변수의 네이밍이 직관적으로 가독성이 있는지

358. 만드신 앱들의 저장소가 각각 따로 존재했나요?

A. 그렇다. 각각 따로 따로 Git 이 존재하였다.
B. 이 저장소들을 한군데로 모은 전체 Full Source Manifest 파일이 있고 전체 빌드를 할 때 이 매니페스트를 보고 소스 싱크를 한 다음에 빌드한다.

359. AOSP 소스에서 앱은 어느 부분에 배치가 되었었나요?

A. 주로 AOSP Full Source Tree Root 에서 vendor/회사명/apps 또는 system 으로 구분해서 배치했다.
B. 3rd Party 앱의 경우는 vendor/회사명/prebuilt/apps 또는 libs 로 지정하여 배치하였다.

360. 현재 안드로이드 버전은 어떤 버전을 사용하고 있나요?

A. 현재 Pie 버전 사용중이다.

361. 버전 업그레이드 때마다 인증을 다시 해야 되지 않나요?

A. 그렇다. 업그레이드될 때마다 XTS 인증을 다시 해야 한다.

362. 인증 수행할 때 애로 사항은 어떤 것들이 있었나요?

A. 한번 CTS 인증 싸이클 돌리는 데에 20 시간 이상 소요된다.

B. 여러 PC 를 사용해서 병렬적으로 수행가능하고 이런 경우에는 PC 개수에 비례해서 테스트 시간이 줄어든다.

C. PASS 항목들이 한번 테스트에서 바로 PASS 되지 않는 경우가 많아 같은 테스트를 여러 번 수동으로 해줘야 한다.

D. 관련해서 환경 설정 (CTS 용 최신 테스트 스크립트 + PC + 동일 조건의 OTT/STB 시료)이 까다롭다.

363. 입사해서 공부를 하게 되면 업무 시간 때는 불가능하고 업무 외 활동을 활용해서 해야 할 것이다. 가능합니까?

A. 가능하다. 업무 시간 때 혼자 공부를 하는 것은 나도 원치 않는다.

B. 업무 시간 때에는 현재 내가 가지고 있는 스킬로 커버할 수 있는 일을 받아서 일단 수행하면서 업무 외 시간 때 추가적인 공부를 더 하겠다.

364. Git 은 많이 사용해봤나요? 보통 브랜치 관리 시 Git-Flow 를 사용하시나요?

A. 완전 git-flow 대로 관리를 하지는 않고 최대한 적용하려고 한다.

B. Develop 용 브랜치를 만들어서 Feature 추가 / 이슈 수정을 하고

C. 최종적으로 Master 브랜치에 Merge 해서 마무리 한다.

365. 이슈 처리할 때 다 뜯어고쳐야 되는 상황일 때 매니저와의 의견 차이나 마찰은 없었나요?

A. 특별히 없었다. 관련해서 Side-Effect 이 많이 발생했다면 문제가 되었을 법 한데 당시 특별한 이슈들은 없어서 잘 마무리 되었었다.

366. Framework 소스 볼 일이 많나요? Framework 소스도 앱 소스와 같이 있나요?

A. 네. AOSP Source 에서 Framework 와 App 소스 검토가 가능하고 이슈 발생 시 App 뿐만이 아니라 Framework 로그도 같이 보기 때문에 소스 검토할 기회는 많다.