**Vývojové platformy**: Operačné systémy iOS a Android sú dve hlavné platformy na vývoj mobilných aplikácií, ktoré majú svoje vlastné vývojové prostredia, nástroje a ekosystémy. Existujú aj multiplatformové vývojové nástroje, ktoré umožňujú vytvárať aplikácie pre viaceré platformy súčasne.

* iOS = Vývojové prostredie: Hlavným vývojovým prostredím pre iOS je Xcode, ktoré poskytuje nástroje na návrh používateľského rozhrania, úpravu kódu, ladenie a testovanie. Programovacie jazyky: Swift a Objective-C. História: iOS bol prvýkrát predstavený v roku 2007 s prvým iPhonom. Súčasnosť: iOS sa vyvinul v robustný ekosystém s pravidelnými aktualizáciami a podporou širokej škály zariadení Apple vrátane iPhonu, iPadu a hodiniek Apple Watch.
* Android OS = Vývojové prostredie: Hlavným vývojovým prostredím pre Android je Android Studio, ktoré ponúka integrované nástroje na návrh, kódovanie, ladenie a testovanie. Programovacie jazyky Java a Kotlin. História: Android bol prvýkrát vydaný v roku 2008 ako otvorená platforma pre mobilné zariadenia. Súčasnosť: Android je najrozšírenejší mobilný operačný systém s podporou širokej škály zariadení od rôznych výrobcov.
* Cross-platform = Vývojové nástroje: React Native, Flutter, Xamarin, Unity. Výhody: možnosť napísať kód raz a nasadiť ho na viacerých platformách (iOS, Android). Nevýhody: Môže byť náročnejší na výkon a prispôsobenie natívnych funkcií.

**Vývojové prostredie**:

* iOS – Xcode = IDE vyvinuté spoločnosťou Apple na vývoj aplikácií pre iOS. Ponúka nástroj Interface Builder, ktorý umožňuje grafický návrh používateľského rozhrania, a integrované nástroje na testovanie a ladenie.
* Android – Android Studio = Oficiálne IDE na vývoj aplikácií pre Android. Poskytuje nástroje na návrh používateľského rozhrania, ladenie a správu projektov.

**História a súčastnosť**:

* iOS = História: Systém iOS 1.0 bol predstavený v roku 2007 s prvým iPhonom. Odvtedy prešiel mnohými zmenami a aktualizáciami vrátane pridania obchodu App Store, multitaskingu a podpory ďalších zariadení, ako je iPad a hodinky Apple Watch. Súčasnosť: Najnovšia verzia iOS obsahuje pokročilé funkcie ako ARKit na rozšírenú realitu, nové bezpečnostné funkcie a vyšší výkon.
* Android = História: Systém Android 1.0 bol predstavený v roku 2008. S každou novou verziou pribúdali nové funkcie, ako napríklad podpora viacerých obrazoviek, vylepšené používateľské rozhranie a pokročilé bezpečnostné funkcie. Súčasnosť: Systém Android ponúka širokú podporu pre rôzne zariadenia a formáty vrátane smartfónov, tabletov, televízorov a automobilov.

**Základy programovania v Jave**: Java je hlavný programovací jazyk používaný na vývoj aplikácií pre Android. Základné pojmy:

* Syntax: Java je objektovo orientovaný jazyk s jasnou a jednoduchou syntaxou.
* Triedy a objekty: základné stavebné prvky jazyka Java. Trieda je šablóna pre objekty, ktoré obsahujú metódy a atribúty. Objekt je inštancia triedy.
* Dedičnosť: mechanizmus, ktorý umožňuje jednej triede dediť atribúty a metódy inej triedy.
* Polymorfizmus: Schopnosť objektov rôznych tried spracúvať rovnaké rozhranie.
* Výnimky: Mechanizmus na spracovanie chýb počas vykonávania programu.

**Úvod do vývojového prostredia Android Studio (LogCat, ADB)**:

* LogCat = LogCat je nástroj na zobrazovanie protokolov spustených aplikácií. Pomáha pri diagnostike problémov a ladení. Zobrazuje výstupy z rôznych úrovní protokolovania, ako sú ladenie, chyby, informácie atď.
* ADB (Android Debug Bridge): ADB je nástroj na komunikáciu so zariadeniami so systémom Android. Umožňuje inštalovať aplikácie, spúšťať príkazy a ladiť aplikácie.

**Android SDK**: Android SDK (Software Development Kit) obsahuje všetky nástroje potrebné na vývoj aplikácií pre Android vrátane knižníc, rozhraní API a nástrojov na ladenie. Obsahuje aj emulátory na testovanie aplikácií na rôznych verziách systému Android.

**Senzory (WIFI, BT, GPS, GSM, kompas, gyroskop, akcelerometer)**: Zariadenia so systémom Android často obsahujú rôzne snímače, ktoré umožňujú vytvárať bohaté a interaktívne aplikácie:

* WIFI = Umožňuje pripojenie k bezdrôtovým sieťam. API na správu pripojení, skenovanie dostupných sietí atď.
* Bluetooth = Umožňuje bezdrôtovú komunikáciu s inými zariadeniami. Podpora rôznych profilov, napríklad klasického Bluetooth a Bluetooth Low Energy (BLE).
* GPS = Umožňuje určovanie polohy pomocou satelitov. API na získanie aktuálnej polohy, sledovanie polohy atď.
* GSM = Umožňuje komunikáciu prostredníctvom mobilných sietí. API na správu telefónnych hovorov, posielanie SMS atď.
* Kompas = Umožňuje určenie smeru. Na určenie smeru vzhľadom na magnetické pole Zeme používa magnetometer.
* Gyroskop = Meria orientáciu a uhlovú rýchlosť. Používa sa na zisťovanie otáčania a naklonenia zariadenia.
* Akcelerometer = Meria zrýchlenie a dokáže rozpoznať pohyb a otrasy. Používa sa na detekciu pádov, zmien orientácie zariadenia atď.

**Android – štruktúra projektu**: Štruktúra projektu Android zahŕňa niekoľko kľúčových komponentov

* Manifest = Súbor AndroidManifest.xml obsahuje dôležité informácie o aplikácii, napríklad oprávnenia, aktivity a služby.
* Kód = Obsahuje zdrojový kód aplikácie, zvyčajne v priečinkoch java alebo kotlin. Hlavné súbory sú triedy, ktoré definujú správanie aplikácie.
* Zdroje (resources) = Obsahuje súbory, ako sú obrázky, rozloženia, reťazce atď. Zdroje sú usporiadané v priečinkoch, ako napríklad res/drawable, res/layout, res/values atď.
* Gradle build toolkit = Nástroj na automatizáciu procesu zostavovania, správu závislostí a konfiguráciu projektu.

**Tvorba UI**: Vytvorenie používateľského rozhrania (UI) v systéme Android zahŕňa

* XML = Slúži na definovanie rozloženia a vzhľadu aplikácie.
* LinearLayout = Usporiada prvky do riadku alebo stĺpca. Používa sa na jednoduché a pravidelné usporiadanie prvkov.
* RelativeLayout = Umožňuje relatívne umiestnenie prvkov voči sebe. Používa sa pri zložitejšom usporiadaní prvkov.
* Štýly = Definujú vzhľad a štýl prvkov aplikácie. Štýly možno definovať v súbore res/values/styles.xml
* Activity = Základný stavebný prvok aplikácie, ktorým je jedna obrazovka s používateľským rozhraním. Každá aktivita má svoj vlastný životný cyklus a môže obsahovať viacero fragmentov.
* Fragment = Modulárna časť používateľského rozhrania, ktorú možno použiť v rámci aktivity. Fragmenty umožňujú flexibilné a opakovane použiteľné komponenty používateľského rozhrania.
* Zmeny konfigurácie = Odstraňovanie problémov súvisiacich so zmenami konfigurácie, ako je napríklad otáčanie obrazovky. Možnosti zahŕňajú použitie ViewModel na zachovanie údajov počas zmien konfigurácie.
* Optimalizácia pre rôzne zariadenia = Použitie rôznych rozložení, zdrojov obrázkov a veľkostí textu pre rôzne veľkosti a rozlíšenia obrazovky. Responzívny dizajn pomocou ConstraintLayout a ďalších nástrojov.

**Životný cyklus aplikácie**:

Activity Lifecycle = Systém Android riadi životný cyklus aktivity prostredníctvom série stavov. Každý stav predstavuje určitú fázu života aktivity:

1. onCreate(): Inicializácia aktivity. Tu sa zvyčajne nastavuje používateľské rozhranie a inicializujú sa potrebné zdroje.
2. onStart(): Aktivita sa stáva viditeľnou pre používateľa, ale ešte nie je v popredí.
3. onResume(): Aktivita sa dostane do popredia a používateľ môže interagovať.
4. onPause(): Aktivita stráca fokus, ale je stále viditeľná. Zvyčajne sa tu ukladajú zmeny, ktoré by mali byť trvalé.
5. onStop(): Aktivita nie je viditeľná. Systém môže uvoľniť zdroje.
6. onDestroy(): Aktivita je zničená. Tu sa uvoľnia všetky zostávajúce zdroje.

Komunikácia medzi aktivitami =

1. Intent (Zámer): Mechanizmus na začatie nových činností alebo komunikáciu medzi činnosťami. Zámery môžu byť explicitné (zamerané na konkrétnu činnosť) alebo implicitné (so všeobecným zámerom, ktorý môže byť naplnený viacerými aplikáciami).
2. BroadcastReceiver: Komponent na prijímanie a spracovanie globálneho vysielania. Vysielania môžu byť systémové (napr. zmena stavu batérie) alebo definované aplikáciou.

**Interakcia s používateľmi**:

* Notifikácie = Informujú používateľa o udalostiach mimo aplikácie. Zobrazujú sa v oznamovacej lište. Systém Android poskytuje rozhranie API oznámení na vytváranie a správu oznámení.
* Toast správy = Krátke správy, ktoré sa používateľovi zobrazia na krátky čas. Neprerušujú aktuálnu činnosť používateľa.
* SnackBar = Rozšírená verzia prípitkových správ s možnosťou akcie. Zobrazuje sa v spodnej časti obrazovky.
* Dialógy = Okná, ktoré vyžadujú interakciu používateľa, ako napríklad AlertDialog na potvrdzovanie akcií.

**Trvalé dáta v aplikácii**:

* SharedPreferences = Ukladanie jednoduchých dvojíc kľúč-hodnota údajov. Slúži na ukladanie nastavení a jednoduchých údajov.
* Databáza SQLite = Relačný systém správy databáz na komplexnejšie ukladanie údajov.
* Správa súborov = Ukladanie a čítanie údajov z internej alebo externej pamäte zariadenia.

**Práce na pozadí**:

* Threads = nízkoúrovňový mechanizmus na spúšťanie kódu na pozadí
* AsyncTask = jednoduchšie API na spúšťanie úloh na pozadí a aktualizáciu používateľského rozhrania
* Services: komponenty pre dlhodobé úlohy na pozadí, ktoré nemajú priamu interakciu s používateľom

**Android Design Guidelines**:

* Dokumentácia: oficiálna dokumentácia k systému Android obsahuje pokyny na navrhovanie aplikácií podľa zásad Material Designu - Navrhnutý tak, aby poskytoval konzistentné používateľské prostredie naprieč zariadeniami a platformami.
* Optimalizácia: osvedčené postupy pre efektívne kódovanie, výkon a používateľskú skúsenosť. Efektívne využívanie pamäte a procesora. Minimalizácia spotreby batérie. Optimalizácia používateľského rozhrania pre rôzne veľkosti obrazovky.
* Používanie knižníc: populárne knižnice ako Retrofit (pre HTTP požiadavky), Glide (pre prácu s obrázkami) a Room (pre prácu s databázami). Retrofit: Knižnica na jednoduchú komunikáciu s rozhraním REST API. Glide: Knižnica na efektívne vyhľadávanie a spracovanie obrázkov. Room: Abstrakcia nad databázou SQLite na jednoduchšiu správu databázových operácií.

**Komunikácia so senzormi - (WIFI, BT, GPS, GSM, kompas, gyroskop, akcelerometer)**: Systém Android poskytuje rozhrania API na komunikáciu so snímačmi, ktoré aplikáciám umožňujú prístup k údajom z rôznych snímačov:

* WIFI: API na správu pripojení k bezdrôtovým sieťam, skenovanie dostupných sietí a správu pripojení.
* Bluetooth (BT): API na komunikáciu s inými zariadeniami prostredníctvom Bluetooth vrátane klasického Bluetooth a Bluetooth Low Energy (BLE).
* GPS: API na získavanie aktuálnej polohy zariadenia pomocou satelitov, sledovanie polohy a ďalšie funkcie súvisiace s polohou.
* GSM: API na správu mobilných sietí, telefónnych hovorov, SMS a ďalších funkcií súvisiacich s mobilnou komunikáciou.
* Kompas: Rozhranie API na prístup k magnetometru zariadenia s cieľom určiť jeho smer vzhľadom na magnetické pole Zeme.
* Gyroskop: API na prístup k gyroskopu zariadenia, ktorý meria orientáciu a uhlovú rýchlosť.
* Akcelerometer: API na prístup k akcelerometru zariadenia, ktorý meria zrýchlenie a dokáže zistiť pohyb a otrasy.

**Zdieľanie údajov medzi aplikáciami/servermi**:

* Content providers: umožňujú aplikáciám zdieľať údaje s inými aplikáciami. Poskytujú štandardizované rozhranie na prístup k údajom, ako sú kontakty, médiá a ďalšie.
* SyncAdapter: mechanizmus na synchronizáciu údajov medzi zariadeniami a servermi. Umožňuje automatizovanú a efektívnu synchronizáciu údajov.
* REST API: rozhranie na komunikáciu medzi klientom a serverom pomocou požiadaviek HTTP. Používa sa na prenos údajov medzi aplikáciami a servermi.

**Podpisovanie aplikácií, publikovanie v službe Google Play**:

* Podpisovanie aplikácie: proces zabezpečenia pravosti a integrity aplikácie pred jej zverejnením. Podpísaná aplikácia obsahuje digitálny podpis, ktorý overuje jej pôvod a zaručuje, že s ňou nebolo manipulované.
* Publikovanie v službe Google Play: proces nahrávania a distribúcie aplikácií prostredníctvom obchodu Google Play. Vývojári si musia vytvoriť účet Google Play Developer a nahrať súbor APK spolu s opisom aplikácie, snímkami obrazovky a ďalšími informáciami.
* Bezplatné x platené aplikácie: rozhodnutie o stratégii monetizácie aplikácie. Bezplatné aplikácie môžu generovať príjmy prostredníctvom reklám alebo nákupov v aplikáciách, zatiaľ čo platené aplikácie vyžadujú jednorazový nákup.
* Reklamy: Integrácia reklamy na generovanie príjmov. Medzi obľúbené reklamné platformy patria Google AdMob, Facebook Audience Network a ďalšie.