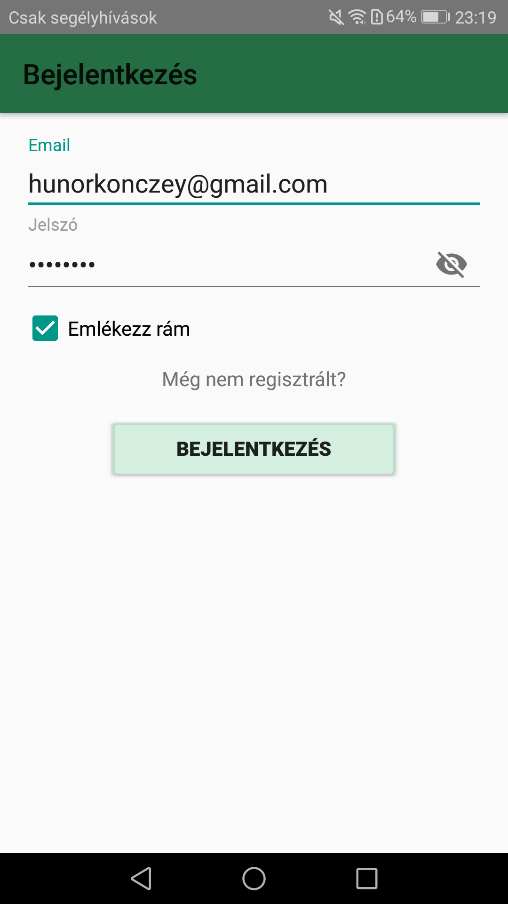
**Android alkalmazás**

Az Android alapú mobil alkalmazás, amelyet készítettem, a diákok számára lesz használható, akik az alkalmazás segítségével tudják megerősíteni az órai jelenlétüket. Az alkalmazás tervezése során a fő szempontok egyike az volt, hogy egyszerű legyen használni, de tudjon olyan funkciókat is, ami például a saját jelenléte nyomon követése.

**Használata:**

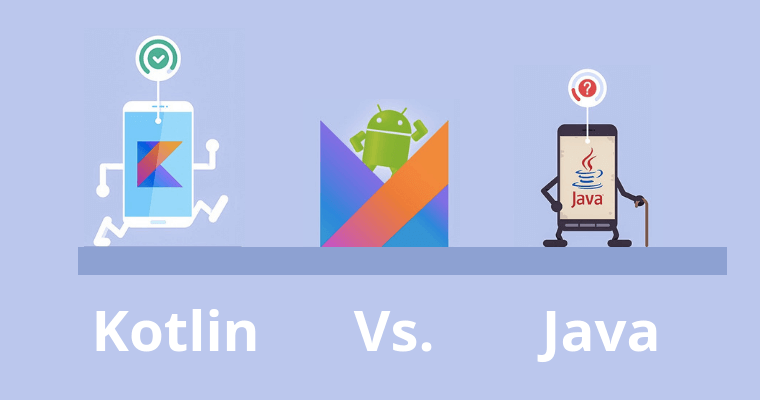
Első sorban a diákunknak regisztrálnia kell az alkalmazáson keresztül, itt szükség-e lesz a neptun azonosítójára, mivel ez egy egyedi azonosító kód. Sikeres regisztráció után szükséges bejelentkezni az alkalmazásba, itt elmenthetjük az adatainkat, ezzel is megkönnyítve a későbbi használatot. Bejelentkezés után a diák fel kell vegye azokat a tárgyakat, amely órákon részt vesz és használva van a rendszer. Sikeres tárgyfelvétel után lehet használni a QR kód olvasót. Megfelelő kódot beszkennelve megtörténik a jelenlét hozzáadása az aktuális órához. Ha a későbbiekben nyomon szeretnénk követni a jelenlétünket, akkor erre is megvan a lehetőség, ahol láthatjuk, hogy hány óra volt megtartva az adott tárgyból.

**Felhasznált technológiák:**

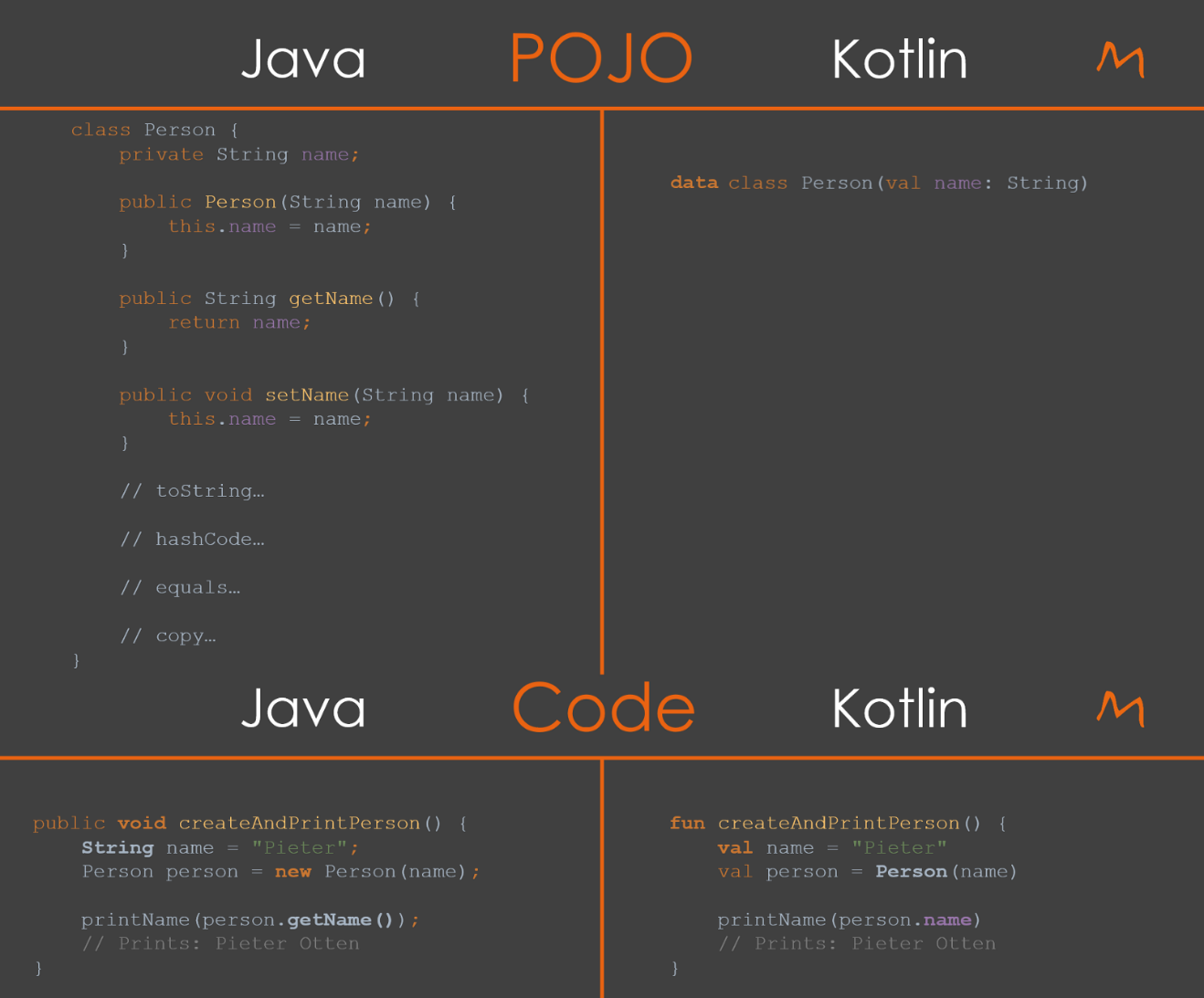
1. **Kotlin vs. Java**

Az alkalmazást Kotlin programozási nyelvben írtam és a következőkben meg indoklom, hogy miért is választottam a már jól megszokott és szeretett Java helyett. A Kotlin nem annyira régi nyelv, hiszen a Google is csak 2017-ben ismerte el, mint második hivatalos nyelv a Java mellett. Azonban napjainkban, már a Google által is ajánlott lett a használata a vetélytársával szemben. Célja a magasabb termelést hivatott növelni, mivel növeli a kódolási sebességet és hatékonyságot ezáltal a fejlesztők gyorsabban és hatékonyabban írhatnak kódokat. Nagyban elősegíti az úgynevezett “boilerplate” kódok csökkenését, ami átláthatóbbá teszi a projekteket.



**A Kotlin nyelv rövid bemutatása:**

Az egyik legnagyobb könnyítés az Android programozásban, hogy ezentúl nincs szükségünk a findViewById-t használni, ha a nézetről szeretnénk elérni valamit. Egyszerűen használhatjuk, mintha már egy deklarált változó lenne a Kotlin képes importálni minden nézetett egy adott osztályon belül. Problémában akkor ütközhetünk, ha több nézetett importálunk be egy osztályba és vannak a nézeten olyan elemek, amelyek az id-ja megegyezik.

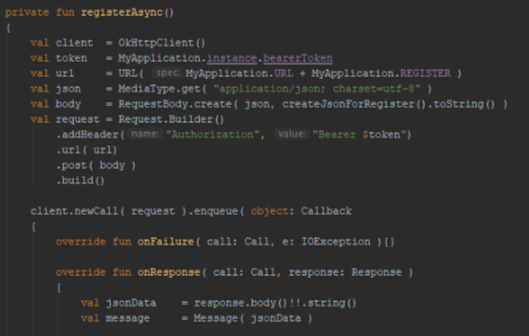


Egy másik nagy előny a Kotlinak, az olyan osztályok leegyszerűsítése, amelyek csak adat tárolásra szolgálnak, mivel lehetőség van az osztály neve mellé írni a data kulcsszót és ezzel teljes mértékben megspórolhatok a getter, setter, konstruktor függvények, mivel a Kotlin gondoskodni fog róluk, hogy a megírásuk nélkül használhatjuk a következőkben. Ezen kívül látható a fenti példakódban, hogy egy új objektum példányosításánál sem kell többé a new kulcsszó, ami kis apróságnak tűnhet, de nagy projektek esetén ez többszáz szót is jelenthet akár.

Nyilvánvalóan a Java fejlesztők egyik legnagyobb rémálma közé tartozik a NullPointerException, mivel a nyelv lehetővé teszi, hogy a fejlesztők null értéket adjanak bármely változónak, de ha hivatkozni szeretnénk rá akkor jön elő a NullPointerException. A Kotlinban minden típus úgynevezett non-nullable, ezért nem is kell ellenőrizni azt, hogy null-e egy objektum. Azonban van egy lehetőség, amely által lehetőségünk van megadni egy változónak, hogy null legyen, ami deklarálása a következő lesz: var name: String? = null. Ezenfelül eltűnnek a statikus tagok is. Összeségében, a Kotlin, ha továbbra is így halad pár éven belül teljesen lecserélheti a Javat, az Androidos alkalmazások fejlesztésében. Nekem volt szerencsém, mindkét nyelvben dolgozni és fejleszteni és a saját tapasztalataim alapján gyorsabb, élvezhetőbb és átláthatóbb lett a kód írása és ez mellett rengeteg plusz lehetőség van benne, ami még szerintem kihasználatlan.

1. **Külső könyvtárak:**
   1. **OkHttp 3.0**

A könyvtár használatához először is engedélyezni kell az internet hozzáférést a programunkban. Ez a könyvtár leegyszerűsíti a kommunikációt az API-k meg az alkalmazásunk között. Mi a kérések után a legtöbb esetben egy json fájl-t várunk vissza a szervertől, amelyben vagy megkapjuk az adatainkat, van visszajön a hiba üzenet a. Ha csak egy listát szeretnénk visszakapni a szervertől, akkor ennek a könyvtárnak a segítségével, pár sor kóddal kész is vagyunk. A json fájlok feldolgozására nem használtam más könyvtárat, mivel eléggé sajátos json fájlokat küld vissza szerver, így egyszerűbb volt osztályokat létrehozni. A visszakapott json-t pedig feldolgozni és objektumokat készíteni belőle. Feldolgozás után pedig a legtöbb adatott megjeleníteni listában.



A fenti rövid kód részletben látható, hogy kezdünk el egy kérést írni az OkHttp segítségével, a fenti kód a regisztrációnál van felhasználva. Szükségünk van először is az Url-re amelyre küldjük a kérést, a regisztrációnál mi jsonként küldjük el az adatokat a szerverhez. A szerver az ellenőrzést, hogy be vagyunk-e jelentkezve token alapján végzi, amelyet bejelentkezés után küld vissza nekünk és mentünk el az alkalmazásunkon belül, majd minden kérés során küldjük a token-t, hogy érvényes-e. Ezzel biztosítva van a biztonságos adatküldés a kliens és szerver között.

* 1. **ZXing**

A ZXing egy olyan könyvtár, amely elvégzi a vonalkód feldolgozást, 1D és 2D-s vonalkódok esetén is. Az alkalmazásban azért van rá szükség, hogy a tanár által kivetített QR kódot a diákok sikeresen betudják olvasni ezzel felkerülnek a jelenléti listára. A könyvtár használata szintén egyszerű és könnyen implementálható. A QR kódot a weboldal generálja ki, ami a tanárok számára lesz elérhető, ez az alkalmazásunk harmadik komponense.