* a Gradle Windows-on nem a legjobb. Sokszor kellett újraindítanom a gépet (néha elég volt csak az IDE-t), mert GradleConnectionException miatt nem tudott fordítani
* A Fragments –ekkel jobban darabolható a kód, mint WP-on
* WP-n a silverlight miatt amúgy is sokmindent egybe kellett írni. Pl talán ResourceDictionary se volt… Csomó mindent az app.xaml fájlba kellett rakni!
* Android: meglepően dinamukusan ment a proxy: volt és működött a beépített süti kezelés, anélkül, h fél napokat kellett vna kódolni hozzá!
  + Nem volt viszont WSDL-ből proxy-t generáló cucc. WP-nél legalább volt, még ha használhatatlan is, mert a süti kezelés silverlight-ban kb kivitelezhetetlen…
* Android-on sokkal több UI dolgot kellett kódból írni, mint WP-n. Viszont ez így könnyebb volt, persze hosszú távon melyik a jobb, az passz…
* WP volt később, szebb az API-ja is. Pl tud olyat, h a képnek src-ben URL-t állítunk be, és kész. Nem kell kézzel megírni a letöltő logikát
* JavaDoc: A html sokkal értelmesebb választás javaDoc-hoz, mint egy külön xml definálása, mint C#-nál
* Mindkettőben van design time data
* Mindkettőre igaz: felejtsd el, ha valami nem-basic dolgot tudtál Java-ban/C#-ban, itt úgyse használhatod
  + Android: anonym Fragment –et tilos használni, mert nincs default ctor-ja, így a keretrendszer nem tudja össze-vissza törölgetni meg újra példányosítani kedvére.
* Android: ez a folytonos Activity/Fragment törlés elég nehézkessé teszi a backendről egyszer már letöltött adatok tárolását. Főleg, mert a Bunble objektum, ami jár hozzá adat tárolás céljából, nem enged Object-et átadni; szóval nem referencia szerint adható csak át a state
* A XAML-ös megközelítés mindenképp egy jóval érettebb gondolat. Android-on eléggé hiányzik az adatkötés… Meg a ViewModel-ek is!
* Ctx menü máshogy műxik. WP-n bármihez beállíthatom, amihez csak akarom. Android-on… talán ott is…
* A XAML sokkal ügyesebb, mint ez az Android xml… Egységes, mindenhol ugyanúgy tudom szerkeszteni a child elementeket (nagyjából). Míg android-nál minden esetben egyéni a cucc…