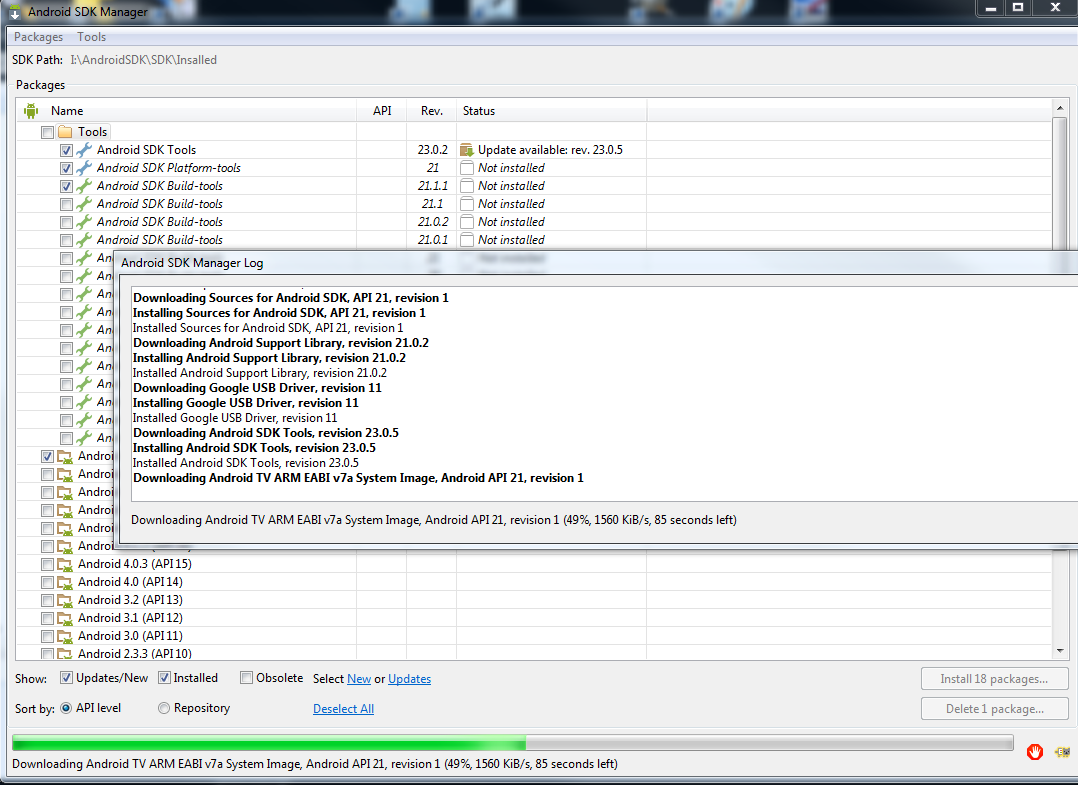
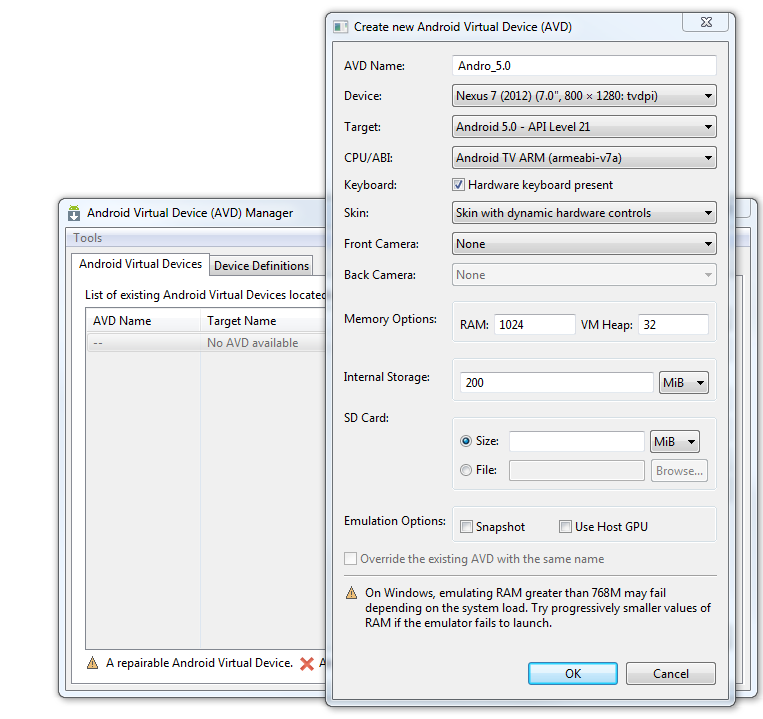
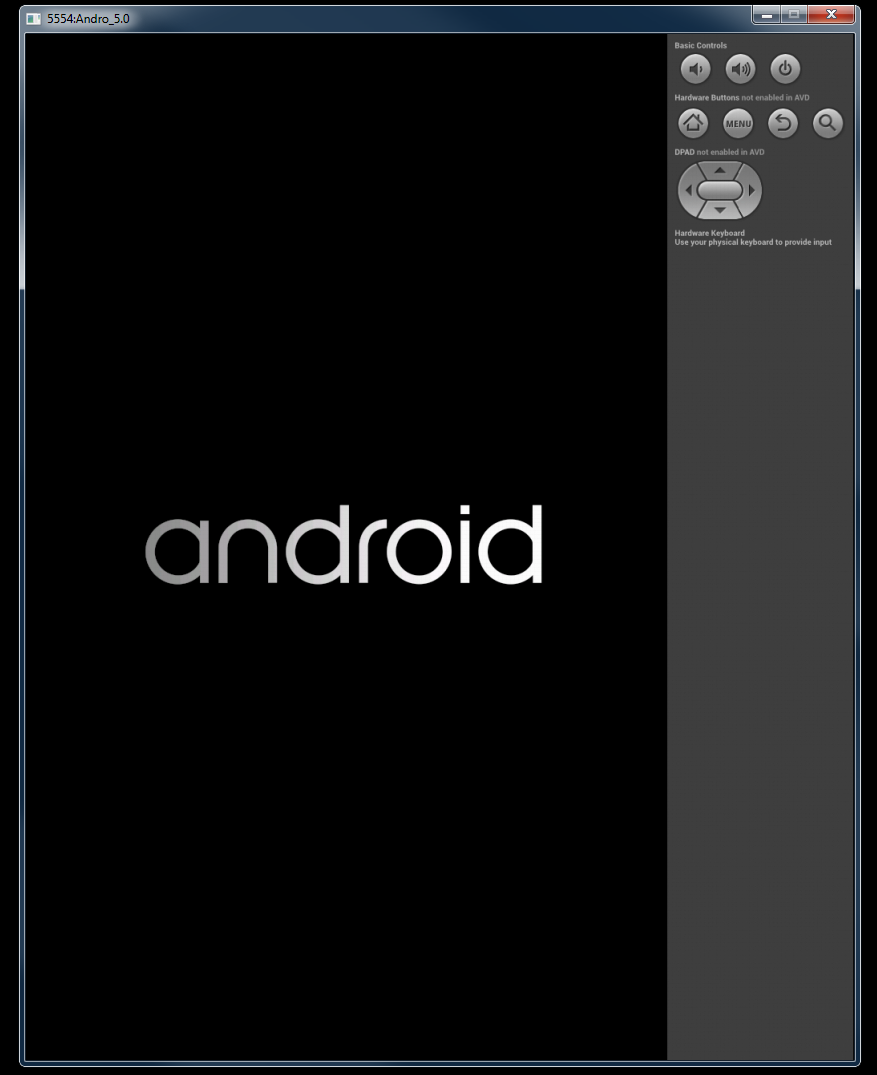
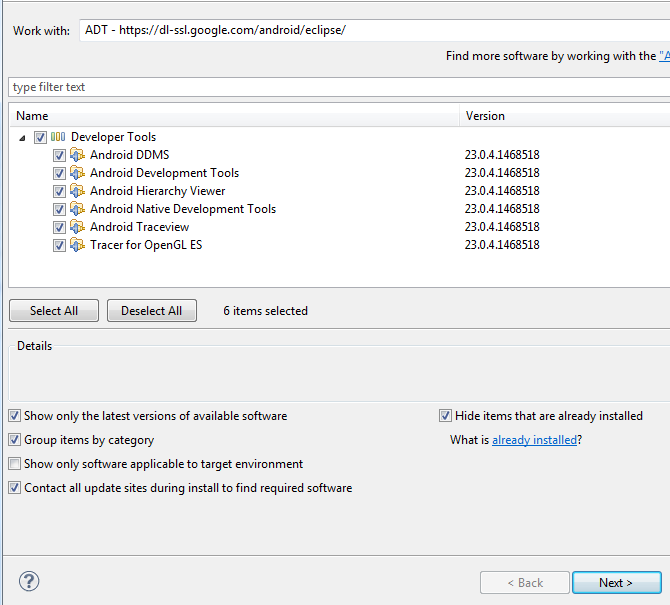
Teszt házi - Lovas István (K971UM)

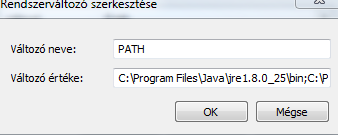
# Környezet beállítása

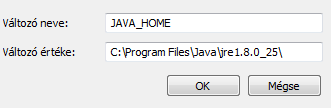


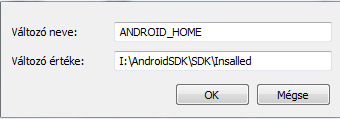


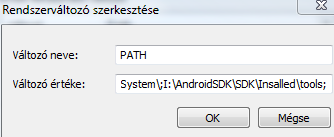


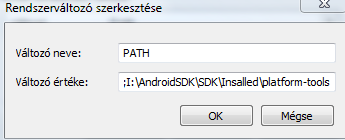


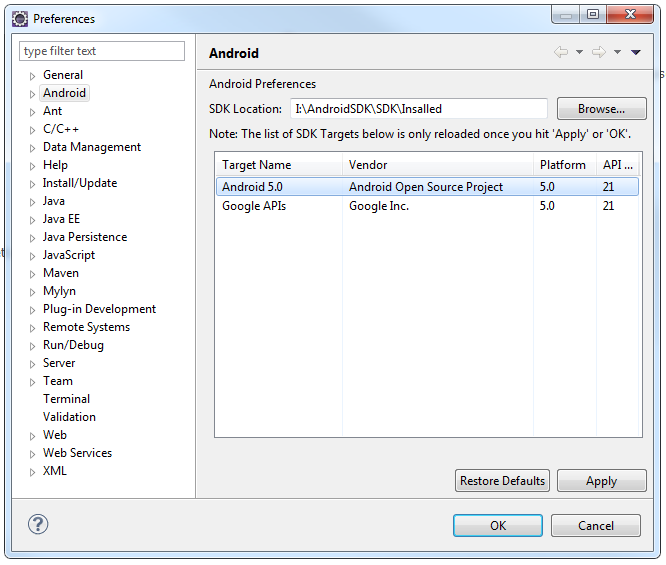


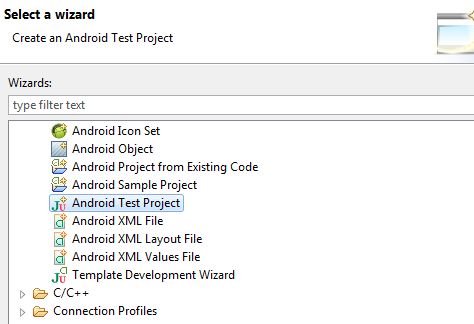












# Kérdések

## Mi a menete a tesztesetek feldolgozásának automatizáláshoz?

1. Elkészítés (QA Team)
2. Automatizálás (Automatization Team)
3. Rangsorolás (QA Team)
4. Egyesít (Automatization Team); Goto 1.

## Sorolj fel három olyan teszt automatizáló eszközt, ami nem szerepel a prezentációban!

HP - QuickTest Professional (QTP)

MS - Coded UI Test (CUIT)

Telerik TestStudio

Bónusz: ez a prezentáció nem tartalmazta a Seleniumot, se a Watirt

## Mi a Facade pattern?

Egy programtervezési minta, mely egy egyszerűsített felület biztosít egy bonyolultabb rendszerhez, ezzel megkönnyítve a vele való munkát, és a rendszer tesztelését. Egy absztrakciós réteget nyújt, szétválasztja a szolgáltató komponenst és annak felhasználóit, így a szolgáltatót egyszerűbb módosítani, csökkenti a függőséget.

## Mi a lényeges különbség a struktúra és a folyamat modellezés között?

A struktúra modell, a tesztelendő program szerkezetét követi, annak a leképzése. A művelet ezen a leképződésen hajtja végre. (Pl.: egy lap leképzése PageObject-tel és annak a metódusait hívjuk sorba)

A Flow modellben a folyamatokat írjuk le, nem érdekes, hogy pontosan mi hajtja végre, maga a cselekvések és azok sorrendje áll a középpontban. Ennek talán kicsit magasabb az absztrakciója, kicsit deklaratívabb, a „mit csináljon”-t adjuk meg, a hogyannal szemben. (persze mélyebb szinten az is le van programozva imperatívan)

## A tri-layer felépítésnek milyen rétegei vannak, és mi a szerepe az egyes rétegeknek?

Core layer: Projecteken átívelő közös újrafelhasználható kódbázis.

Inter layer: Az adott applikációban előforduló összes kód.

Test layer: Maguk a teszt esetek, amik felhasználják az alattuk lévő szint szolgáltatásait