

# JPA alapú teszteléstámogatás

Készítette: Ferencz Endre, Bozóki Szilárd  
Konzulensek: Budai Péter, Dr. Goldschmidt Balázs

# Bevezető

- A tesztelés jelentősége
- Az automatizált tesztelés
  - Előnyei:
    - konzisztens eredmények és adatok
    - megismételhetőség
    - könnyű karbantarthatóság
    - hatékonyabb erőforrás felhasználás
    - jelentés készítése
- Automatizált tesztadat-generálás
- A Java Persistence API szerepe



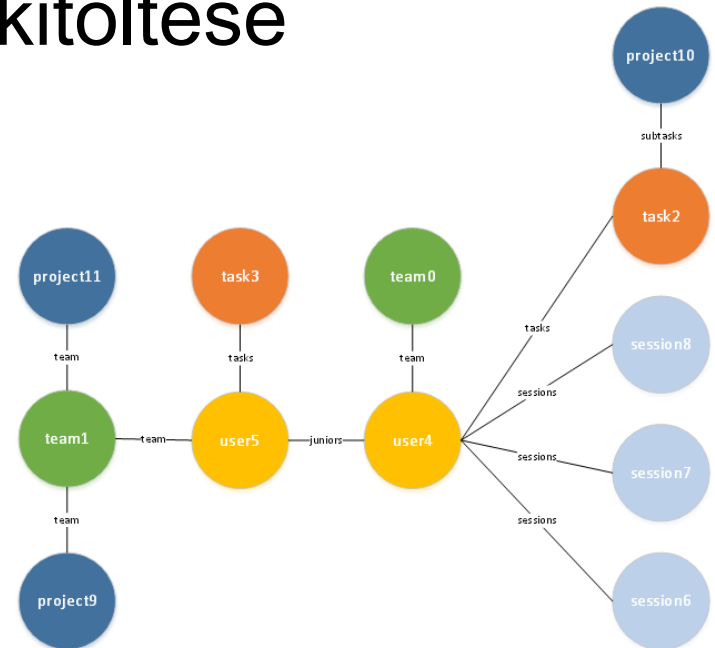
# A probléma felvetése

- A legtöbb eszköz véletlenszerű mintákat ad
- Elérhető szolgáltatások:
  - numerikus értékekre vonatkozó korlátok,
  - eloszlások,
  - szöveges paraméterek,
  - minta adatok használata,
  - referenciális integritás (idegen kulcs),
  - egyediség.
- Igény mélyebb összefüggésekre:
  - osztályszinten,
  - objektum szinten.



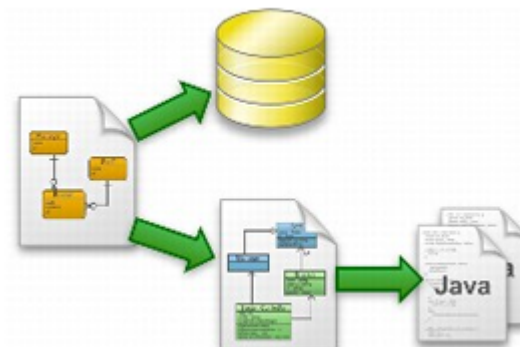
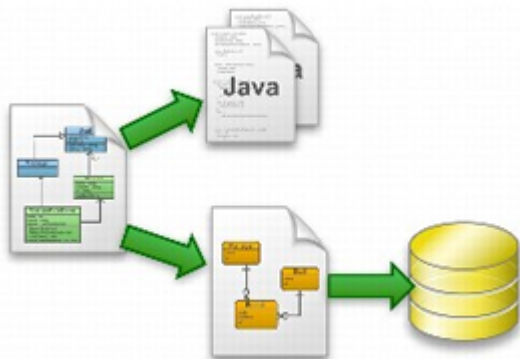
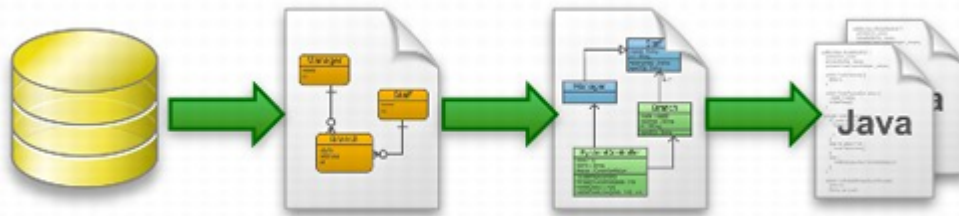
# Mélyebb összefüggések

- Objektorientált programozás szempontjából speciális esetek
- Redundáns attribútumok kitöltése
- Gráf alapú feltételek:
  - körmentesség,
  - többszörös összefüggés,
  - elkülönülő szigetek,
  - rekurzív adatstruktúrák.



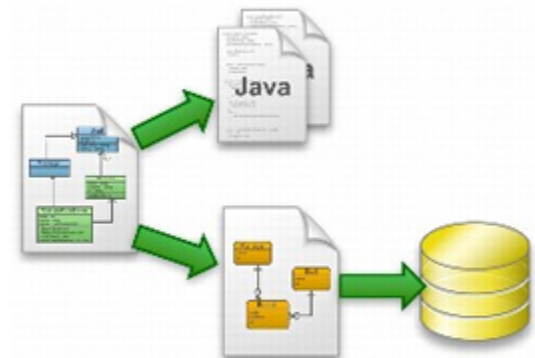
# Adatbázismodell elemzése

- Relációs adatbázis és objektumorientált modell kapcsolata



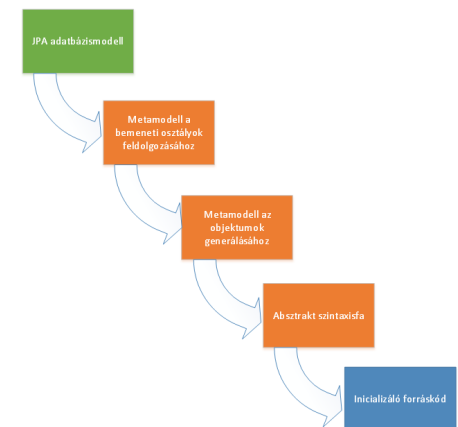
# Adatbázismodell elemzése

- Az adatbázismodell JPA annotációkkal ellátott entitásosztályokból áll össze
- Feldolgozásuk
  - Java Reflection API
- Eredmény
  - Java nyelv metamodellje szerinti modell



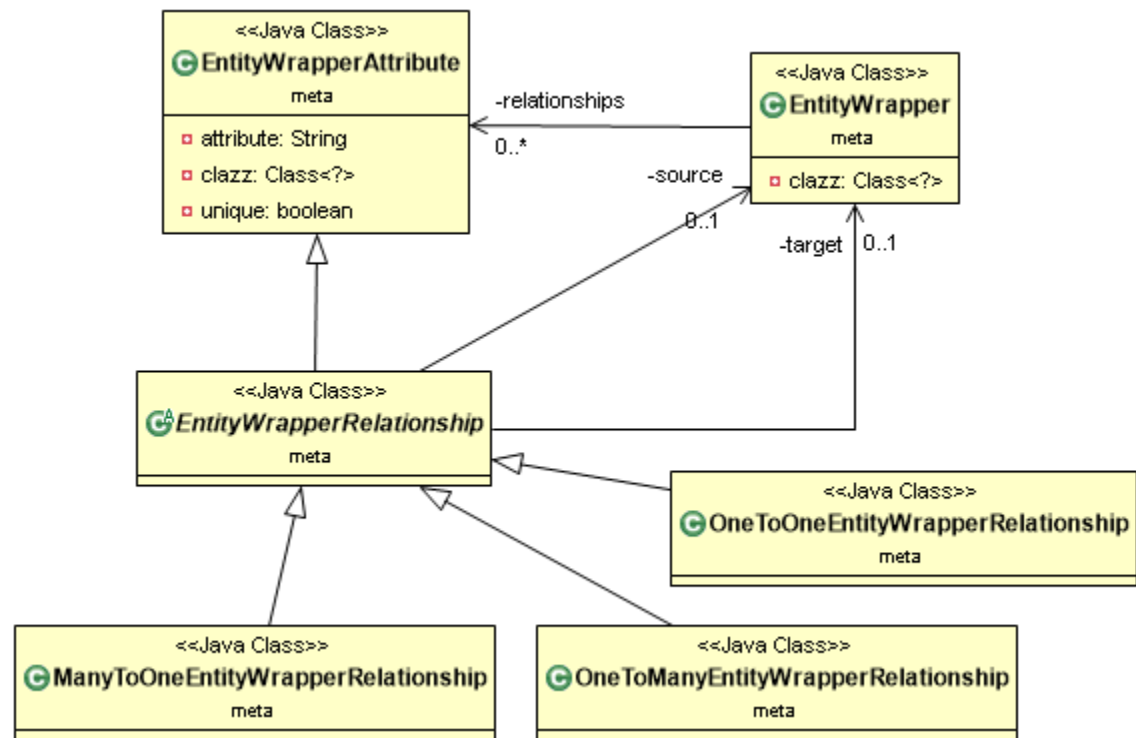
# Automatizált generálás folyamata

1. Lefordított osztályok
  - osztálykönyvtár (jar)
2. JPA adatbázismodell
  - Java nyelv metamodelleje szerint
3. Bemeneti osztályok
  - specializált metamodel (1) szerint
4. Generált objektumok
  - specializált metamodel (2) szerint
5. Absztrakt szintaxisfa
  - Java nyelv használt elemei alapján
6. Inicializáló forráskód



# Metamodel az osztályok kezeléséhez

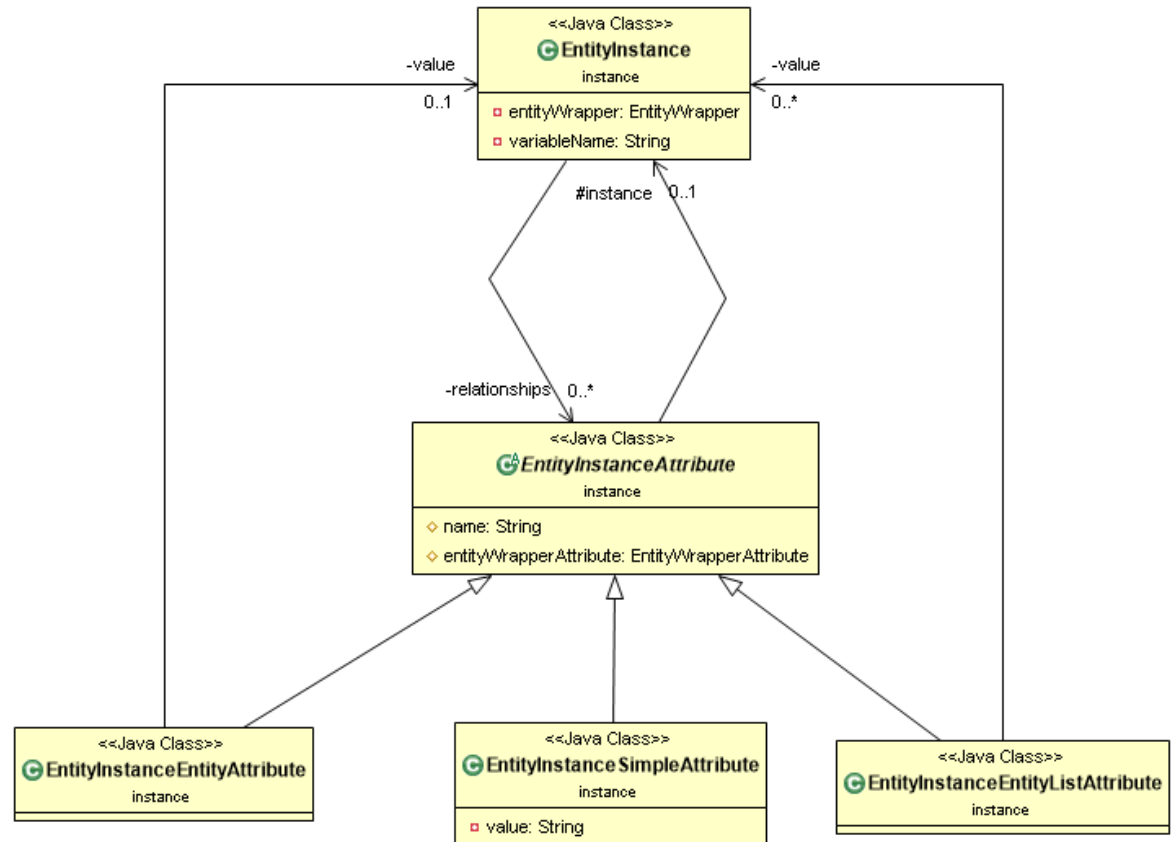
- Entitás
- Attribútum
- *Kapcsolat*
- Egy-egy
- Egy-több
- Több-egy





# Metamodel az objektumok kezeléséhez

- Entitáspéldány
- *Attribútum*
  - Egyszerű
  - EntitásLista
  - Entitás

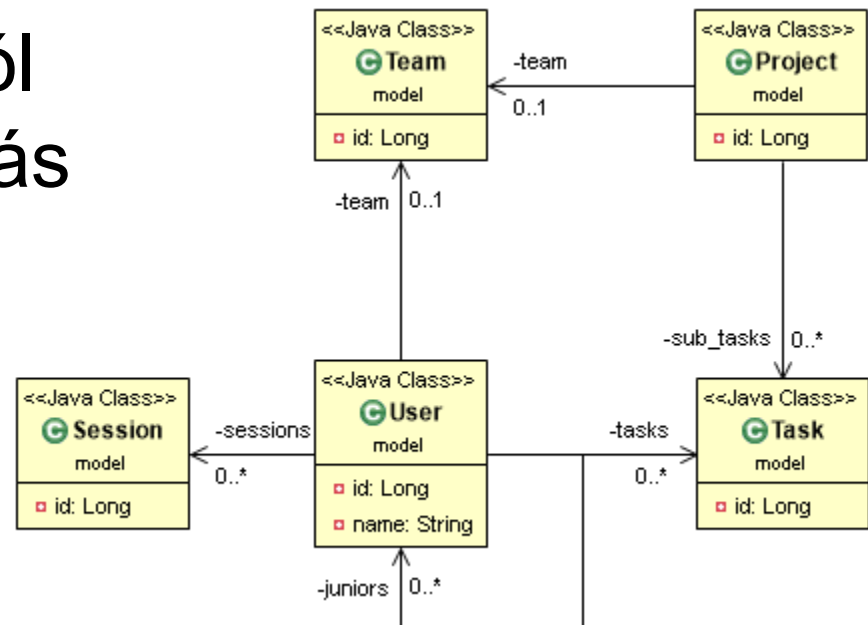


# Generálási stratégia

- **Bejárési lehetőség két szinten:**
  - osztályok,
  - objektumok.
  
- **Felelőssége:**
  - attribútumok kitöltése,
  - változónevek létrehozása,
  - generált példányok számának meghatározása,
  - kapcsolatok kialakítása.

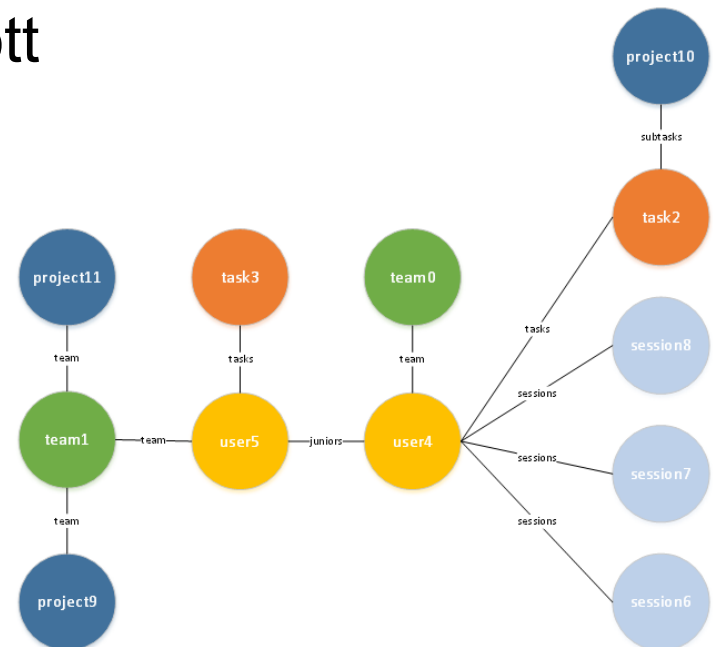
# Tesztelés - naív stratégia

- Egyszerű bemeneti modell
- Véletlenszerű generálás
  - Paraméterezhető
- Minta valós adatokból
- Referenciális integritás
- ~ Elérhető eszközök képességei



# Tesztelés - összetett stratégia

- Egyszerű bemeneti modell
- Attribútumok generálása
- Fa struktúra generálása
  - Kruskal algoritmus módosított változata
- Osztály és objektum szintű bejárás is szükséges



# Továbbfejlesztési lehetőségek

- További stratégiák fejlesztése
- Leírókból felépíthető stratégiák
  - XML, JSON, egyéb leírók.
- Grafikus felhasználói felület
  - paraméterek,
  - kimenet objektum szintű összeállítása.
- Java Data Objects (JDO) támogatása
- Adapterek más generátorokhoz
  - típusok, adatminták és megszorítások.

# Továbbfejlesztési lehetőségek

- Párhuzamosított generálás és tranzakciókezelés
- Intelligens típusfelismerés név alapján
- JUnit integráció
- Jelentések készítése
- Fejlesztőkörnyezetekhez bővítmény

**Köszönjük a figyelmet!**

Kérdések?