JPA alapú teszteléstámogatás

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

Bevezető

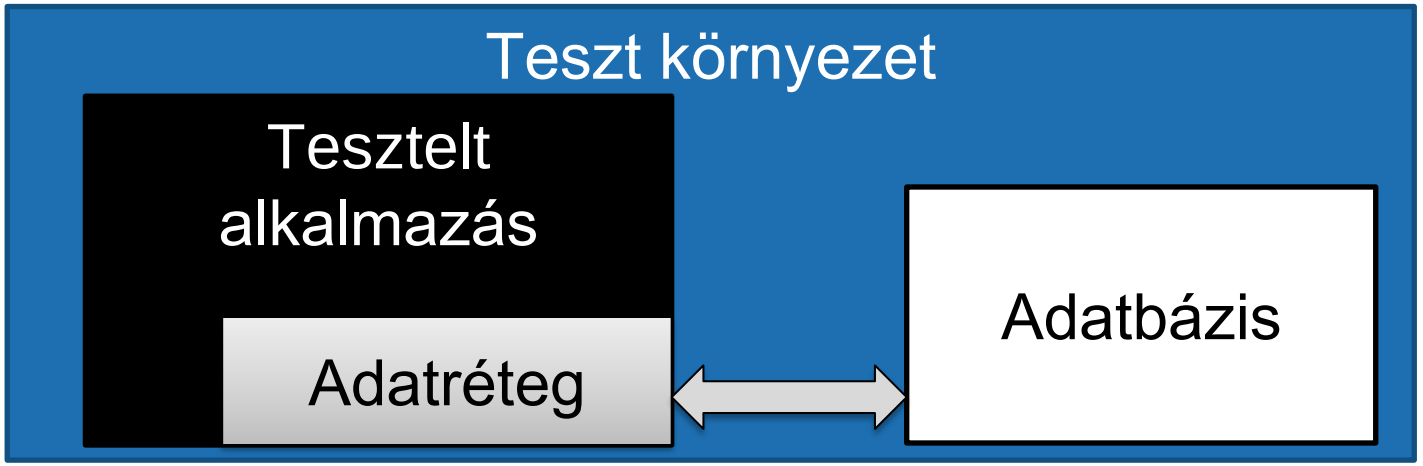
- A tesztelés jelentősége
- Az automatizált tesztelés
  - Előnyei:
    - konzisztens eredmények és adatok
    - megismételhetőség
    - könnyű karbantarthatóság
    - hatékonyabb erőforrás felhasználás
    - jelentés készítése
- Automatizált tesztadat-generálás
- A Java Persistence API szerepe

A környezet és a cél

Alkalmazás - fekete doboz

Adatréteg - szürke doboz

entitások, sémák (adattípusok) láthatóak



Cél: az adatbázist tesztadatokkal feltöltő **forráskód generálása**

Példa kimenet 1/2

```
package generated;
import model.Session;
import model.User;
import javax.persistence.EntityManager;
public class Generated {
EntityManager em = null;
public void generate() {
User user0 = new User();
User user1 = new User();
Session session2 = new Session();
Session session3 = new Session();
Session session4 = new Session();
user0.setId(0l);
```

Példa kimenet 2/2

```
user0.setName("Endre");
user0.getSessions().add(session4);
user0.getSessions().add(session2);
user1.setId(1l);
user1.setName("Szilárd");
user1.getSessions().add(session2);
user1.getSessions().add(session3);
session2.setId(2l);
session3.setId(3l);
session4.setId(4l);
em.persist(user0);
em.persist(user1);
em.persist(session2);
em.persist(session3);
em.persist(session4);
```

A probléma felvetése

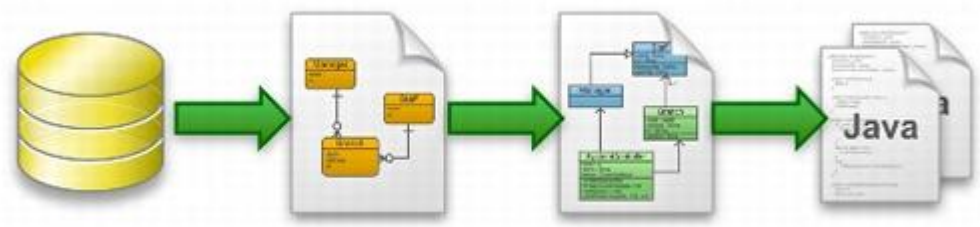
- A legtöbb eszköz véletlenszerű mintákat ad
- Elérhető szolgáltatások:
  - numerikus értékekre vonatkozó korlátok,
  - eloszlások,
  - szöveges paraméterek,
  - minta adatok használata,
  - referenciális integritás (idegen kulcs),
  - egyediség.
- Igény mélyebb összefüggésekre:
  - osztályszinten,
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

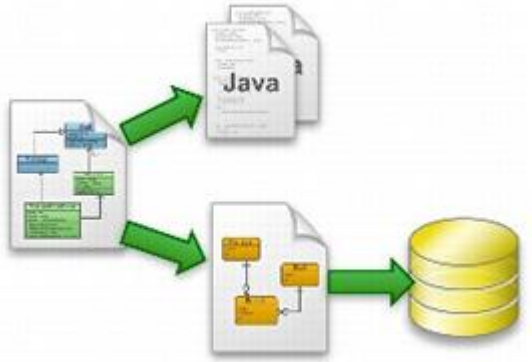
Mélyebb összefüggések

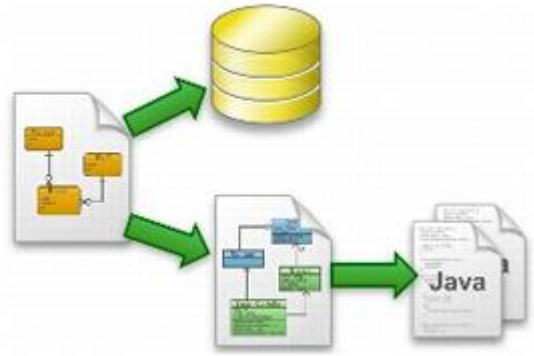
- Objektumorientált programozás szempontjából speciális esetek
- Redundáns attribútumok kitöltése
- Gráf alapú feltételek:
  - körmentesség,
  - többszörös összefüggés,
  - elkülönülő szigetek,
  - rekurzív adatstruktúrák.

Adatbázismodell elemzése

- Relációs adatbázis és objektumorientált modell kapcsolata

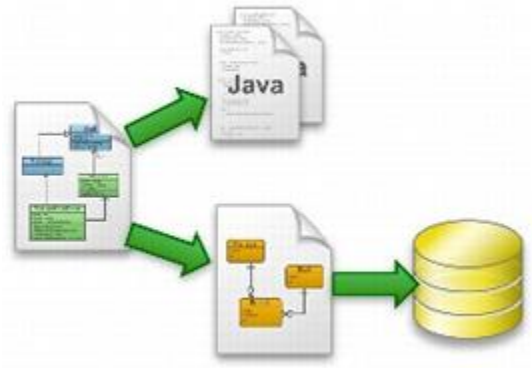






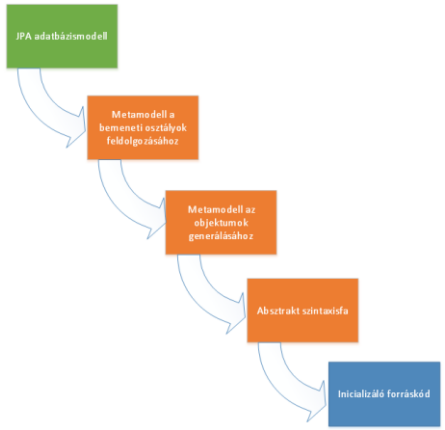
Adatbázismodell elemzése

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



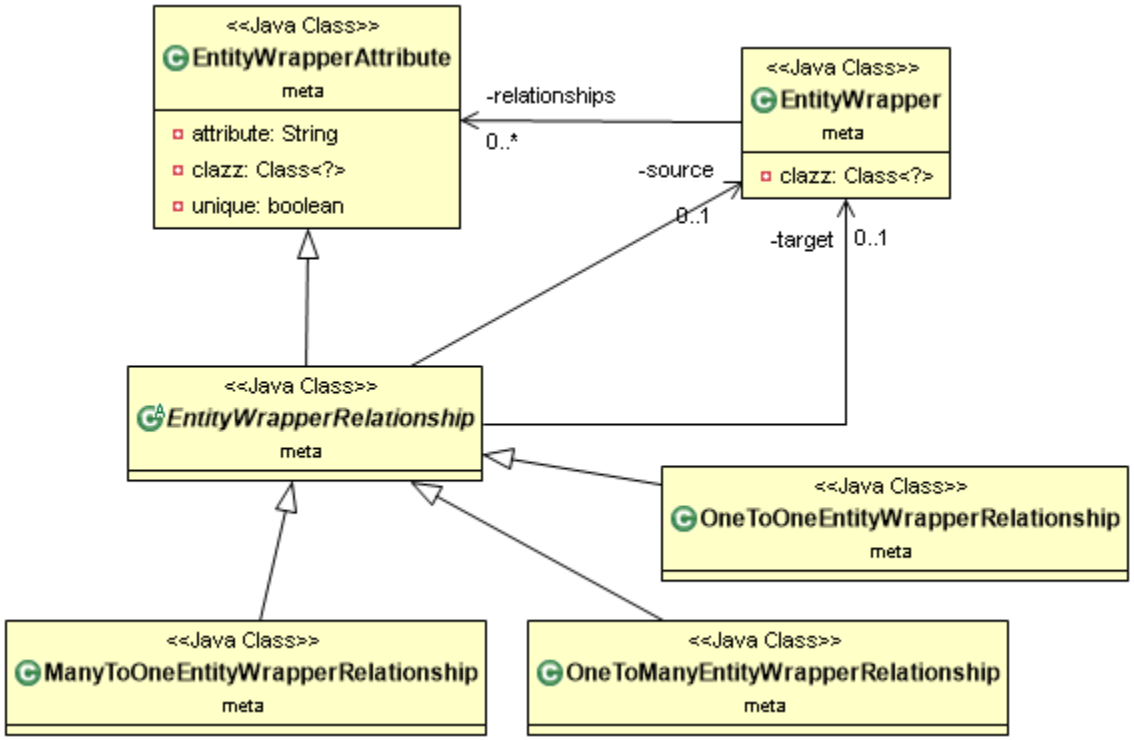
Automatizált generálás folyamata

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



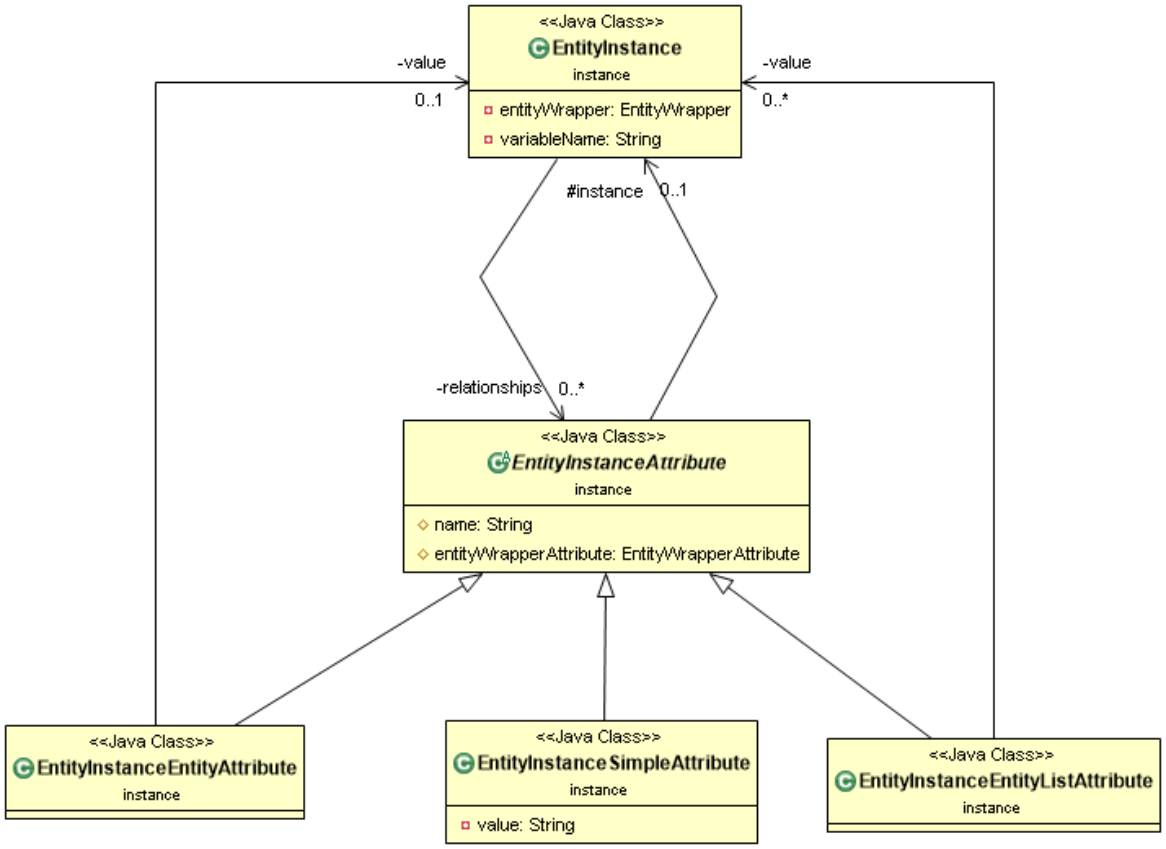
Metamodell az osztályok kezeléséhez

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



Metamodell az objektumok kezeléséhez

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



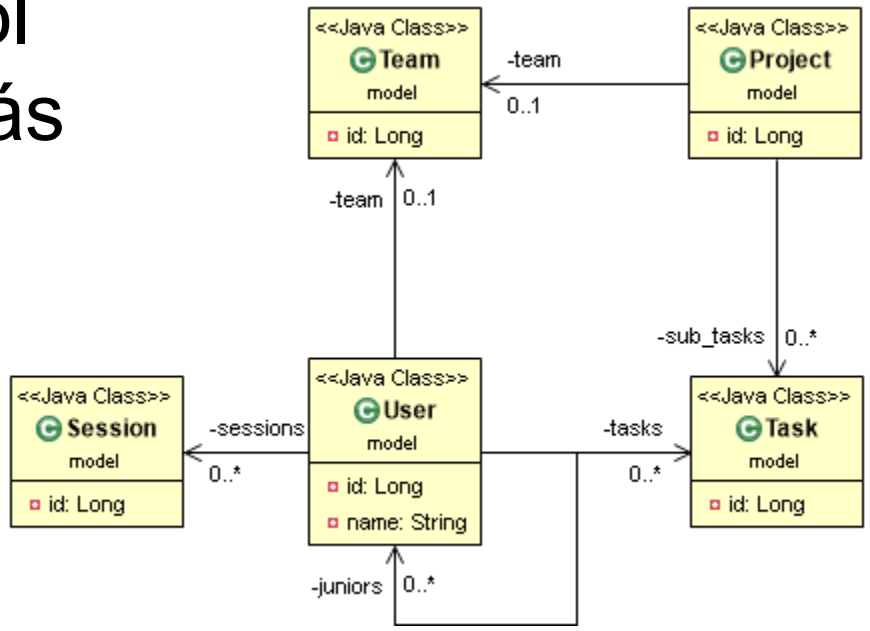
Generálási stratégia

- Bejárási 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

Köszönjük a figyelmet!

Kérdések?