

# *Slovenská technická univerzita v Bratislave*

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

---

## Databázová aplikácia (s O/R mapovačom)

Dokumentácia v2

Predmet: Databázové systémy

Autor: Ivan Gulis

Študijný program: Informatika

Semester: 4.

Ak. rok: 2014/2015

## Zadanie: Databáza komixovej ságy

Tvorba komixovej ságy zahŕňa zálohovanie a prácu so všetkými zložkami komixu. Vo firme, ktorá tvorí komixy sú zamestnaní umelci – autori, ktorí kreslia postavičky. Tu sa vyžaduje evidencia zamestnancov: ich mien, odkedy sú zamestnaní a ich osobné čísla. Každá postava je originálne kreslená jedným autorom, aby mala vždy svoj vlastný, ručne kreslený štýl. Tiež musí mať vlastné meno, ID a typ (napr. démon, ninja...). Tieto postavy sa potom kreslia na stránky komixu. Všetky existujúce komixy musia byť uložené v databáze a rozlišovať sa číslom komixu. Každý komix má mať vlastný názov a dátum vytvorenia. Do jednotlivých komixov je ďalej potrebné vkladať stránky, a do stránok kresliť existujúce postavy a písať bubliny s textom. Na každej stránke by mal byť krátky opis situácie.

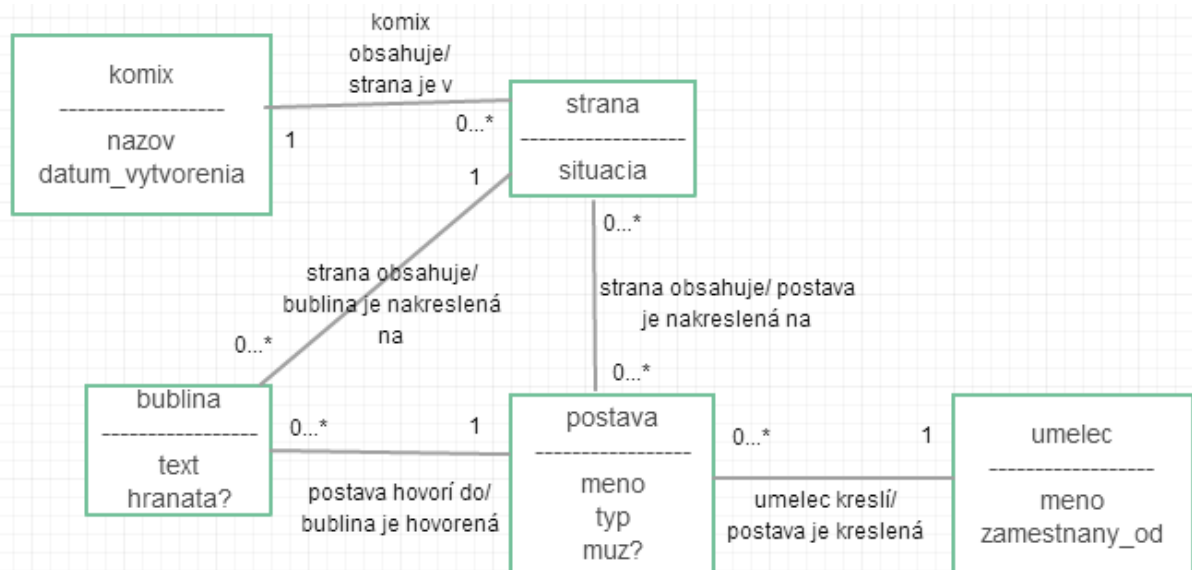
## Scenáre:

Aplikácia rozoberá 8 scenárov (jednoduchšie aj zložitejšie);

1. Scenár slúži na vytvorenie novej postavičky. Zahŕňa zobrazenie tabuľky už vytvorených postáv, textové polia pre meno a typ postavy, a možnosť zvoliť pohlavie. Nakoniec je potrebné vybrať autora, ktorý danú postavičku bude kresliť. Autori sú zobrazení v ComboBoxe. Nakoniec je potrebné potvrdenie tlačidlom.
2. Scenár slúži na zmenu názvu existujúceho komixu. Komixy sú zobrazené v ComboBoxe. Po výbere komixu je potrebné vyplniť nový názov a update potvrdiť. Tiež sa tu nachádza tabuľka už vytvorených komixov.
3. Scenár slúži na vyhodenie jedného z autorov. Je tu zobrazená tabuľka zamestnancov, ktorí kreslia nejakú postavičku. V 2 ComboBoxoch sú autori, z ktorých je potrebné vybrať autora, ktorí dostane výpoveď, a autora, ktorí bude ďalej kresliť jeho postavičky. Ak autor nekreslí žiadnu postavičku, 2. ComboBox nie je podstatný.
4. Scenár vypíše štatistiku, ktorý autor kreslí koľko mužských a koľko ženských postáv. Výpis je formou tabuliek a na konci je vypísaný celkový súčet existujúcich postáv.
5. Scenár slúži ako search. V ComboBoxe sú zobrazené existujúce postavičky. Po vybraní jednej z nich a potvrdení sa zobrazí konkrétny autor, ktorý danú postavičku kreslí, jeho ID, a typ postavičky.
6. Scenár slúži na insert a delete celého komixu, a na pridávanie strán do vybraného komixu. Po zadaní názvu komixu sa komix pridá do vytvorenej tabuľky a ComboBoxu, ktorý slúži pre označenie komixu, do ktorého sa budú pridávať stránky alebo sa celý zmaže. Ako dátum vytvorenia sa používa súčasný dátum.
7. Scenár zaobstaráva príjem nových zamestnancov, budúcich autorov. Je tu textové okno pre vyplnenie mena a tabuľka už zamestnaných pracovníkov.
8. Scenár nakreslí postavičku na stranu. Sú tu 3 ComboBoxy, najskôr je potrebné vybrať komix, po potvrdení sa objavia jeho strany. Nakoniec sa vyberie postavička, ktorá sa má nakresliť a vyplní sa krátky opis činnosti („čo postavička práve robí?“). Tabuľka je zobrazená len pre stránky vybraného komixu.

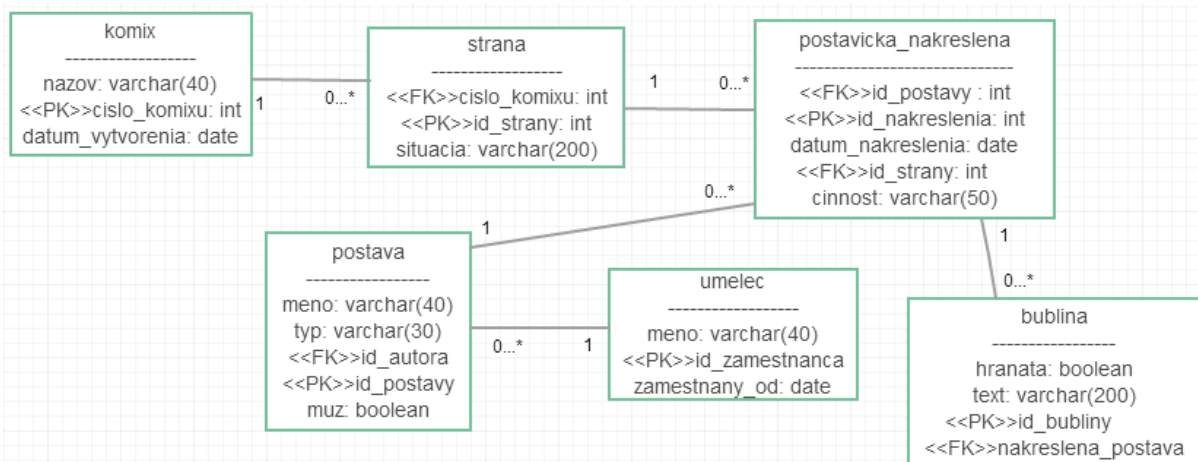
## Diagramy:

Logický model:



Entity komix, strana, bublina, postava, umelec. Každý komix má určitý počet strán a každá strana patrí jednému komixu. Na strane sa nachádza určité množstvo nakreslených postáv a bublín. Každá bublina je nakreslená len na jednej strane, ale každá postava môže byť nakreslená na viacerých stranách. Jedna postava za svoju existenciu použije veľa bublín na rozprávanie, no každá bublina je hovorená len jednou postavou. Každý umelec kreslí svoje postavičky, no každá postava je kreslená práve jedným umelcom.

Fyzický model:

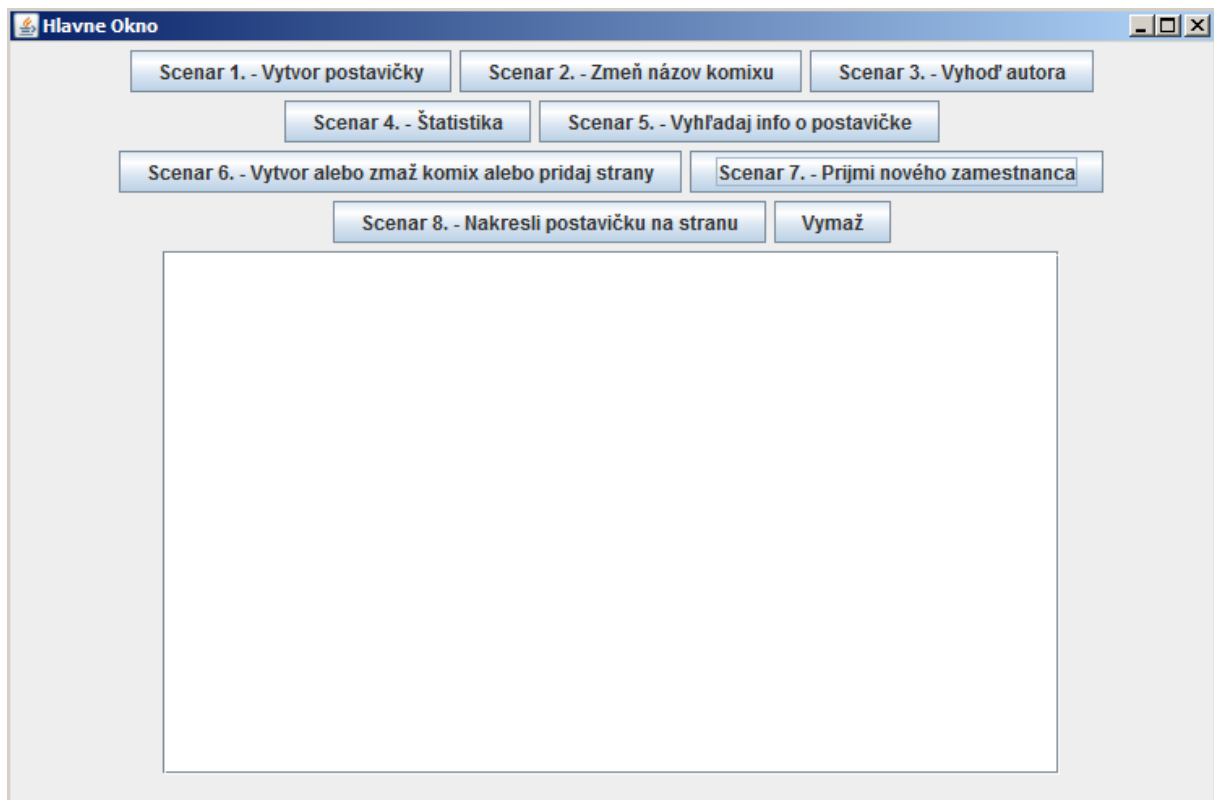


Komix má atribúty názov, primárny kľúč a dátum vytvorenia. Strana má cudzí kľúč rovný primárnemu komixu. Okrem toho má svoj primárny kľúč a varchar na opis situácie. Umelec má svoje meno, primárny kľúč a dátum, od kedy je zamestnaný. Vznikla medzientita postavicka\_nakreslena, ktorá spája strany a postavy (obsahuje cudzie kľúče pre stranu a postavu). Varchar „cinnost“ slúži na krátky opis činnosti, ktorú nakreslená postavička vykonáva. Ku každému nakresleniu sú ďalej priradené bubliny. Jedna nakreslená postava ich môže povedať viac (často 1veta 1 bublina). Bublina obsahuje text, svoj primárny kľúč, cudzí

klúč nakreslenia a boolean či je hranatá. Postava obsahuje cudzí klúč od svojho autora, svoj primárny klúč, meno, typ (ktorý môže byť ľubovoľný od zvierat po príznaky ľudí) a boolean, či je daná postava muž.

## Opis návrhu a implementácie:

Ako prostredie som zvolil Eclipse, s jazykom Java. Aplikácia má jednoduché GUI s hlavným oknom, a každý scenár si ďalej otvára vlastné okno.



Do databázy sa pripájam funkciou pripojSa(), ktorú volám v maine. Connection získavam cez funkcie z Hibernate, a celú inicializovanú premennú SessionFactory mám uloženú v hlavnej triede.

```
public static void pripojSa(){  
    ...  
    Configuration configuration = new Configuration();  
    configuration.configure("hibernate.cfg.xml");  
    StandardServiceRegistry serviceRegistry = new  
    StandardServiceRegistryBuilder().applySettings(  
        configuration.getProperties()).build();  
    sessionFactory = configuration.buildSessionFactory(serviceRegistry);  
    ...  
}
```

K databáze ďalej prístupujem cez Session a Transaction. Pre prístup k dátam vytváram kritériové dopyty a vytváram listy:

```
Session session = sessionFactory.openSession();  
Transaction ts = null;  
try {  
    ts = session.beginTransaction();  
    Criteria crit = session.createCriteria(Postava.class);  
    List<Postava> postavy = crit.list();  
    ts.commit();  
}
```

```

} catch (Exception e) {
ts.rollback();
} finally {
session.close();
}

```

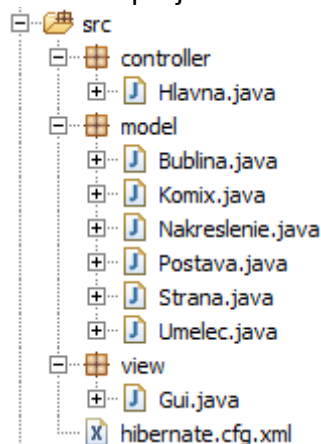
Nakoniec commit, prípadne rollback a session.close,

## Mapovanie

Pre konfiguráciu je využitý „hibernate.cfg.xml“

V tomto xml súbore sú prihlasovacie údaje (a iné potrebné informácie) pre databázu, a zoznam class, ktoré sa majú namapovať. Všetky classy sú mapované cez anotácie. Použité sú napr. @column, @id, @OneToMany, @ManyToOne, a pri primárnych kľúčoch využívam typ Identity.

Štruktúra projektu:



Mapované classy sú v modeli.

## Scenáre implementácia:

Príklad vrátenia listu cez session:

```

Criteria crit = session.createCriteria(Postava.class);
list1 = crit.list()

```

Jednotlivé scenáre sú implementované vo funkciách hlavnej triedy, a volané do GUI.

1. Scenár využíva session k vráteniu objektov postáv a autorov pre tvorbu tabuľky, a ComboBoxu existujúcich autorov. Pre insert sa využíva session.save().
2. Scenár v session vracia objekty komixov a vkladá ich do ComboBoxu a tabuľky. Pre update sa zvolí konkrétny komix a nastaví sa mu nový názov.
3. Scenár pomocou session vracia autorov a dáva ich do ComboBoxov a do tabuľky. Pri deletovaní zamestnanca sa jeho meno z argumentu funkcie použije na nájdenie správneho autora a vykoná sa session.delete(). Pred tým sa ale presunú postavy, ktoré autor kreslil náhradnému autorovi.
4. Scenár sa skladá zo session, ktoré vráti všetkých autorov, a z atribútov sa vyplnia 2 tabuľky a textové pole, pre výpis daných štatistík. Listy postáv autorov sa prechádzajú vo for cykle. Využitý je napr. addOrder pre id zamestnancov.
5. Scenár využíva session pre vrátenie listu na vyplnenie ComboBoxu postavičiek a výslednej tabuľky. Najskôr sa vyhľadá konkrétna postavička a z nej sa vyberie jej autor.

6. Scenár je zložitejší. Pre insert nového komixu sa využíva obsah textového pola. Všetky komixy sa hneď sez session vracajú do tabuľky a ComboBoxu komixov. Z názvu vybraného komixu sa nájde konkrétny komix a vytvorená nová stránka sa mu pridá do listu. Je tu možnosť aj vybraný komix zmazať (kde sa využíva `session.delete()`).
7. Scenár má textové pole pre meno zamestnanca, a cez session sa vyberú zamestnanci, ktorí sa pridajú do tabuľky. Po stlačení tlačidla sa pridá nový zamestnanec (pomocou `session.save()`).
8. Scenár využíva session pre vrátenie listu nakreslení a listu komixov a stránok. (Pre ComboBoxy všetkých komixov, a pre všetky strany vybraného komixu.) V tabuľke sú zobrazené nakreslenia postavičiek na stranách len vybraného komixu. Následne sa tlačidlom vytvorí nové nakreslenie, ktoré sa cez `session.save()` uloží.