

Programovanie, algoritmy, zložitosť / ÚINF/PAZ1C

PAZ1C

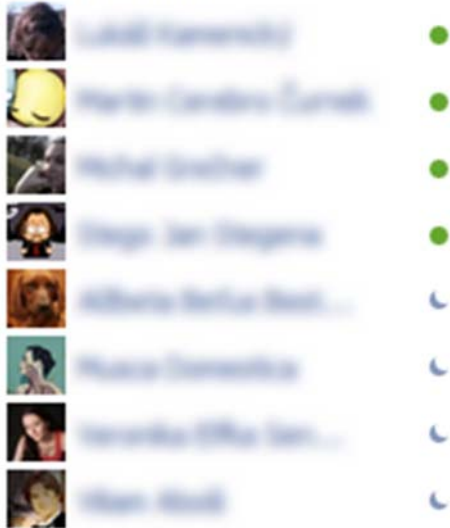
Róbert Novotný
robert.novotny@upjs.sk

27. 10. 2010

Príklad: aplikácia pre sociálnu sieť

PAZ1C

Friends Online



Chcem, aby to
bolo ako
Facebook, ale
červené!



Ako navrhovať komplexné aplikácie?

PAZ1C

- komplexné aplikácie sú delené na vrstvy

Prezentačná vrstva

- používateľské rozhranie
- zobrazenie a získavanie informácií od používateľa

Vrstva aplikačnej logiky

- biznis logika
- príkazy, rozhodovania, spracovanie dát

Databázová vrstva

- úložisko dát
- typicky relačná databáza/súborový systém

three-tier
application

Ako navrhovať komplexné aplikácie?

PAZ1C

Fakt musím mať všetky tri vrstvy?

- **single-tier** application
- „všetko na hromade“
- klasické skripty / pascalovské súbory / úlohy z PAZ1c
- aplikácia načíta zo súboru, požuje dáta, vyplúje ich

Dvojvrstvová: two-tier application

PAZ1C

Prezentačná vrstva + aplikačná logika

- používateľské rozhranie
- zobrazenie a získavanie informácií od používateľa priamo prepojená s biznis logikou

Databázová vrstva

- úložisko dát
 - typicky relačná databáza/súborový systém
- dáta bývajú často na samostatnom stroji

Výhody trojvrstvovej architektúry

PAZ1C

- možno mať **viacero druhov** používateľských rozhraní
 - kto by nechcel AIS s prístupom cez mobil?
- aplikačná logika je **nezávislá od miesta** uloženia dát
 - čo ak zrazu potrebujeme ukladať dáta inak?
 - čo ak zrazu potrebujeme vylepšenú databázu?
- každá vrstva môže byť v prípade potreby modifikovaná / nahradená
- každá vrstva môže byť realizovaná na **samostatnom stroji**

Rady starej matere

PAZ1C



1. navrhujte jadro aplikácie tak, aby bolo **nezávislé** od používateľského rozhrania
2. navrhujte jadro aplikácie tak, aby bolo **nezávislé** od spôsobu uloženia dát

Viacvrstvové aplikácie v Java

PAZ1C

- interfejsy jedným z prostriedkov pre návrh viacvrstvových aplikácií
- v každej vrstve budeme mať:
 - **interfejsy** = operácie s príslušnou vrstvou
 - **objekty** = prenášajú dáta medzi vrstvami



Príklad: sociálna sieť

PAZ1C

V sociálnej sieti chcem mať svoj profil s menom, e-mailom a fotkou. Chcem mať možnosť zverejňovať statusy (krátke texty do 160 znakov). Okrem toho chcem mať zoznam priateľov, ktorí budú obdivovať moje statusy, i naopak – ja budem môcť vidieť všetky statusy mojich priateľov.

Navrhujeme triedy: metóda podstatných mien

PAZ1C

sociálna sieť, profil, meno, e-mail, fotka, status,
zoznam priateľov

- ~~sociálna sieť~~: náš systém
- **profil**:
 - meno (String)
 - e-mail (String)
 - fotka (obrázok)
- **zoznam priateľov**
 - = zoznam profilov
 - jeden profil má zoznam profilov, ktorí sú jeho priateľmi
- **status**

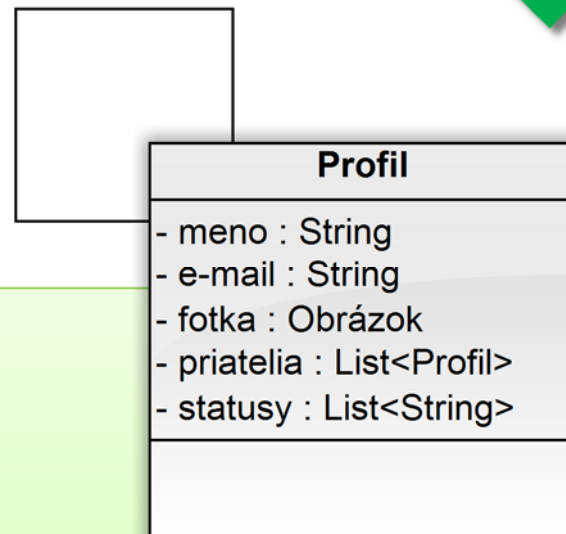
Navrháme triedy!

PAZ1C

- čo so **statusmi**?
 - status: String
- profil môže niesť zoznam statusov

```
public class Profil {  
    private String meno;  
    private String e-mail;  
    private Obrázok fotka;  
    private List<Profil> priatelia  
        = new ArrayList<Profil>();  
    private List<String> statusy  
        = new ArrayList<String>();  
    //...  
}
```

zoznam priateľov



Schopnosti profilu?

PAZ1C

- pridaj priateľa
- odober priateľa
- publikuj status

Vytvorme interfejs!

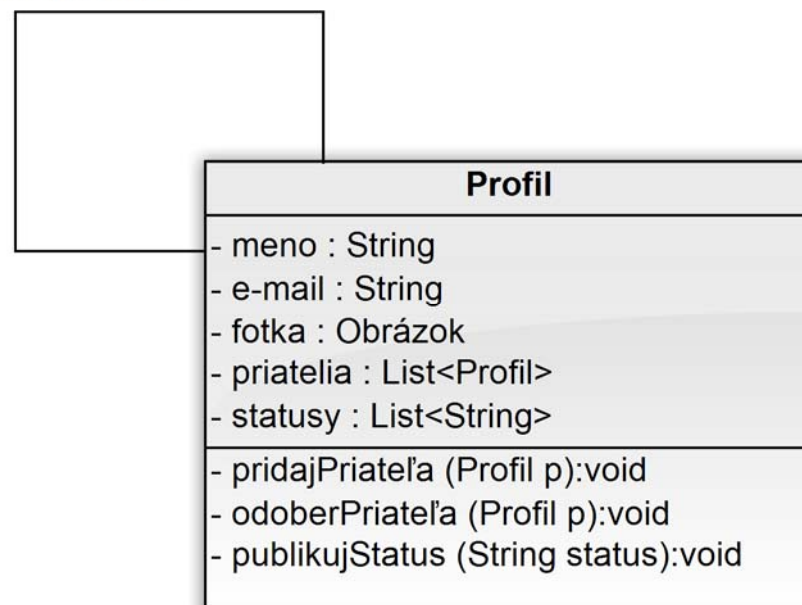
```
public interface Profil {  
    void pridajPriateľa(Profil p);  
    void odoberPriateľa(Profil p);  
    void publikujStatus(String status);  
}
```

V akom sme stave?

PAZ1C

- máme interfejs **Profil** i triedu **Profil**
- nič nám nebráni ich **zlúčiť**
 - **KISS**: keep it simple, stupid
- v skutočnosti sme sa mali súčasne zamýšľať nad schopnosťami i stavom

zoznam priateľov



Napíšme si test!

PAZ1C

- test je programátorovo GUI
- ukazuje použitie tried a metód a ľahkú použiteľnosť

```
Profil bill = new Profil("Bill Gates");  
//...  
Profil steve = new Profil("Steve Jobs");  
//..  
bil.pridajPriatela(steve);  
steve.pridajPriatela(bill);  
//..  
for(Profil gp : bill.getPriatelie()) {  
    System.out.println(gp);  
}
```



Otázky na zamyslenie

PAZ1C

- naozaj stačí na status len text?
 - čo ak chcem evidovať dátum publikovania?
- čo ak má niekto trilión priateľov?
- a chrlí každý deň trilión statusov?
- vzťah „je priateľom“ by mal byť symetrický
 - ak sa Bill stane priateľom Steva, Steve sa stane priateľom Billa

...a ešte ďalšie otázky na zamyslenie

PAZ1C

- ako **vyhládam** konkrétny profil?
 - ako získam konkrétny profil do databázy?
- ako **zobrazím všetky** profily?
- ako môžem založiť **nový** profil a udržiavať ho v dátovom úložisku?
 - dátové úložisko: databáza, súbor...

CRUD!

Profil je entita

PAZ1C

- **profil** = je jedna zo základných tried v našej doméne
 - **doména**: oblasť, ktorú sa snažíme modelovať v informačnom systéme
- každý **profil** je jednoznačne identifikovaný
 - v našom systéme to zatiaľ nemodelujeme
 - predpokladáme, že profil je identifikovaný menom a mailom
- inštancia profilu putuje systémom

Entita!

Štyri základné operácie s entitami

PAZ1C

C-reate

- vytvorenie novej inštancie entity

R-ead

- načítanie podľa kritérií

U-pdate

- aktualizácia entity

D-elete

- odstránenie

CRUD!

Ako to vyriešiť v Java?

PAZ1C

Data Access Object [DAO]

- vytvoríme interfejs, ktorý bude obsahovať metódu pre každú z CRUD operácií!

```
public interface ProfilDao {  
    Profil najdiProfil(Integer id);  
    void ulož(Profil p)  
    List<Profil> najdiPodľaPríkladu(Profil p);  
    void odstráň(Profil p);  
}
```

Data Access Objects

PAZ1C

- **ProfilDao** je len sada operácií
 - kontrakt!
- predstavuje interfejs („rozhranie“) k databázovej vrstve
 - v skutočnosti môže byť databázou aj súbor
- potrebujeme vytvoriť triedu, ktorá bude hovoriť **ako** sa súbor uloží do dátového uložiska

Konkrétne objekty DAO

PAZ1C

```
public class SúborovýProfilDao
    implements ProfilDao
{
    public Profil najdiProfil(Integer id) {
        /* otvoríme súbor, nájdeme záznam s
           daným ID, vytvoríme inštanciu Profilu
        */
    }
    void ulož(Profil p) {
        /* otvoríme súbor, zapíšeme na koniec údaje z
           profilu
        */
    }
}
```

Ako získame inštanciu DAO?

PAZÍC

- ak máme swingovské okno **ProfilForm** upravujúce profil používateľa, ako môžem **uložiť** inštanciu **Profilu**?
- využijeme **kompozíciu**!
 - ProfilForm **potrebuje** DAO = HAS-A

```
public ProfilForm extends JDialog {  
    private ProfilDao profilDao;  
}
```

Odkiaľ získame inštanciu DAO?

PAZ1C

- návrhový vzor **Factory**
 - **továreň**, ktorá vyprodukuje inštanciu DAO
- návrhový vzor **Singleton**
 - **jedináčik**: trieda, ktorá má v celom systéme vždy jedinú inštanciu

```
public ProfilForm extends JDialog {  
    private ProfilDao profilDao =  
        ProfileDaoFactory.getProfilDao();  
}
```

Továreň!

static = premenná, ktorej obsah zdieľajú VŠETKY inštancie danej triedy

```
public class ProfilDaoFactory {  
    private static ProfilDao profilDao;  
  
    public ProfilDao getProfilDao() {  
        return profilDao;  
    }  
}
```

```
static {  
    profilDao = new ProfilDao();  
    /** inicializácia profilDao **/  
}
```

static blok = vykoná sa raz pri načítavaní prvej inštancie triedy

Vrstvová architektúra

PAZ1C

- prístup k databáze je oddelený od aplikačnej logiky
- a je **oddelený** od používateľského rozhrania
- trieda implementujúca interfejsy DAO sú navyše čierna skrinka
- a možno ich **nahrádzať** inými implementáciami