# údajozákladne, medzitváre, ux

@RoboNovotny

UINF/PAZ1c

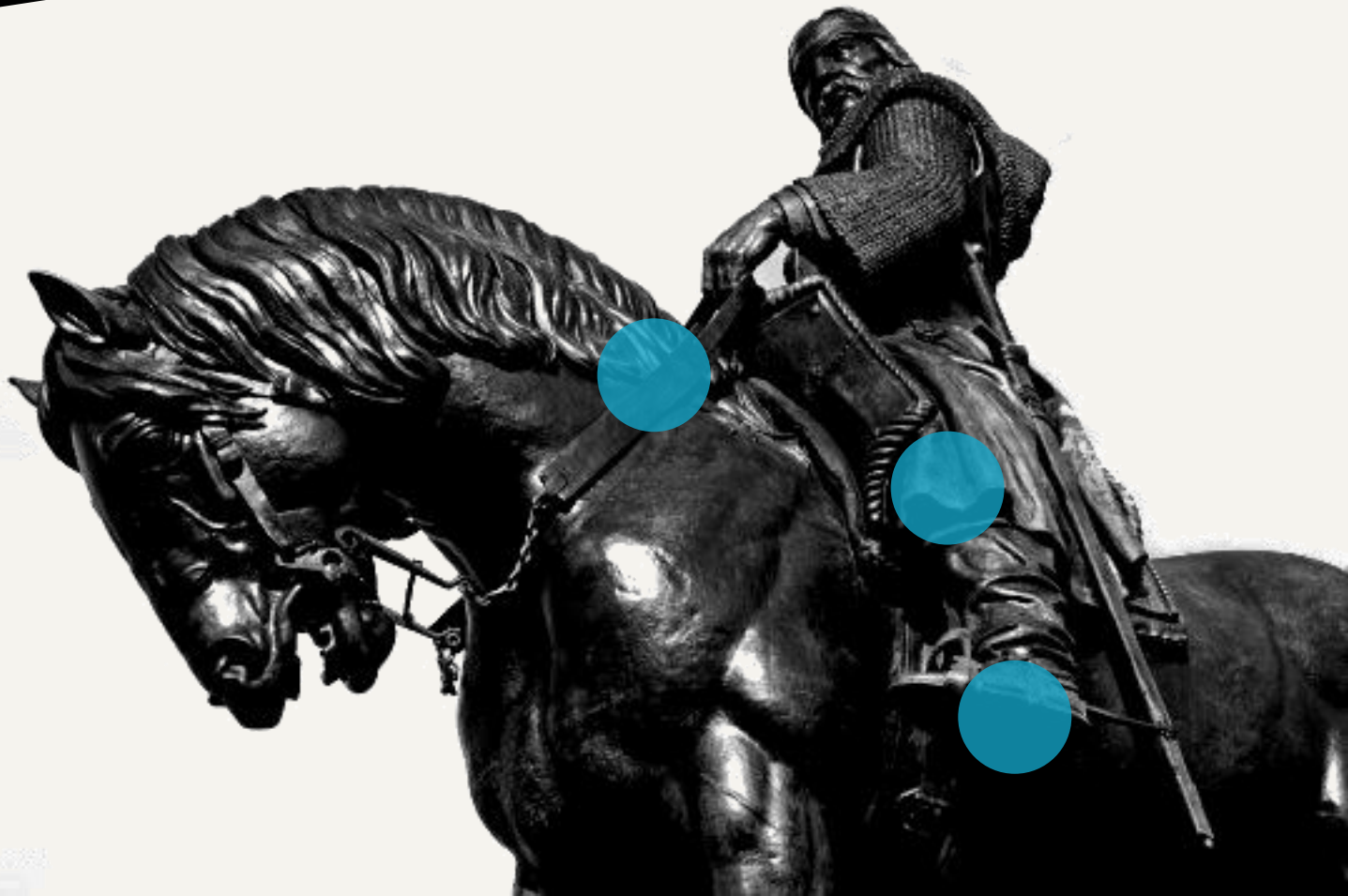
epizóda 3

30/sept/15

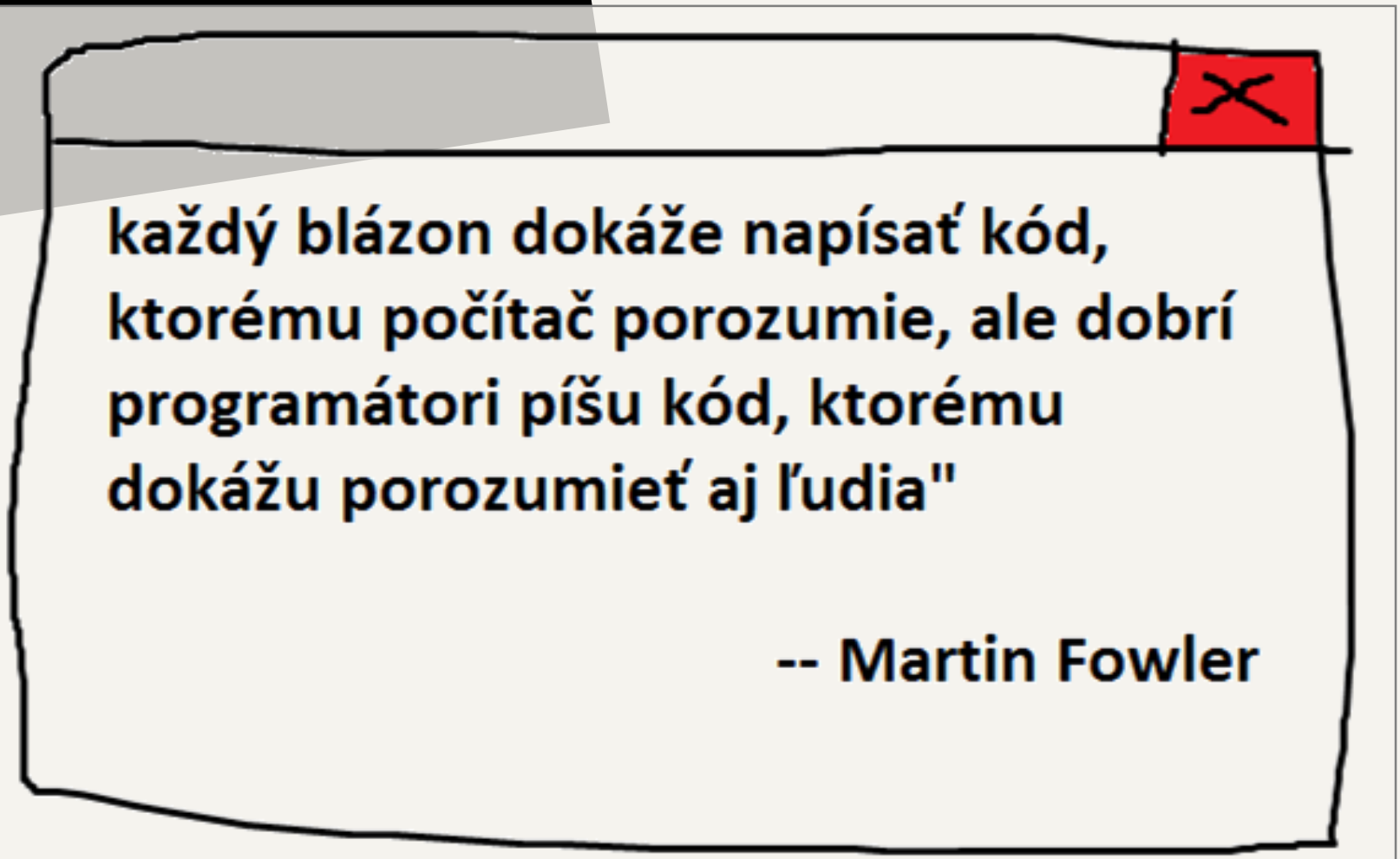
# plán na dnes

- používateľské rozhrania
- závislosti medzi triedami
- kontrakty pre triedy
- SQL a Java
- skladanie projektu z kociek

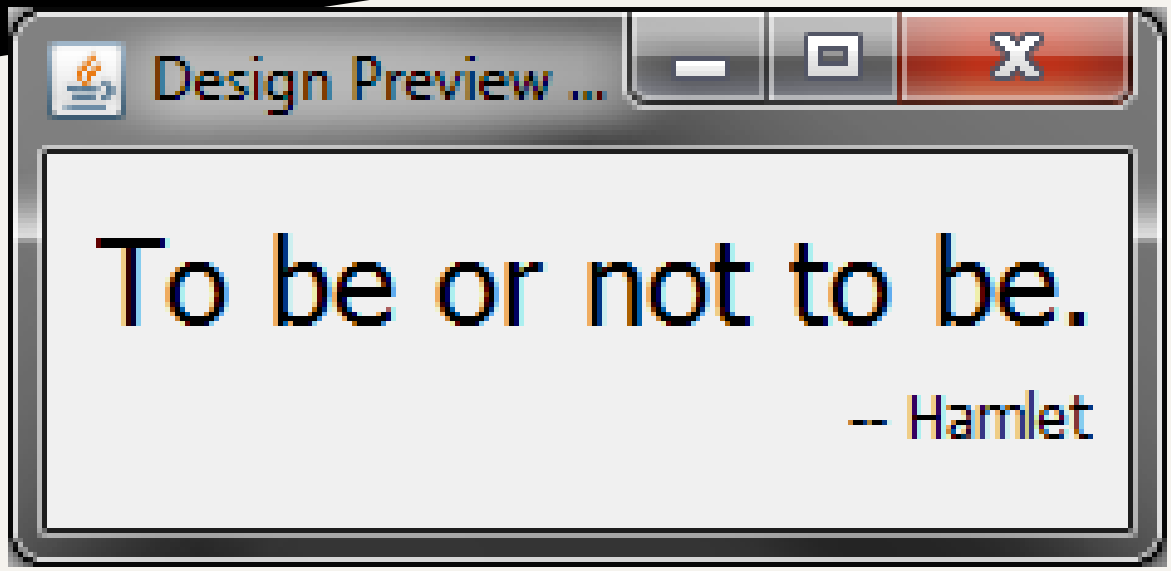
# UX / UI



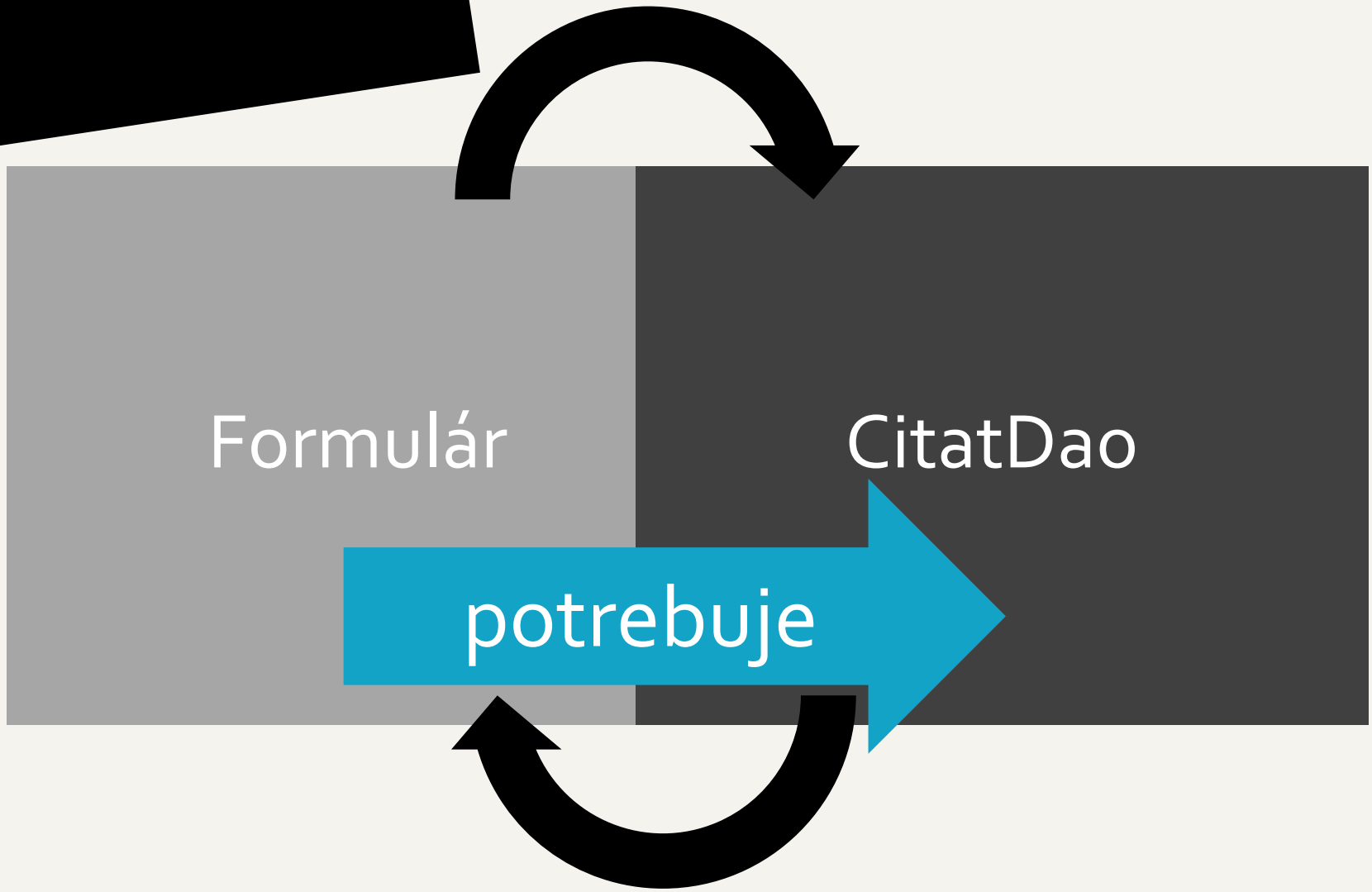
# Návrh UI


každý blázon dokáže napísať kód,
ktorému počítač porozumie, ale dobrí
programátori píšu kód, ktorému
dokážu porozumieť aj ľudia"
-- Martin Fowler

# Windows UI



# prepojenie GUI a DAO



# dependency

objekt vyžaduje pre fungovanie iný objekt

kolaborácia

závislosť

# kód: inštančná premenná

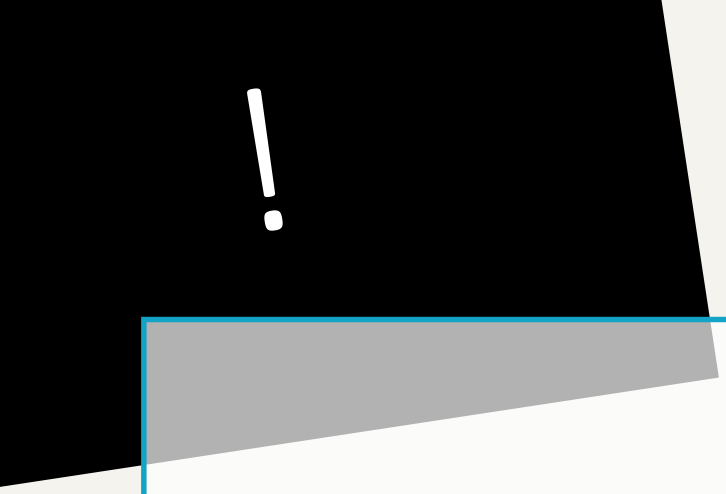
```
class HlavnyFormular {
  private CitatDao citatDao;
    ...
}
```

```
class CitatDao {
    ...
}
```

# kód

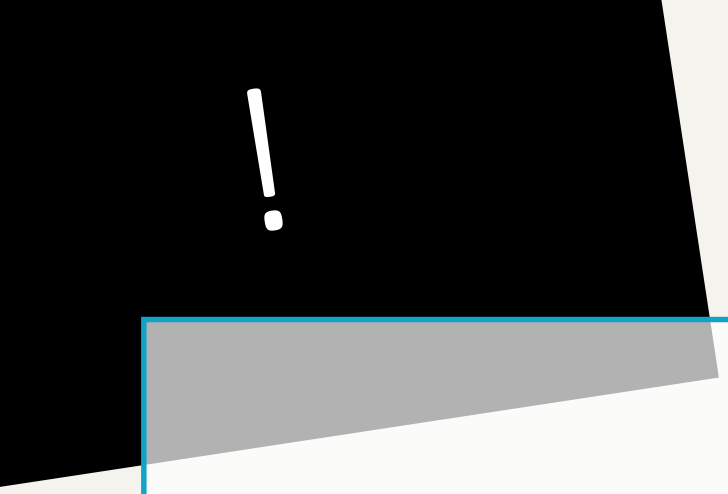
```
class HlavnyFormular {
  private CitatDao citatDao = new CitatDao();
   ...
}
```

trieda si vytvorí závislosť sama



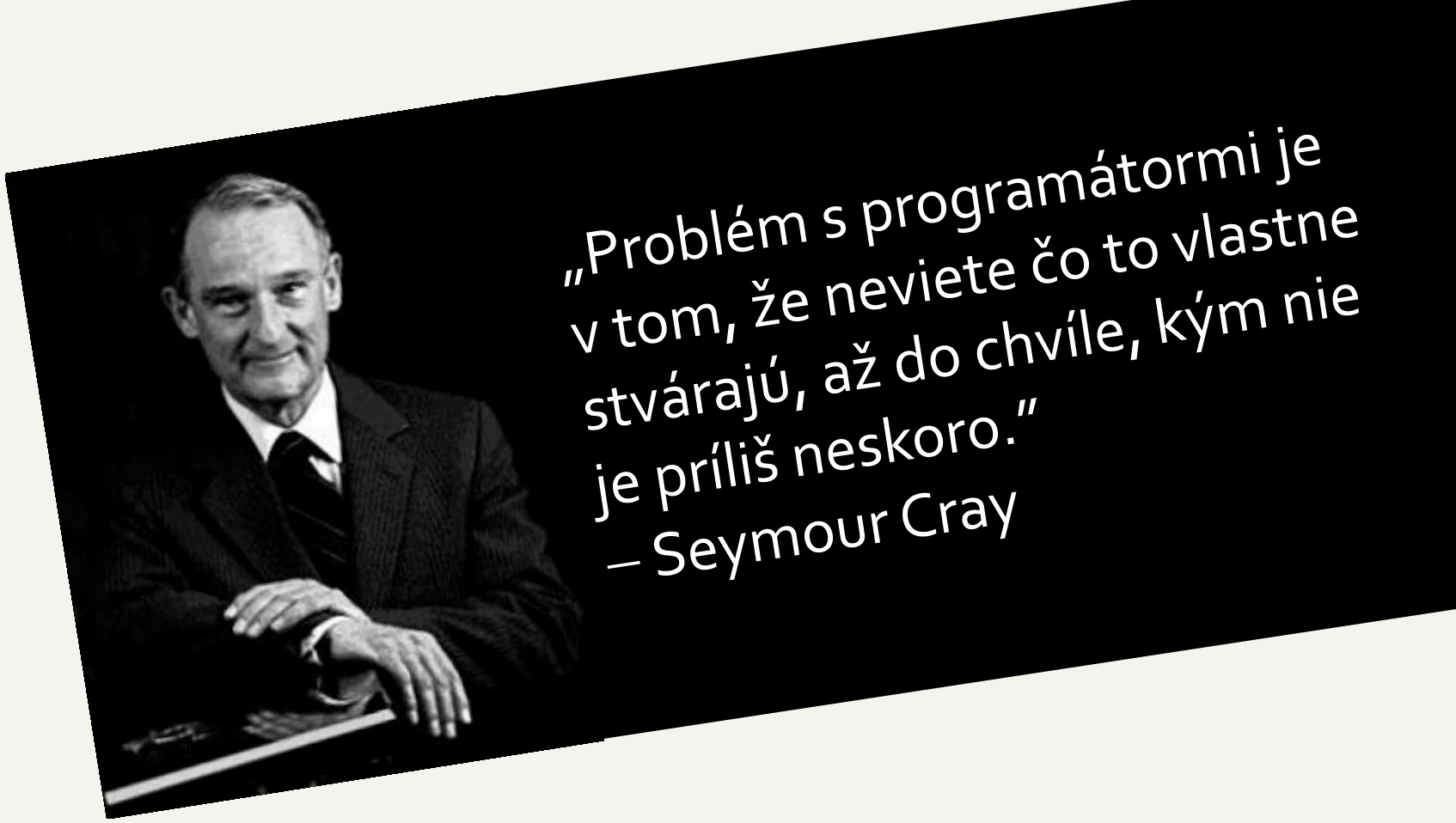
gettre a settre netreba

setter na prebíjanie implementácie



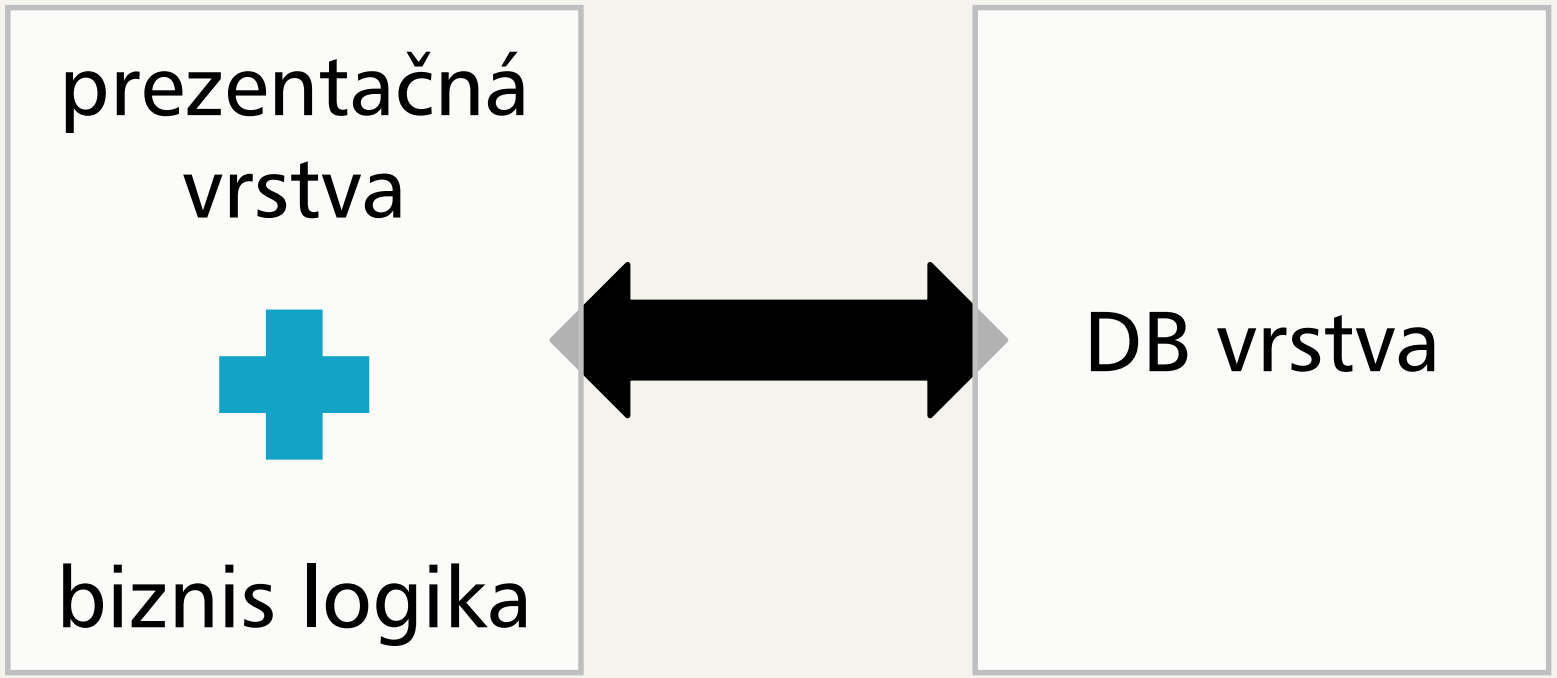
trieda sa zadrôtuje s implementáciou

čo ak chceme implementáciu meniť?

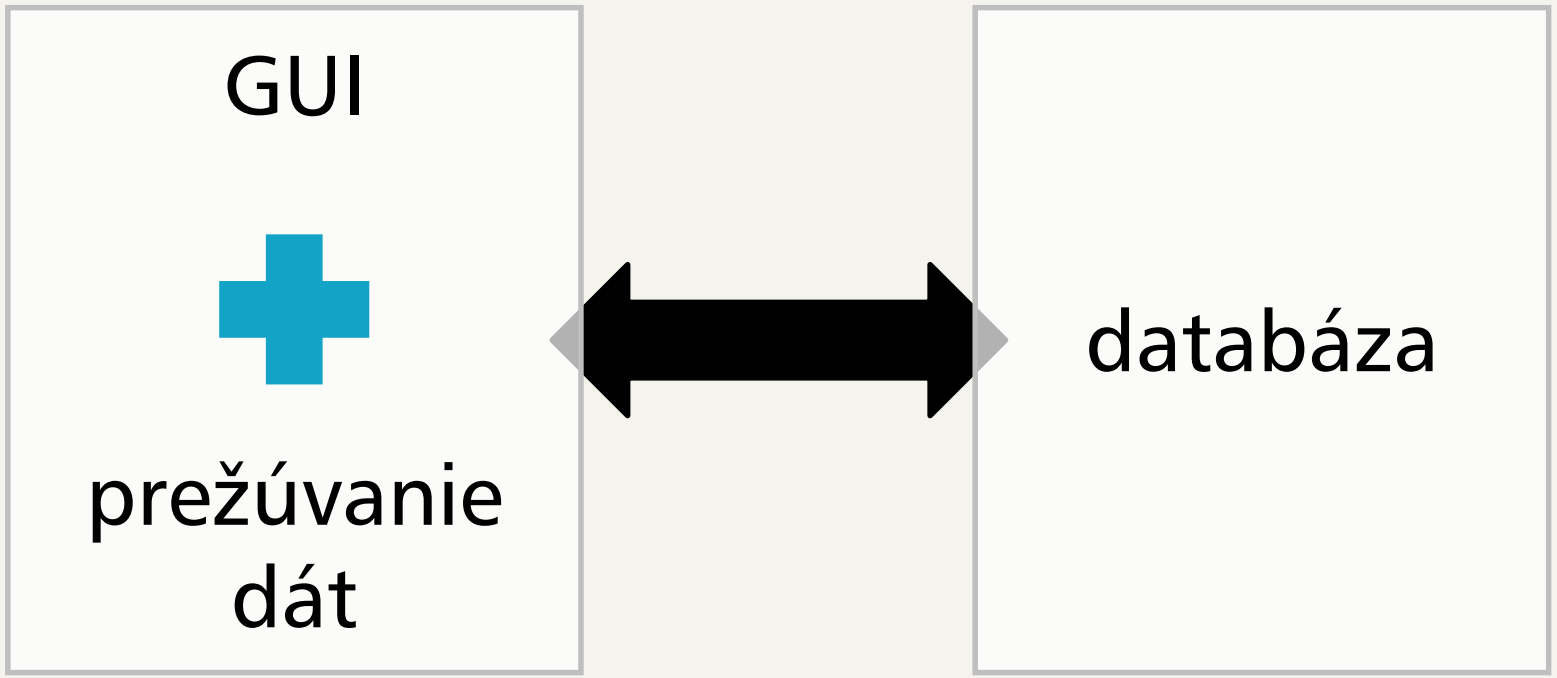

„Problém s programátormi je
v tom, že neviete čo to vlastne
stvárajú, až do chvíle, kým nie
je príliš neskoro.“
– Seymour Cray

ale aplikácia
funguje!

# 2vrstvové aplikácie



# 2vrstvové aplikácie



kľúč pre rozpoznanie 2tier aplikácií

požiadavka na iné GUI vyvoláva mdloby

kľúč pre rozpoznanie 2tier aplikácií

v kóde GUI sa žongluje s dátami

kľúč pre rozpoznanie 2tier aplikácií

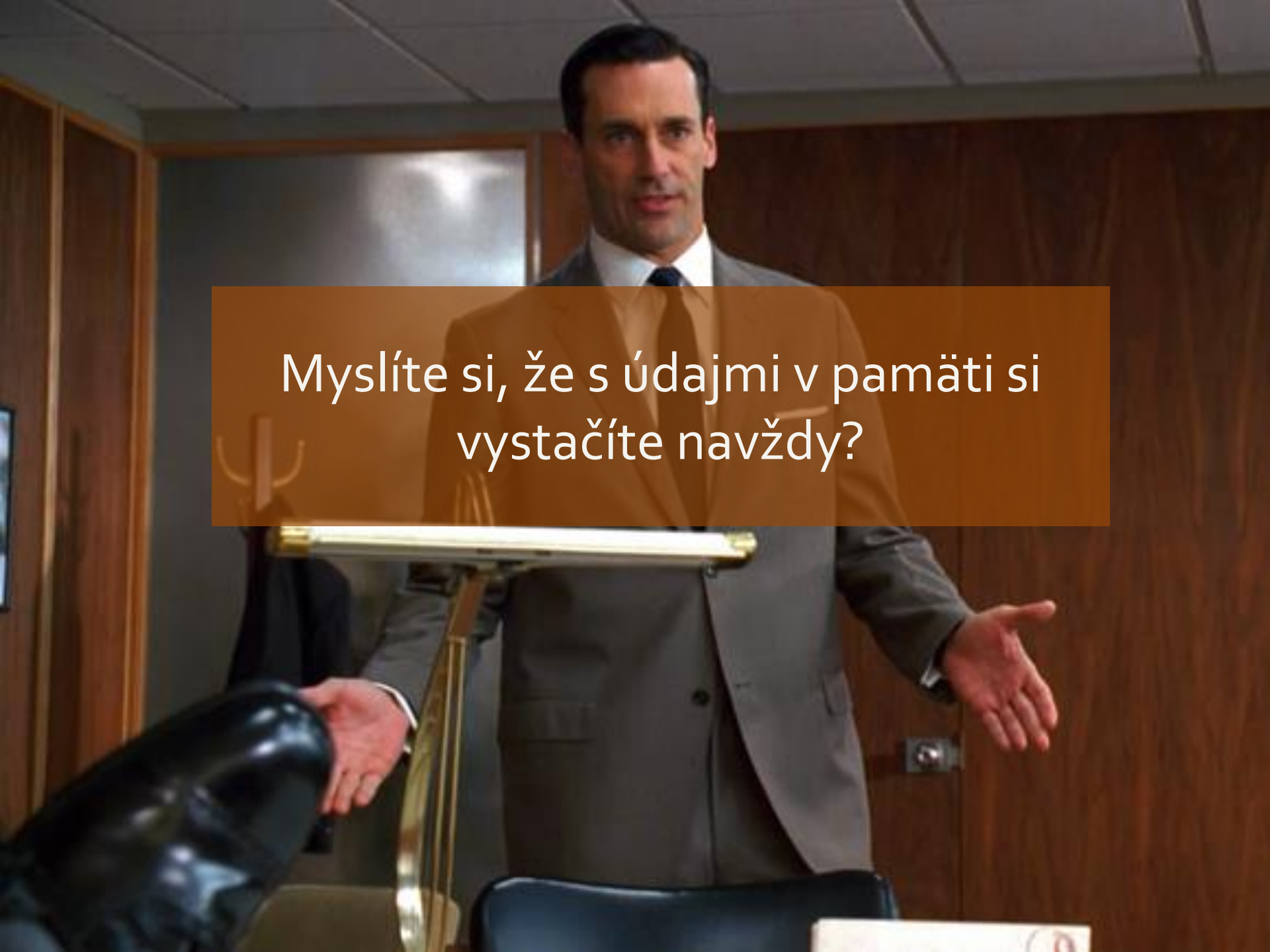
dáta sú na vzdialenom serveri,
zvyšok je v rámci aplikácie

# kľúč pre rozpoznanie 1tier aplikácií

všetko na kope

utility

shellskripty


Myslíte si, že s údajmi v pamäti si vystačíte navždy?

migrujme na citáty
v databáze

vymieňame perzistentnú vrstvu!

refaktor

# prepis CitatyDao

```
itat();
mc^2");
tein");

t();
("YOLO");
        citat2.setAutor("Drake");
-
-       save(citat2);
+
+       save(citat2);
    }
-
-
-
+
    Citat findById(Long id) {
-       for(Citat citat : citaty) {
-           if(citat.getId().equals(id)) {
+       for (Citat citat : citaty) {
+           if (citat.getId().equals(id)) {
                return citat;
            }
        }
        return null;
    }
-
+
    void save(Citat citat) {
-       citat.setId(dalsieId);
-       citaty.add(citat);
-       dalsieId++;
```

prepísať všetky metódy?

zahodíme CitatDao nad súborom?

uvažujme
úplne inak

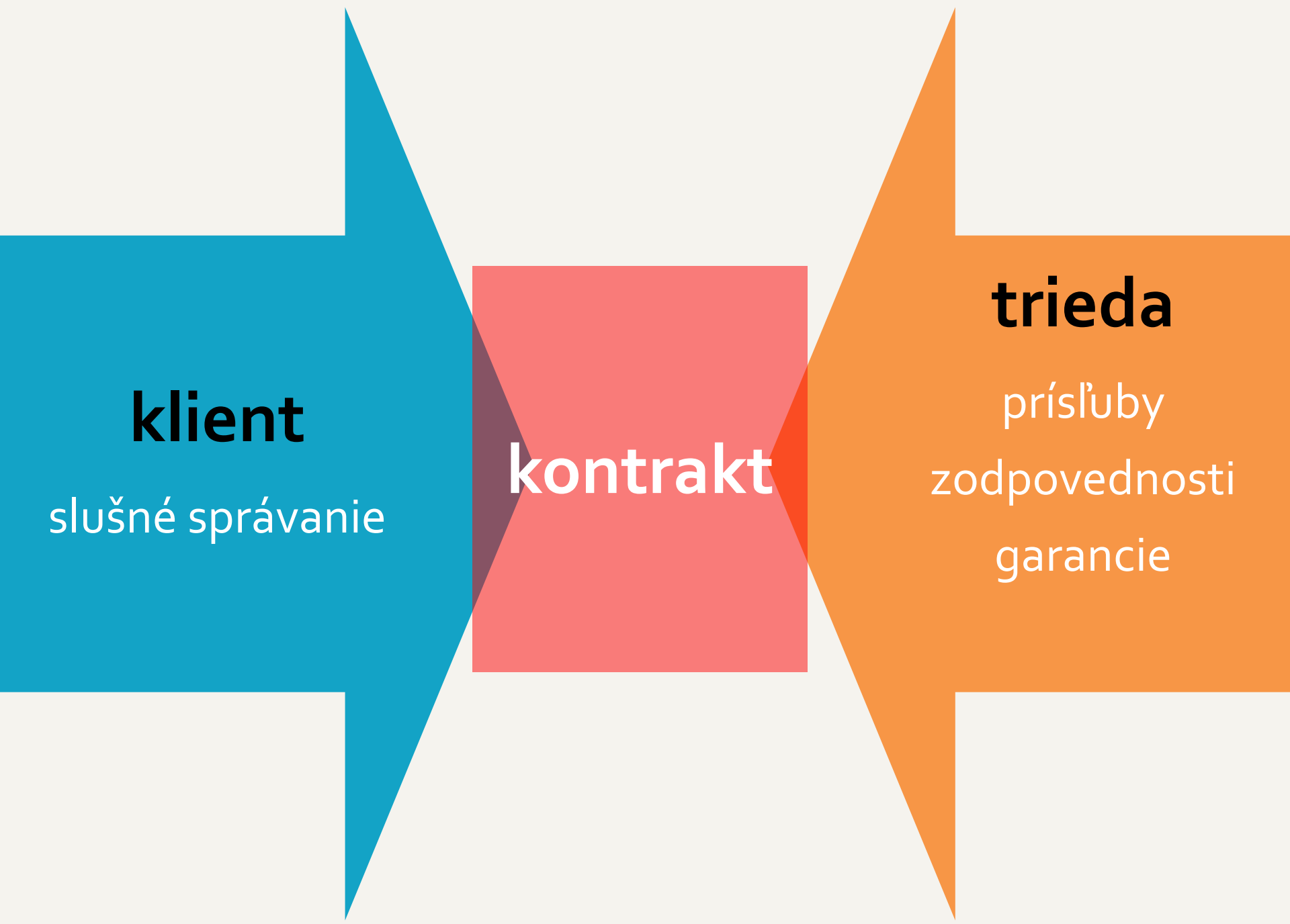
trieda je čierna skrinka

# čo, nie ako

pri návrhu triedy začíname od schopností

implementácia až v druhom rade

vonkajší pohľad na triedu


klient
slušné správanie
kontrakt
trieda
prísľuby
zodpovednosti
garancie

# interfejs

sada operácií, ktoré objekt poskytuje navonok

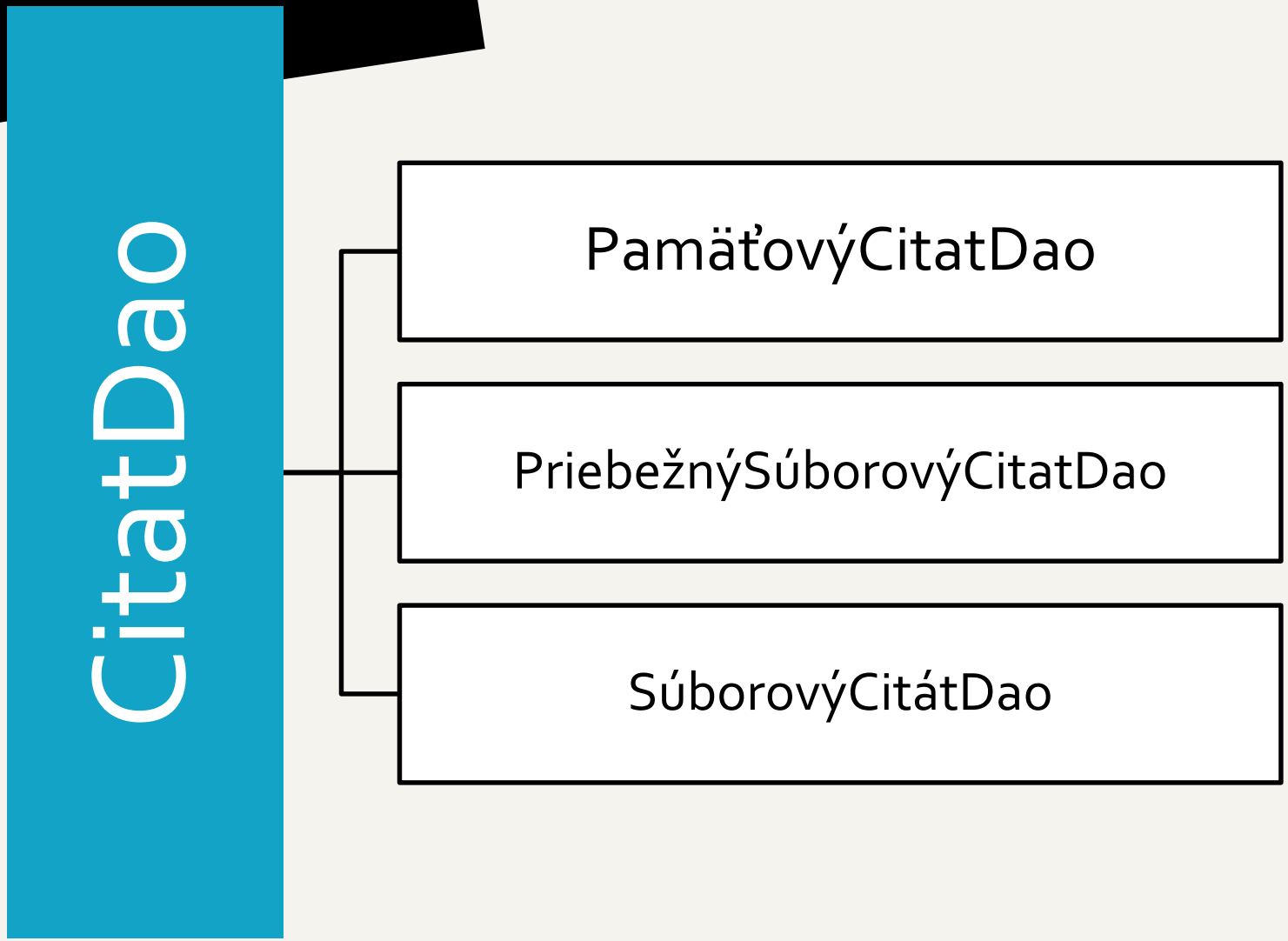
= rozhranie
= protokol

```
public interface CitatDao {
    Citat najdiPodlaId(Integer id);
    List<Citat> najdiPodlaAutora(String autor);
    List<Citat> najdiPodlaZdroja(String fraza);
    List<Citat> listPodlaDatumu();

    void save(Citat c);
    void delete(Citat c);
}
```

ktokoľvek vie naplniť kontrakt

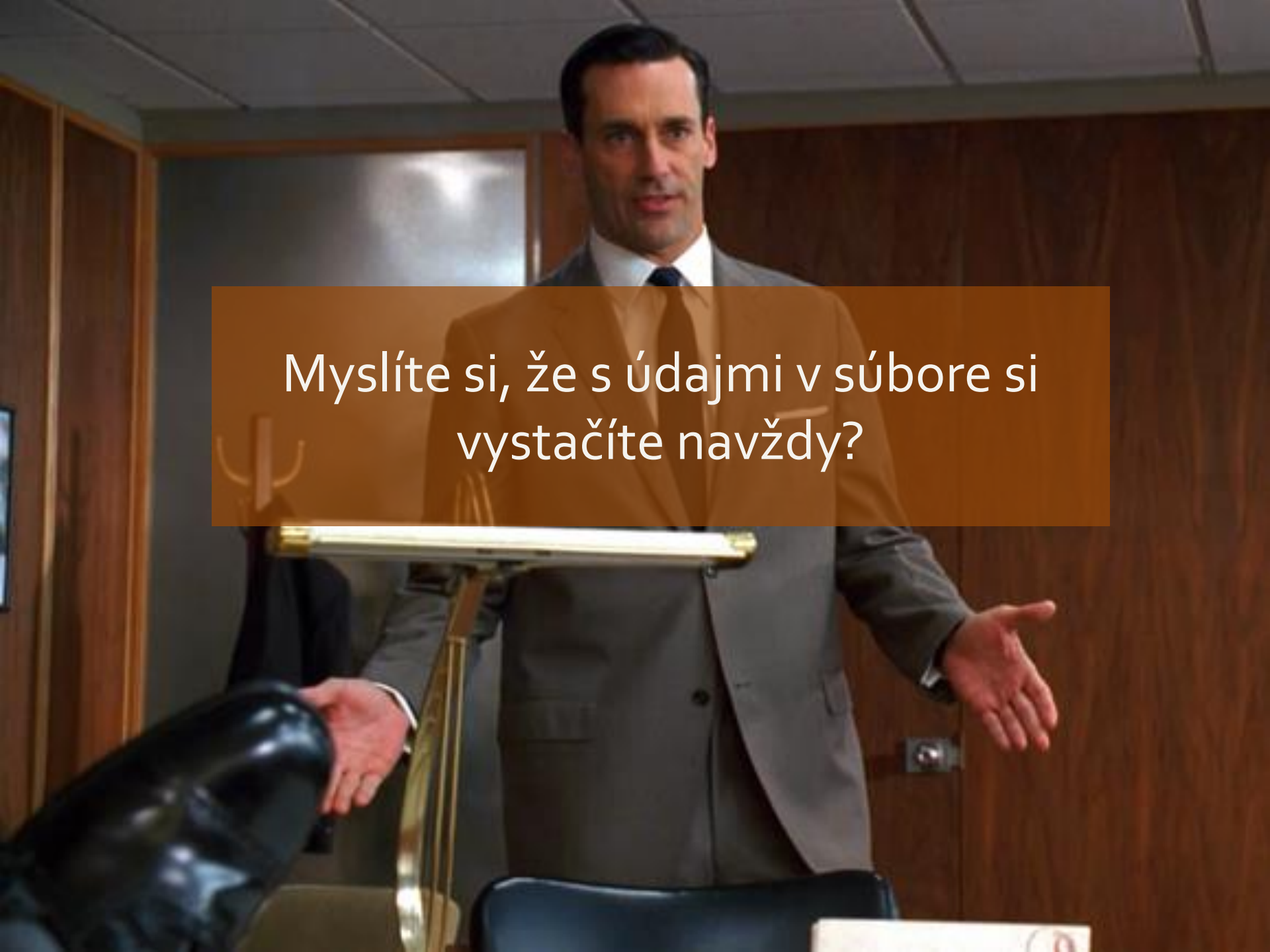
```
public class PamatovyCitatDao
        implements CitatDao
{
   private List<Citat> citaty  = new ArrayList<>();

   public List<Citat> vratVsetky() {
      return this.citaty;
   }
   /* implementácie ostatných metód */
}
```

ktokoľvek vie naplniť kontrakt



# použitie

```
public class Formular {
  private CitatDao citatDao
        = new SuborovyCitatDao();
   ...
}
```


Myslíte si, že s údajmi v súbore si vystačíte navždy?

# prístup k databázam

# relačné databázy

dominantný spôsob ukladania dát

hojný výskum

kilá implementácií

JDBC = Java API
pre SQL

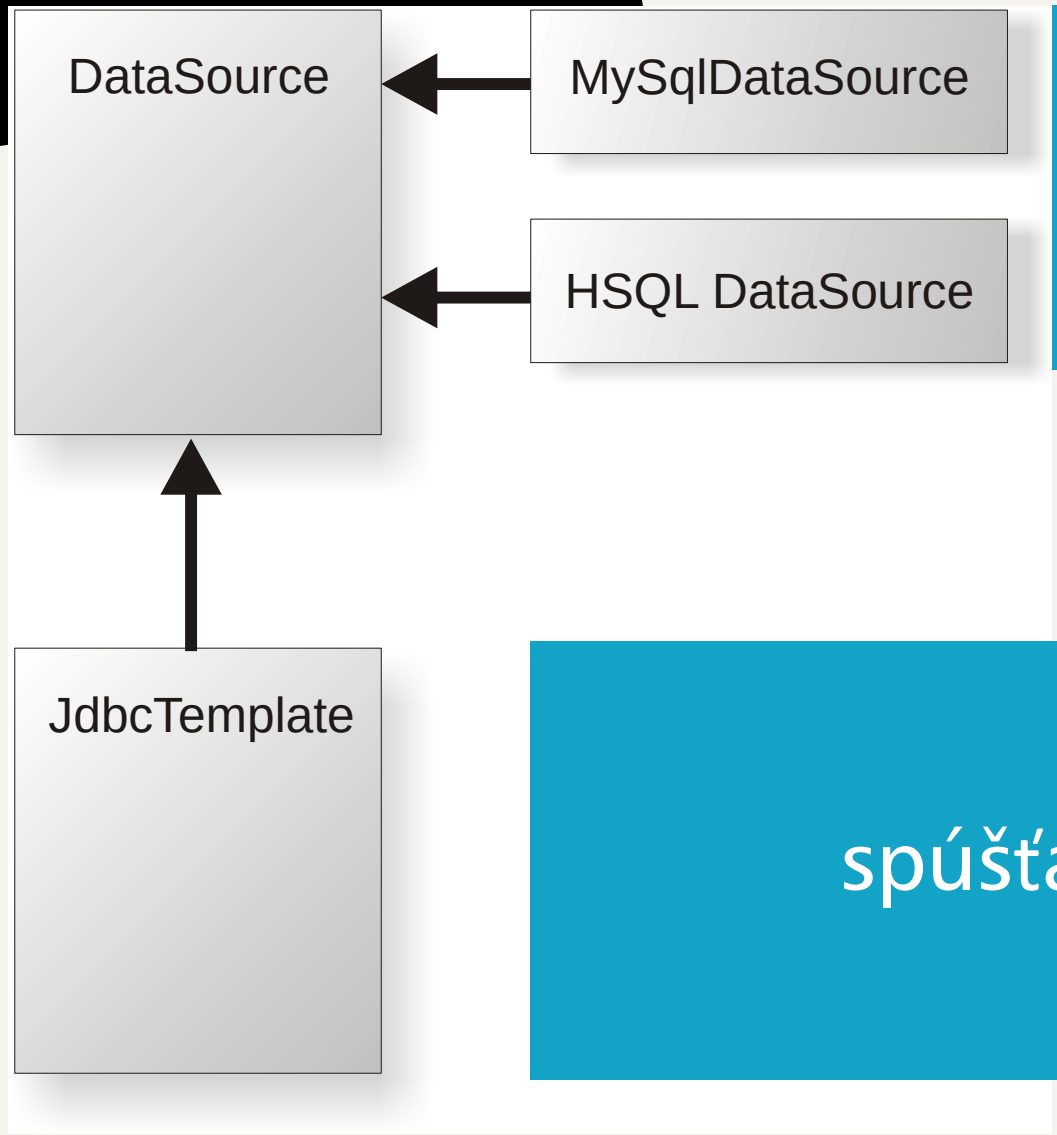
„Takmer každý rozsiahlejší projekt si vytvorí vlastnú nadstavbu nad JDBC.“

– Ja

# Spring Framework

„Budujme lepšie enterprise aplikácie“

# Spring JDBC



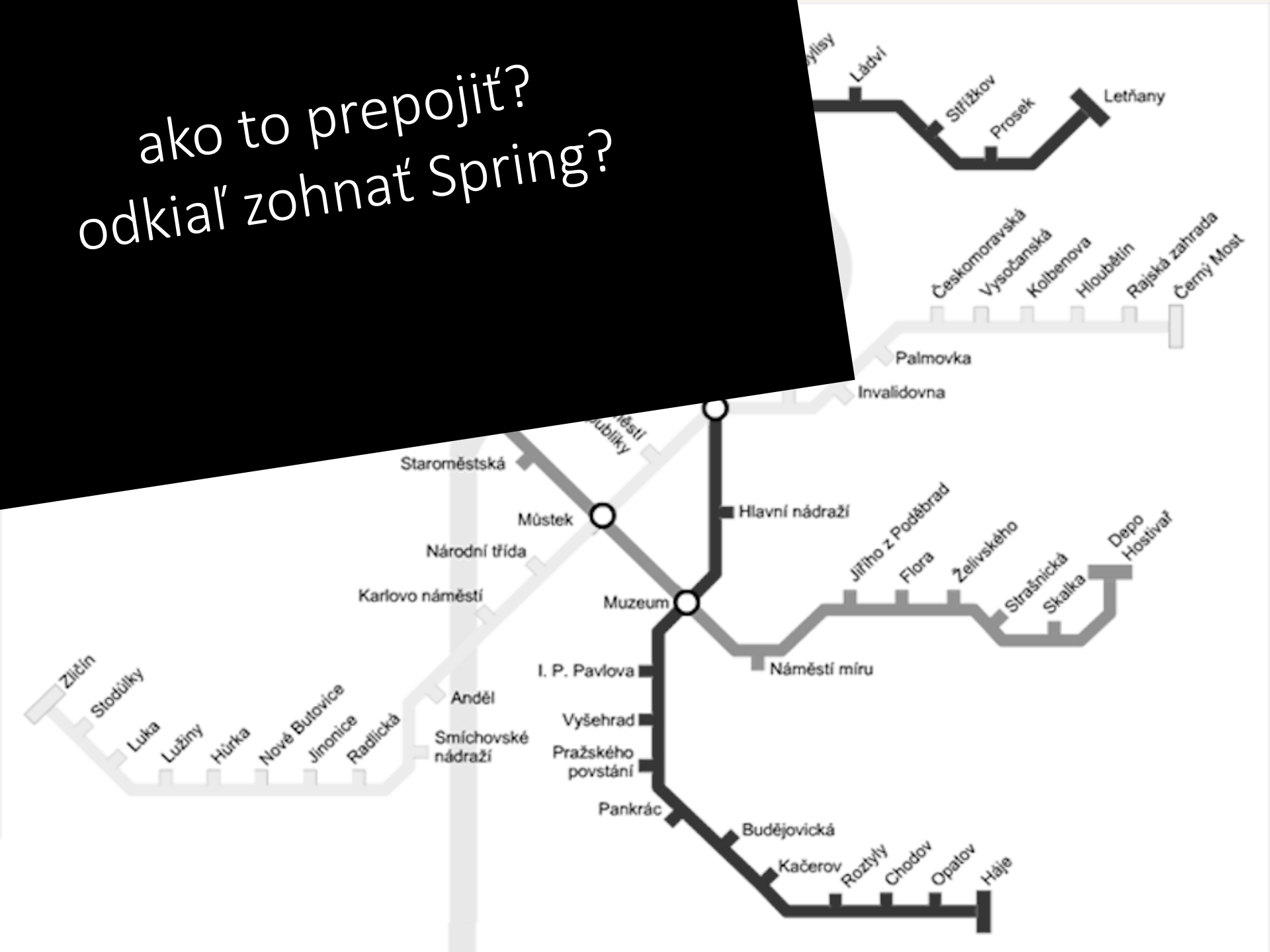

komunikácia s DB

spúšťanie SQL

# Nadišla jar v krajine DAO

http://bit.ly/1rcQnI3

# ako to prepojiť?
# odkiaľ zohnať Spring?



mnoho o mavene

http://bit.ly/1rcQTpr

# Take away

- Inštančné premenné riešia kolaboráciu
- interfejsy = kontrakt
- Spring JDBC = Java + SQL
- Maven = infraštruktúra

Otázky?