

# Objektově orientované databáze a Java

Petr Pelán  
xpelan@fi.muni.cz

A stylized, white, handwritten signature logo of Petr Pelán, consisting of fluid, interconnected loops and lines.

# V čem je problém?

- Potřebujeme ukládat instance tříd
- Třídy na sebe mohou odkazovat
- Můžeme pozměnit definice tříd
- a jiné...

# Hibernate

- Systém pro ukládání objektů z Javy do „Vaší“ relační databáze
- Nejpopulárnější
- Free software
- SQL
- ([www.hibernate.org](http://www.hibernate.org))

# Hibernate – feature list

- Transparent persistence without bytecode processing
- Object-oriented query language
- Flexible object / relational mappings
- Simple APIs
- Automatic primary key generation
- Object/Relational mapping definition
- HDLCA (Hibernate Dual-Layer Cache Architecture)
- Ultra-high performance
- J2EE integration
- And more....

# Hibernate - princip

- Vezměte Vaši třídu a rozšířte ji o „getters“ a „setters“
- Vytvořte XML konfigurační soubor, ve kterém řeknete, která data mají být persistentní
- Nastavte Hibernate parametry Vašeho databázového serveru (další XML soubor)
- Využijte přiložené API funkce
- Zkompilujte

# Ozone

- Objektová databáze v Javě
- Free software
- Žádné SQL – čistá Java
- Databázový server nebo lokální adresář
- ([www.ozone-db.org](http://www.ozone-db.org))

# Ozone - princip

- Vytvořte rozhraní
  - Které rozšiřuje OzoneRemote
  - Definujte hlavičky funkcí
  - Označte funkce, které mohou modifikovat data
- Vytvořte třídu
  - Která rozšiřuje třídu OzoneObject a implementuje Vaše rozhraní
  - Místo `this` používejte `self()`

# Ozone - princip(2)

- Zavolejte Ozone Post Processor (OPP)
  - Vytvoří se proxy třída, která implementuje Vaše rozhraní
  - Vytvoří se factory, která umí vytvářet instance této proxy třídy
- Nyní jsme připraveni pracovat s databází



# Ozone – databázový server

- Z příkazové řádky zavoláte Ozone funkce, které
  - Připraví adresář pro ukládání dat
  - Vytvoří databázi
  - Rozjedou ozone DB server na zvoleném portu
- Ve Vašem Java programu otevřete danou databázi na daném stroji na daném portu

# Ozone – lokální databáze

- Ve Vašem Java programu vytvoříte novou lokální databázi, která bude uložena na daném adresáři
- Nebo otevřete již existující lokální databázi z daného adresáře

```
LocalDatabase db = new LocalDatabase();  
String dbLocation = „adresar“;  
if(!db.exist(dbLocation))  
{  
    db.create(dbLocation);  
}  
db.open(dbLocation);
```

# Ozone – vložení objektu do DB

- Nevytváříme objekt, který vkládáme do DB
- Nýbrž necháme databázi, aby si vytvořila objekt daného typu
- K tomuto objektu přistupujeme přes vytvořené rozhraní

```
ClassInterface obj =  
    (ClassInterface) db.createObject(  
        ObjectImpl.class.getName(),  
        0,  
        „object_name“);
```