

# BookStore

Pavel Mička, Bogdan Kostov

# Obsah

- 1 O aplikaci
- 2 Konfigurace a struktura aplikace
- 3 JUnit

# O aplikaci

- Jednoduchá aplikace demonstrující některé principy CP1
  - JPA mapování (@MappedSuperclass, Cascading)
  - Business logika
  - Komunikace s okolím aplikace pomocí Data transfer objektů
  - Testování funkčnosti pomocí *JUnit*

# persistence.xml

```
<persistence ...>
  <persistence-unit name="sample" transaction-type="RESOURCE_LOCAL">
    ...
    <property name="hibernate.connection.username" value="postgres"/>
    <property name="hibernate.connection.password" value="postgres"/>
    <property name="hibernate.connection.url" value="jdbc:postgresql://
      localhost:5432/bookstore"/>
    <property name="hibernate.hbm2ddl.auto" value="update"/>
    <property name="hibernate.show_sql" value="true"/>
    <property name="hibernate.dialect" value="org.hibernate.dialect.
      PostgreSQLDialect"/>
  </persistence-unit>
</persistence>
```

- Nastaveno pro použití s PostgreSQL
  - V PostgreSQL vytvořte databázi bookstore
  - Tabulky se nagentují automaticky
  - Heslo a username == *postgres*

## Balíčky aplikace

- bo — business objects — entity + jejich logika
- dao — data access objects — přístup do databáze
- provider — interní služby aplikace
- helper — pomocné utility
- service — interface aplikace + agregativní logika
- dto — data transfer objects — objekty určené pro komunikaci aplikace s jejím okolím (controllery, webové služby, serializace atp.)

# JUnit 1.

- Testovací framework
- Od verze 4 využívá anotace pro deklaraci testovacích případů
- Třída reprezentuje sadu testů (anotované metody pak slouží k jejich výkonu)
  - *@BeforeClass* — statická inicializační metoda, provede se při inicializaci testovací sady
  - *@Before* — metoda se provede před každým jednotlivým testem
  - *@Test* — testovací případ
  - *@After* — metoda se provede po každém jednotlivém testu
  - *@AfterClass* — metoda se provede po provedení všech testů

## jUnit II.

- Pro kontrolu splnění dílčích invariantů používáme statické metody třídy *Assert* (obvykle je importujeme do testovací sady všechny pomocí deklaráce *import static org.junit.Assert.\**)

```
@Test
public void testBookDeletedWhenUserRemoved() {
    Long userId = addUser();
    String title = "Bob a Bobek, kralici z
        klobouku";
    bookService.addBook(title, userId);
    assertEquals(1, bookService.getAllBooks().
        size());
    userService.deleteUser(userId);
    assertEquals(0, bookService.getAllBooks().
        size());
}
```

## jUnit III.

- Je vhodné také použít knihovnu *Cobertura*, jež generuje report pokrytí kódu testy (řádky, větve)
- IDE *Netbeans* umí generovat kostru testů (pravé tlačítko na soubor v Záložce projects/Tools/Create jUnit Tests)