**Zadanie**

Vytvorte nový java application projekt s názvom **projekt**.

***Poznámka:****Netbeans vám v projekte vytvorí balík****projekt****a v ňom triedu****Projekt****.*

V balíku **entities** implementujte entitnú triedu **Book**, ktorá umožní pracovať s databázou tabuľkou **BOOK** so stĺpcami:

**ISBN    VARCHAR    Primárny kľúč**

**NAME    VARCHAR**

**PRICE   DOUBLE**

Tabuľka obsahuje informácie o knihách: NAME je názov knihy a PRICE jej cena.

V triede **Projekt***(bol vytvorený automaticky s projektom)* následne implementujte dve metódy, ktoré s využitím entitnej triedy a entity managera pracujú s tabuľkou BOOK:

**public bool bookUpdate( String isbn, String name, Double price)**

Metóda vyhľadá v databáze záznam o knihe podľa zadaného isbn (klúč) a aktualizuje údaje, pričom:

* Ak kniha v databáze neexistuje, vytvorí novú knihu so zadanými údajmi a vráti true.
* Ak v databáze už existuje, aktualizuje údaje o knihe takto:
  + ak argument metódy name je zadaný (nenulový) ale v stĺpci NAME je NULL, zapíše hodnotu do stĺpca.
  + ak argument metódy name je zadaný ale stĺpec NAME obsahuje inú hodnotu, vráti false (a nerobí žiadne zmeny v databáze).
  + ak je argument metódy price zadaný, aktualizuje hodnotu v stlpci PRICE.

Po úspešnom zápise zmien do databázy vráti metóda true.

**public void priceListUpdate(Map<String, Double> priceList)**

 Argument priceList je mapa obsahujúca cenník kníh: kľúč je isbn knihy a hodnota jej cena.

 Pre každú položku z cenníka vyhľadá podľa kľúča záznam v databáze a aktualizuje cenu.

 Ak kniha s daným isbn v databáze ešte neexistuje, vytvorí ju (názov knihy ostane prázdny).

*Návod: Na aktualizáciu použite metódu bookUpdate, pričom názov knihy nezadáte.*

**Poznámky k implementácii a hodnoteniu**

 ◦ *Pre otestovanie metód si môžete vytvoriť program (main metódu), nie je to však súčasťou zadania, nebude sa testovať ani hodnotiť.*

 ◦ *Pri implementácii entitných tried môžete postupovať tak, že najprv vytvoríte entitnú triedu a tabuľku si necháte vygenerovať alebo naopak najprv vytvoríte tabuľku v DB a z nej vygenerujete entitnú triedu.*

 ◦ *Vaša implementácia bude validovaná projektom pomocou unit-testov. Validačný projekt bude mať vlastný persistence.xml a bude používať derby/sample databázu. Preto vám odporúčam použiť pri vývoji tiež derby/sample databázu. Môžete však použiť aj inú relačnú databázu (príp. aj iné vývojové prostredie).****Nevyhnutné však je dodtržať nasledovné****:*

**Pozor - dôležité!**

 ◦ Mená balíkov, tried, metód a typy ich argumentov musia byť presne také ako je uvedené v zadaní.

 ◦ Tabuľka BOOK musí mať presne takú štruktúru (názov tabuľky , mená a typy stĺpcov) ako je v zadaní.

 ◦ Persistence unit musí mať meno **projektPU** (malo by byť takto automaticky inicializované, pre istotu si to však overte v súbore persistence.xml)

**Pokyny pre odovzdanie**

Do **miesta odovzdania**v AISe odovzdávate dva zdrojové súbory:

**Book.java**

**Projekt.java**

*Prosím****nezipovať!***

**Zadanie**

Vytvorte nový java application projekt s názvom **projekt**.

***Poznámka:****Netbeans vám v projekte vytvorí balík****projekt****a v ňom triedu****Projekt****.*

V balíku **entities** implementujte entitnú triedu **Book**, ktorá umožní pracovať s databázou tabuľkou **BOOK** so stĺpcami:

**ISBN    VARCHAR    Primárny kľúč**

**TITLE   VARCHAR**

**PRICE   DOUBLE**

Tabuľka obsahuje informácie o knihách: TITLE je názov knihy a PRICE jej cena.

V triede **Projekt***(bol vytvorený automaticky s projektom)* následne implementujte dve metódy, ktoré s využitím entitnej triedy a entity managera pracujú s tabuľkou BOOK:

**public bool updateBook( String isbn, String title, Double price)**

Metóda vyhľadá v databáze záznam o knihe podľa zadaného isbn (klúč) a aktualizuje údaje, pričom:

* Ak kniha v databáze neexistuje, vytvorí novú knihu so zadanými údajmi a vráti true.
* Ak v databáze už existuje, aktualizuje údaje o knihe takto:
  + ak argument metódy title je zadaný (nenulový) ale v stĺpci TITLE je NULL, zapíše hodnotu do stĺpca.
  + ak argument metódy title je zadaný ale stĺpec TITLE obsahuje inú hodnotu, vráti false (a nerobí žiadne zmeny v databáze).
  + ak je argument metódy price zadaný, aktualizuje hodnotu v stlpci PRICE.

Po úspešnom zápise zmien do databázy vráti metóda true.

**public void updatePriceList(Map<String, Double> priceList)**

 Argument priceList je mapa obsahujúca cenník kníh: kľúč je isbn knihy a hodnota jej cena.

 Pre každú položku z cenníka vyhľadá podľa kľúča záznam v databáze a aktualizuje cenu.

 Ak kniha s daným isbn v databáze ešte neexistuje, vytvorí ju (názov knihy ostane prázdny).

*Návod: Na aktualizáciu použite metódu updateBook, pričom názov knihy nezadáte.*

**Poznámky k implementácii a hodnoteniu**

 ◦ *Pre otestovanie metód si môžete vytvoriť program (main metódu), nie je to však súčasťou zadania, nebude sa testovať ani hodnotiť.*

 ◦ *Pri implementácii entitných tried môžete postupovať tak, že najprv vytvoríte entitnú triedu a tabuľku si necháte vygenerovať alebo naopak najprv vytvoríte tabuľku v DB a z nej vygenerujete entitnú triedu.*

 ◦ *Vaša implementácia bude validovaná projektom pomocou unit-testov. Validačný projekt bude mať vlastný persistence.xml a bude používať derby/sample databázu. Preto vám odporúčam použiť pri vývoji tiež derby/sample databázu. Môžete však použiť aj inú relačnú databázu (príp. aj iné vývojové prostredie).****Nevyhnutné však je dodtržať nasledovné****:*

**Pozor - dôležité!**

 ◦ Mená balíkov, tried, metód a typy ich argumentov musia byť presne také ako je uvedené v zadaní.

 ◦ Tabuľka BOOK musí mať presne takú štruktúru (názov tabuľky , mená a typy stĺpcov) ako je v zadaní.

 ◦ Persistence unit musí mať meno **projektPU** (malo by byť takto automaticky inicializované, pre istotu si to však overte v súbore persistence.xml)

**Pokyny pre odovzdanie**

Do **miesta odovzdania**v AISe odovzdávate dva zdrojové súbory:

**Book.java**

**Projekt.java**

*Prosím****nezipovať!***