

# B-OOP 2025: Zadanie č. 5

10.04.2025

Stiahnite si zdrojové kódy ku zadaniu 4. Nájdete v nich priečinky **src** a **test**. Majú nasledujúcu štruktúru:

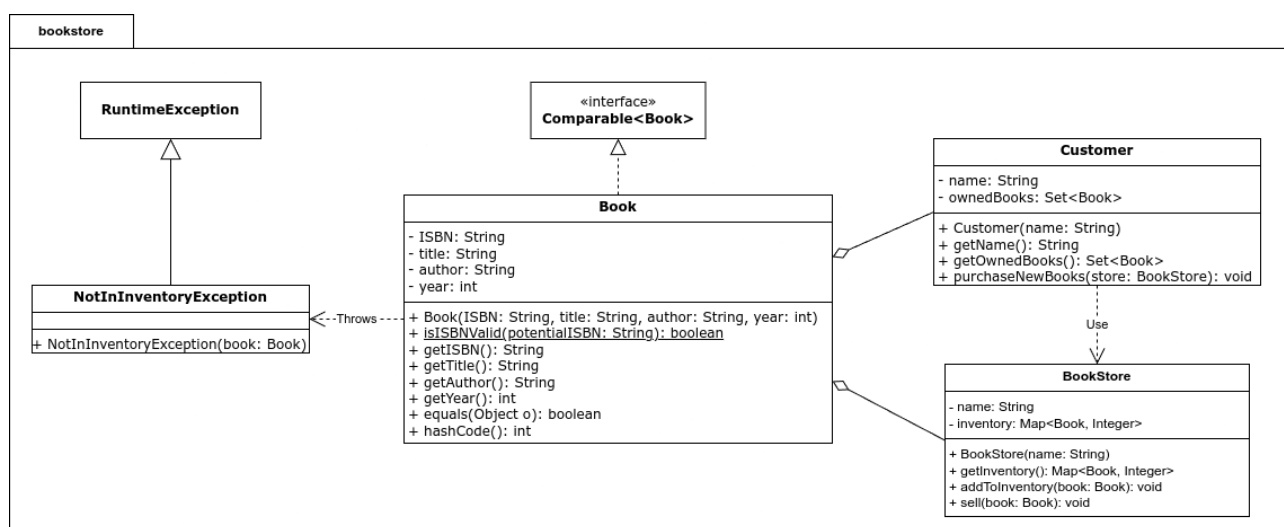
```
z5.zip
├── src
│   ├── bookstore
│   │   ├── Book.java
│   │   ├── BookStore.java
│   │   ├── Customer.java
│   │   └── NotInInventoryException.java
│   └── test
│       ├── bookstore
│       │   ├── BookTest.java
│       │   ├── BookStoreTest.java
│       │   ├── CustomerTest.java
│       │   └── NotInInventoryException.java
```

Vašou úlohou je oboznámiť sa s kolekciami jazyka Java a preukázať túto znalosť implementáciou tried v balíčku **bookstore**.

Naštudujte si:

- metódy rozhrania **Set** (dostupné tu).
- metódy rozhrania **Map** (dostupné tu).
- hlavné implementácie týchto rozhraní a ako si medzi nimi vybrať (môžete čerpať odtiaľto).

Potom sa môžete pustiť do implementácie jednotlivých tried. Vaša implementácia musí byť v súlade s nižšie uvedeným UML diagramom tried (tzn. všetko, čo je uvedené v diagrame, musí nevyhnutne byť vo vašej implementácii; ak potrebujete, môžete si navyše definovať vlastné atribúty a metódy), ako aj s dokumentačnými komentármi v poskytnutých zdrojových kódoch.



Obr. 1: UML diagram tried pre zadanie 5

V texte nižšie uvádzame krátky popis fungovania jednotlivých tried.

Trieda **Book** je zjednodušenou reprezentáciou knihy. Obsahuje ISBN, názov, meno autora a rok vydania. Definujte konštruktor, ktorý akceptuje všetky tieto parametre a validuje ich. ISBN musí mať platnú kontrolnú

sumu (viď nižšie), názov knihy a meno autora nesmú byť null ani prázdne a rok nesmie byť záporný. Ak je porušená ktorákoľvek z týchto podmienok, konštruktor vyhodí `IllegalArgumentException`.

ISBN má 2 možné formáty. ISBN-10 a ISBN-13. ISBN-10 obsahuje 10 číslíc alebo 9 číslíc a znak X. Znak X sa môže vyskytovať iba na konci ISBN-10 a má hodnotu 10. ISBN-13 obsahuje 13 číslíc. Znak v ISBN môžu byť rozdelené do skupín oddelených pomlčkami či medzerami. Pre potreby tohto zadania takéto delenie nepripúšťame. Ak ISBN obsahuje iný znak než číslice 0-9 alebo znak X, automaticky ho považujeme za nesprávne. Pre kontrolu správnosti nemusíte brať do úvahy význam číslíc ani skupín. Postačí, keď spočítate kontrolnú sumu. Jej výpočet je pekne popísaný tu.

Tu uvedieme nasledujúce 2 príklady výpočtu:

- 080442957X je validné ISBN-10. Na overenie najprv spočítame sumu  $s$ :  $s = (0 * 10) + (8 * 9) + (0 * 8) + (4 * 7) + (4 * 6) + (2 * 5) + (9 * 4) + (5 * 3) + (7 * 2) + (10 * 1) = 209$ . Potom sa opýtame, či je  $s$  modulo 11 rovné 0. V tomto prípade platí  $209 \% 11 = 0$ . Preto je ISBN platné. Inak by bolo neplatné.
- 9780306406157 je validné ISBN-13. Na overenie najprv spočítame sumu  $s$ :  $s = (9 * 1) + (7 * 3) + (8 * 1) + (0 * 3) + (3 * 1) + (0 * 3) + (6 * 1) + (4 * 3) + (0 * 1) + (6 * 3) + (1 * 1) + (5 * 3) + (7 * 1) = 100$ . Potom sa opýtame, či je  $s$  modulo 10 rovné 0. V tomto prípade platí  $100 \% 10 = 0$ . Preto je ISBN platné. Inak by bolo neplatné.

Atribúty triedy `Book` sa nebudú po vytvorení inštancie meniť. Ku všetkým atribútom implementujte `get` metódy. Tiež musíte implementovať metódy `equals` a `hashCode`.

Knihy sú predávané v kníhkupectve. To reprezentujeme triedou `BookStore`. Kníhkupectvo má názov a inventár. Názov kníhkupectva je parametrom konštruktora a nesmie byť null ani prázdny reťazec. Ak by taká situácia nastala, konštruktor vyhodí výnimku `IllegalArgumentException`.

Inventár bude reprezentovaný mapou. Na tento účel odporúčame `HashMap`. Kľúče budú knihy a hodnoty budú počty kníh na sklade. Knihu do inventáru pridávame metódou `addToInventory`. Ak sa kniha v inventári ešte nenachádza, uloží sa do inventára ako nový kľúč s hodnotou 1. Ak sa v inventári už nachádza, inkrementuje sa počet o 1.

Z inventáru sa knihy odoberajú metódou `sell`. Ak sa kniha nachádza v inventári v nenulovom počte, metóda `sell` dekrementuje počet kusov tejto knihy o 1. Ak sa nenachádza v inventári (tzn. kniha neexistuje ako kľúč v mape alebo je počet kusov nulový), metóda `sell` vyhodí výnimku `NotInInventoryException`. Kniha, ktorá bola pridaná do inventáru, sa z neho nikdy nevymaže. Ak je počet kusov danej knihy nulový, jednoducho to značí, že kniha aktuálne nie je dostupná. To ale neznamená, že v budúcnosti nebude opäť naskladnená. Preto nedáva zmysel úplne odstraňovať knihy z inventáru, keď ich počet dosiahne hodnotu 0.

Výnimka `NotInInventoryException` dedí od `RuntimeException`. Jej konštruktor berie ako parameter knihu, ktorá nie je na sklade. Môžete predpokladať, že kniha, ktorá sa dostane na vstup konštruktora výnimky `NotInInventoryException`, bude vždy odlišná od hodnoty null. Pole `message` výnimky musí byť vo formáte:

```
"Book ISBN " + ISBN + " is not in inventory!"
```

To je možné docieľiť adekvátnym volaním `super()`.

Knihy nakupuje zákazník. Ide o zákazníka, ktorého by si prialo mať každé kníhkupectvo. Z každej knihy vlastní práve jeden kus, knihy má zoradené podľa ich roku publikácie (ak sa náhodou stane, že rok publikácie je na niektorých knihách zhodný, tak tieto knihy usporiada lexikograficky podľa ich ISBN) a kupuje každú knihu, ktorú ešte nemá.

Tento zákazník je reprezentovaný triedou `Customer`. Konštruktor tejto triedy prijíma parameter `name`, ktorý musí byť odlišný od null a neprázdny. Ak to neplatí, tak vyhodí výnimku `IllegalArgumentException`. Nakoľko zákazník vlastní z každej knihy práve jeden kus (a nemá záujem kupovať duplikáty), knihy má uložené v množine (`Set`). Vyberte vhodnú konkrétnu implementáciu množiny, ktorá usporadúva svoje prvky podľa roku (resp. podľa ISBN v prípade rovnosti).

Detailne si preštudujte UML diagram, ktorý sme vám poskytli. Obsahuje niekoľko detailov, ktoré vás navedú na správnu implementáciu usporadúvania kníh, ako aj na ich použitie ako kľúčov `HashMap`.

Zákazník sa príležitostne rozhodne, že nastal čas nakúpiť nové knihy (metóda `purchaseNewBooks`). V takom prípade prehľadá inventár kníhkupectva a zavolá metódu `sell` pre každú knihu, ktorú ešte nevlastní. Ak je nákup úspešný, vypíše sa na štandardný výstup správa v tvare:

```
"Customer " + name + " has purchased the book " + ISBN
```

Zároveň si zákazník pridá knihu do kolekcie. Ak je nákup neúspešný (tzn. metóda `sell` vyhodí výnimku `NotInInventoryException`), tak sa vypíše na štandardný výstup správa v tvare:

```
"Customer " + name + " could not buy the book " + ISBN
```

Správa o úspechu/neúspechu sa vypíše pre každú knihu, ktorú sa zákazník rozhodne kúpiť. Ak je kníhkupectvo `null`, metóda `purchaseNewBooks` vyhodí výnimku `IllegalArgumentException`.

Zdrojové kódy k zadaniu 5 obsahujú podmnožinu unit testov, ktoré budú použité na hodnotenie vašich zadaní. Tieto testy môžete použiť na otestovanie vašej implementácie pred odovzdaním. Prejdenie týmito testami je nutná podmienka na to, aby ste získali nenulové hodnotenie!

**Pozor!** Je dôležité, aby ste dodržali predpísané názvy tried a metód, ktoré vyplývajú z popisu zadania a z priloženého UML diagramu. Nemeňte štruktúru projektu! Ak to považujete za potrebné, môžete si v jednotlivých triedach vytvoriť dodatočné pomocné metódy a premenné. Môžete rozšíriť existujúce triedy o implementáciu dodatočných rozhraní. Nemeňte metódy, ktoré už sú implementované.

V AIS je vytvorené miesto odovzdania OOP - **Zadanie 5** do 2025-04-17T23:59:00+01:00. Do tohto miesta odovzdania nahráte **zip** archív (to znamená nie **rar**, ani **tar...**), ani žiadny iný formát než **zip**. Váš archív bude mať nasledujúcu štruktúru (za predpokladu, že ste nevytvorili dodatočné triedy):

```
archiv.zip
├── src
│   └── bookstore
│       ├── Book.java
│       ├── BookStore.java
│       ├── Customer.java
│       └── NotInInventoryException.java
```

Vaše zadanie bude hodnotené pomocou automatizovaných testov, aby sme si overili, či splňuje všetky náležitosti. Zadania, ktoré splnia uvedené náležitosti, a prejdú všetkými automatizovanými testami, budú ohodnotené 1 bodom.