**Book**

* Importování knihoven lombok.Getter, lombok.Setter a lombok.RequiredArgsConstructor.
* Deklarace třídy Book, která rozšiřuje třídu Item.
* Anotace @Getter, @Setter a @RequiredArgsConstructor pro automatickou generaci getterů, setterů a konstruktoru s požadovanými argumenty.
* Deklarace atributů title, author a genre.
* Definice konstruktoru, který inicializuje atributy title, author a genre.
* Přepsání metody toString() pro vygenerování řetězce obsahujícího informace o knize.

**Echoer**

* Importujte třídy pro manipulaci s vstupem a výstupem a síťovou komunikaci.
* Definujte třídu Echoer, která dědí od třídy Thread pro asynchronní zpracování.
* Vytvořte konstruktor třídy Echoer, který přijímá soket jako vstupní parametr.

V metodě run():

a. Vytvořte objekt BufferedReader pro čtení ze vstupního proudu soketu.

b. Vytvořte objekt PrintWriter pro zápis do výstupního proudu soketu.

c. Spusťte nekonečnou smyčku pro čtení ze vstupu soketu.

d. Přečtěte zprávu ze vstupního proudu soketu.

e. Pokud je zpráva "exit", ukončete smyčku.

f. Jinak, pokud začíná zpráva "RESERVATION: ", zpracujte rezervaci a odešlete potvrzení.

g. V opačném případě odešlete zprávu "UNKNOWN COMMAND".

V bloku catch zachyťte výjimky IOException a vypište chybu.

Ve vnitřním bloku finally uzavřete soket.

**Genre**

Importujte knihovnu Lombok pro automatické generování getteru.

Definujte výčtový typ Genre.

Pro každý žánr definujte konstantu výčtového typu s odpovídajícím názvem.

Každé konstantě výčtového typu přiřaďte řetězcovou hodnotu představující název žánru.

Vytvořte privátní atribut genre pro uchování názvu žánru.

Vytvořte konstruktor, který přijímá řetězec a nastaví hodnotu atributu genre.

Přetěžte metodu toString(), která vrací název žánru jako řetězec.

**Item**

* Importujte výjimky ItemAlreadyBorrowedException a UserNotFoundException.
* Importujte knihovnu Lombok pro automatické generování getterů a setterů.
* Definujte abstraktní třídu Item.
* Definujte atributy třídy: isAvailable, borrowedFrom, borrowedTo, user.
* Vytvořte konstruktor třídy Item.
* Definujte abstraktní metodu getTitle().
* Vytvořte metodu isAvailable(), která vrací stav dostupnosti položky.
* Vytvořte metodu borrowToUser(), která půjčuje položku uživateli a nastavuje časy výpůjčky.
* Vytvořte metodu returnToLibrary(), která vrací položku do knihovny.
* Vytvořte metodu reserveToUser(), která rezervuje položku uživateli.
* Přetěžte metodu toString(), která vrací textovou reprezentaci objektu třídy Item.

**Exception** ItemAlreadyBorrowedException

* Definujte veřejnou třídu ItemAlreadyBorrowedException.
* Tato třída rozšiřuje výjimkovou třídu RuntimeException.
* Definujte konstruktor, který přijímá zprávu a volá konstruktor rodičovské třídy s touto zprávou.

**Exception** ItemDoesNotExistException

Vytvořte veřejnou třídu ItemDoesNotExistException.

Tato třída rozšiřuje výjimkovou třídu RuntimeException.

Definujte konstruktor, který přijímá zprávu a volá konstruktor rodičovské třídy s touto zprávou.

**Client**

* Importujte potřebné třídy pro manipulaci se vstupem a výstupem a síťovou komunikaci.
* Definujte veřejnou třídu LibraryClient.

V metodě main():

a. Vytvořte soket pro spojení se serverem na adrese "localhost" a portu 5000.

b. Vytvořte objekty pro čtení ze serveru a zápis na server.

c. Vytvořte objekt třídy Scanner pro čtení vstupu od uživatele.

d. V nekonečné smyčce:

i. Vyzvěte uživatele k zadání textu pro odeslání.

ii. Přečtěte vstup od uživatele.

iii. Pokud vstup není "exit":

1. Pošlete text na server.

2. Přečtěte odpověď ze serveru.

3. Vypište odpověď ze serveru.

* e. Pokud je vstup "exit", ukončete smyčku.
* V bloku catch zachyťte výjimku IOException a vypište chybu spojenou s klientem.

**Server**

* Importujte výjimky ItemAlreadyBorrowedException a UserNotFoundException.
* Importujte třídy pro manipulaci se vstupem a výstupem a síťovou komunikaci.
* Importujte třídy pro vytvoření objektu uživatele, knihy a knihovny.
* Definujte třídu LibraryServer.
* Vytvořte statického uživatele user pomocí User.Builder.
* Vytvořte instance knih book1 a book2.
* Vytvořte statickou metodu generateLibrary(), která generuje knihovnu s několika knihami.

V metodě main():

a. Vytvořte nový serverový soket, který naslouchá na portu 5000.

b. V nekonečné smyčce:

* i. Přijímejte nové připojení a pro každé připojení vytvořte nový vlákno ClientHandler.
* Vytvořte vnořenou třídu ClientHandler, která dědí od třídy Thread.
* V konstruktoru ClientHandler přijímejte soket a knihovnu.

V metodě run():

a. Vytvořte objekty pro čtení ze vstupu klienta a zápis na výstup klienta.

b. V nekonečné smyčce:

i. Čtěte vstup od klienta.

ii. Pokud klient poslal "exit", ukončete smyčku.

iii. Pokud klient poslal "RESERVATION: ", zpracujte rezervaci.

iv. Pokud klient poslal "HELP", pošlete seznam dostupných položek.

* v. Pokud klient poslal jiný příkaz, pošlete zprávu "UNKNOWN COMMAND".
* V bloku catch zachyťte výjimku IOException a vypište chybu spojenou se serverem.
* Ve vnitřním bloku finally uzavřete soket.

**Magazine**

* Importujte knihovnu Lombok pro automatické generování getterů, setterů a konstruktoru.
* Definujte třídu Magazine, která dědí od třídy Item.
* Definujte atributy publisher, magazineName a publicationDate.
* Vytvořte konstruktor třídy Magazine, který přijímá všechny atributy a je označen jako @RequiredArgsConstructor.
* Přetěžte metodu getTitle(), která vrací název časopisu.
* Přetěžte metodu toString(), která vrací textovou reprezentaci objektu třídy Magazine.

DVD

ty samé kroky jako jsou na Magazine, s ohledem na zadání

**Main**

V metodě main():

a. Vytvořte uživatele user1 a user2 pomocí builderu třídy User.

b. Vytvořte uživatelské kanály pro user1 a user2.

c. Vytvořte seznam uživatelů obsahující user1 a user2.

d. Vypište informace o uživatelích.

e. Vytvořte instance knih.

f. Vytvořte seznam knih a vypište jejich informace.

g. Přidejte knihy do seznamu knih.

h. Vytvořte instance magazínů a novin.

i. Vytvořte seznam magazínů a novin a vypište jejich informace.

j. Přidejte magazíny a noviny do seznamu.

k. Vytvořte seznam všech položek.

l. Vytvořte knihovnu a přidejte uživatelské kanály pro všechny položky.

m. Půjčte knihu, magazín a noviny uživateli 1 a rezervujte noviny pro uživatele 2.

n. Vypište informace o půjčených a rezervovaných položkách.

o. Zkontrolujte dostupnost knih.

p. Vypište informace o tom, kolik položek má každý uživatel půjčených a rezervovaných.

* q. Vypište dostupné položky v knihovně.

**Newspaper**

Importujte knihovnu Lombok pro automatické generování getterů, setterů a konstruktoru.

Definujte třídu Newspaper, která dědí od třídy Item.

Definujte atributy publisher, headline a date.

Vytvořte konstruktor třídy Newspaper, který přijímá všechny atributy a je označen jako @RequiredArgsConstructor.

Přetěžte metodu getTitle(), která vrací nadpis novin.

Přetěžte metodu toString(), která vrací textovou reprezentaci objektu třídy Newspaper.

**User**

* Importujte knihovnu Lombok pro automatické generování getterů a setterů.
* Definujte třídu User<T extends Item>, která má parametrizovaný typ T, omezený na typ Item.
* Definujte atributy name, borrowedItems a reservedItems.
* Definujte privátní konstruktor třídy User a statickou vnořenou třídu Builder<T>.
* V konstruktoru třídy User inicializujte atributy z builderu.
* V metodě Builder nastavte jméno.
* Implementujte metodu borrowItem(T item), která půjčí položku uživateli, pokud je dostupná nebo ji již uživatel půjčil a položka není již půjčená.
* Implementujte metodu returnOrUnreservedItem(T item), která vrátí nebo zruší rezervaci položky uživatelem.
* Implementujte metodu reserveItem(T item), která rezervuje položku uživateli, pokud je dostupná a není již rezervovaná.
* Implementujte metody getBorrowedItemsList() a getReservedItemsList(), které vracejí seznam půjčených a rezervovaných položek.
* Implementujte metody hasBorrowedItems() a hasReservedItems(), které vracejí true, pokud uživatel má nějaké půjčené nebo rezervované položky.
* Implementujte metody countBorrowedItems() a countReservedItems(), které počítají počet půjčených a rezervovaných položek.

**UserChannel**

* Definujte rozhraní UserChannel<T extends Item>, kde T je parametrizovaný typ omezený na typ Item.
* Definujte metody onItemBorrowed(T item), onItemReturned(T item) a onItemReserved(T item), které přijímají jako parametr položku a nevracejí žádnou hodnotu.
* Tyto metody slouží k informování uživatelského kanálu o událostech souvisejících s půjčením, vrácením a rezervací položek.

**UserChannelImpl**

* Vytvořte třídu UserChannelImpl<T extends Item>, která implementuje rozhraní UserChannel<T>.
* Definujte privátní atribut user typu User<T>.
* Vytvořte konstruktor třídy, který přijímá uživatele.
* Implementujte metody onItemBorrowed(T item), onItemReturned(T item) a onItemReserved(T item), které vypisují zprávy o půjčení, vrácení a rezervaci položek, obsahující jméno uživatele a název položky.

**UserNotFoundException**

* Definujte třídu UserNotFoundException jako odvozenou třídu od RuntimeException.
* Implementujte konstruktor třídy, který přijímá zprávu chyby a volá konstruktor nadřazené třídy s touto zprávou.