CODEFORCES
Sponsored by TON

|
popovgleb | Выйти

ГЛАВНАЯ ТОП КАТАЛОГ СОРЕВНОВАНИЯ ТРЕНИРОВКИ АРХИВ ГРУППЫ РЕЙТИНГ EDU API КАЛЕНДАРЬ ПОМОЩЬ

ЗАДАЧИ ОТОСЛАТЬ МОИ ПОСЫЛКИ СТАТУС ПОЛОЖЕНИЕ ЗАПУСК

A. Online Book Store

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Introduction

In this assignment, you will develop a system for the management of an **online bookstore**. Implemented system should support book insertion, controlling access to books, users subscriptions, and notifying users about books updates. The main objective of the assignment is to apply design patterns covered in SSAD classes.

Entities

The main entities present in the system are the following:

- **Book**: Represents a book entity and has *title*, *author*, and *price* (floating point number) properties. All books are available in two formats: *Textual* and *Audio*.
- **User**: There are two types of users: *Standard* and *Premium*. Users of both types can subscribe to get notifications about books updates. However, *Standard* users can access only *textual* version of books, while *Premium* users can access both *textual* and *audio* versions of books.

Operations

Your system should support the following set of operations:

Input command	Description	Output
createBook *title author price*	Creates a new book with the given properties	If book with the given title already exists it should print message "`Book already exists`" without quotes; otherwise there should be no output.
createUser *user_type username*	Creates a new user according to the given type (*standard* or *premium*)	If user with the given username already exists it should print message "`User already exists`" without quotes; otherwise there should be no output.
subscribe *username*	Subscribes user with the given username to get notifications about books' prices updates	If user with the given username is already subscribed it should print message "`User already subscribed`" without quotes; otherwise there should be no output.
unsubscribe *username*	Unsubscribes user with the given username from getting notifications about books' prices updates	If user with the given username is not subscribed it should print message "`User is not subscribed`" without quotes; otherwise there should be no output.

IU SSAD Spring 2024

Закрытая

Участник

☆

→ Соревнования группы

- SSAD_Assignment_4
- SSAD_Assignment_3
- SSAD_Assignment_2
- SSAD_Assignment_1

SSAD_Assignment_4

Соревнование идет

4 дня

Участник

☆

→ Языки

Только перечисленные языки могут быть использованы для решения задач соревнования

SSAD_Assignment_4:

- GNU G++14 6.4.0
- GNU G++17 7.3.0
- GNU G++20 13.2 (64 bit, winlibs)
- C# 8, .NET Core 3.1
- C# 10, .NET SDK 6.0
- C# Mono 6.8
- Java 21 64bit
- Java 8 32bit

→ Отослать?

Язык: GNU G++20 13.2 (64 bit, wi ▾

Выберите файл: Выберите файл Файл ...выбран

Отослать

→ Набранные баллы

	Баллы
A	

updatePrice *title* *new_price*	Updates the price of the book with the given title and notifies all subscribed users about the update	If there are no subscribed users, there should be no output; otherwise it should print the following message without quotes for all subscribed users "*username* notified about price update for *title* to *new_price*". Note that users should be notified in the same order in which they subscribed. In other words, if user A subscribed before user B, user A should be notified about update before user B.
readBook *username* *title*	User with the given username reading book with the given title	It should print the following message without quotes "*username* reading *title* by *author*"
listenBook *username* *title*	User with the given username listening book with the given title	In case of *standard* user trying to listen a book it should print the following message without quotes "No access". In case of *premium* user it should print the following message without quotes "*username* listening *title* by *author*"
end	Used to announce end of the input file	

Please check example given below for better understanding of the input and output formats.

Error Handling

There are four potential problems in input that you need to handle:

Problem	Response
Creating a book with the title that already exists in the system	Book already exists
Creating a user with the username that already exists in the system	User already exists
Trying to subscribe user that is already subscribed	User already subscribed
Trying to unsubscribe user that is not subscribed	User is not subscribed

After error handling execution **should not** be terminated. Instead, it should continue until `end` command.

In is important to note the following constraints:

- It is **guaranteed** that all books will have **one-word name** (for example, "AnnaKarenina" instead of "Anna Karenina"). The same applies for the book authors (for example, "LeoTolstoy" insted of "Leo Tolstoy"). Therefore, you do not need to handle multiple-words cases.
- There are no test cases with subscribing or unsubscribing users that **does not exist** in the system.
- There are no test cases with updating a price of book that **does not exist** in the system.
- There are no test cases with **non-existent user** attempting to read/listen a book.
- There are no test cases with user attempting to read/listen **non-existent book**.
- There are no test cases with other types of **invalid input**.

Example

Please carefully read the following example for better understanding of the assignment:

Input	Output
createBook AnnaKarenina LeoTolstoy 10	

createBook 1984 GeorgeOrwell 8	
createBook AnnaKarenina Tolstoy 12	Book already exists
createUser standard nick	
createUser premium john	
createUser premium nick	User already exists
createBook CrimeAndPunishment Dostoevsky 18	
createUser standard mickey	
updatePrice AnnaKarenina 13	
readBook nick 1984	nick reading 1984 by GeorgeOrwell
readBook john 1984	john reading 1984 by GeorgeOrwell
listenBook john AnnaKarenina	john listening AnnaKarenina by LeoTolstoy
listenBook mickey AnnaKarenina	No access
subscribe mickey	
subscribe john	
subscribe mickey	User already subscribed
unsubscribe nick	User is not subscribed
updatePrice AnnaKarenina 11	mickey notified about price update for AnnaKarenina to 11 john notified about price update for AnnaKarenina to 11
unsubscribe john	
end	

***Note that in case of updatePrice command output lines should be separated by SINGLE newline character (check test cases below).**

Design Patterns

There are several requirements that you should satisfy regarding design patterns:

- You should use at least one creational design pattern **that you did not use in Assignment 3**.
- You should use at least one structural design pattern **that you did not use in Assignment 3**.
- You should use at least one behavioral design pattern **that you did not use in Assignment 3**.

In addition to provided minimal requirements you are allowed to enhance your solution with the usage of other design patterns studied during the SSAD course.

Here are examples of design patterns application:

- **Factory Design Pattern** can be used to create different types of users.
- **Proxy Design Pattern** can be used to control access to the book for different types of users.
- **Observer Design Pattern** can be used to inform users about books' price updates.

Input

- Input will consist of no more than 100 commands specified above.
- The end of input is specified by `end` command.
- Each input command has new line character at the end (including the last command).
- Standard input should be used.

Output

- Output for each command should have new line character at the end.
- Standard output should be used.

Examples

input Скопировать

```
createBook AnnaKarenina LeoTolstoy 10
createBook 1984 GeorgeOrwell 8
createBook AnnaKarenina Tolstoy 12
createUser standard nick
createUser premium john
createUser premium nick
createBook CrimeAndPunishment Dostoevsky 18
createUser standard mickey
updatePrice AnnaKarenina 13
readBook nick 1984
readBook john 1984
listenBook john AnnaKarenina
listenBook mickey AnnaKarenina
subscribe mickey
subscribe john
subscribe mickey
unsubscribe nick
updatePrice AnnaKarenina 11
unsubscribe john
end
```

output Скопировать

```
Book already exists
User already exists
nick reading 1984 by GeorgeOrwell
john reading 1984 by GeorgeOrwell
john listening AnnaKarenina by LeoTolstoy
No access
User already subscribed
User is not subscribed
mickey notified about price update for AnnaKarenina to 11
john notified about price update for AnnaKarenina to 11
```

input Скопировать

```
createBook MobyDick HermanMelville 12.99
createUser standard Ahab
subscribe Ahab
readBook Ahab MobyDick
unsubscribe Ahab
updatePrice MobyDick 14.50
end
```

output Скопировать

```
Ahab reading MobyDick by HermanMelville
```

input Скопировать

```
createBook TheHobbit JRRRTolkien 8.45
createUser premium Bilbo
createBook Silmarillion JRRRTolkien 10.99
subscribe Bilbo
listenBook Bilbo TheHobbit
readBook Bilbo Silmarillion
updatePrice TheHobbit 9.00
unsubscribe Bilbo
end
```

output Скопировать

```
Bilbo listening TheHobbit by JRRRTolkien
Bilbo reading Silmarillion by JRRRTolkien
Bilbo notified about price update for TheHobbit to 9.00
```

Privacy Policy

При поддержке

УНИВЕРСИТЕТ ИТМО