Online Java Compiler IDE

```java
import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;

// Book class representing a book in the online bookstore
class Book {
    private String title;
    private double price;

    public Book(String title, double price) {
        this.title = title;
    this.price = price;14 }

    public String getTitle() {
    return title;18
    }

    public double getPrice() {
    return price;22
    }
}

// User class representing a user in the online bookstore
class User {
    private String name;

    public User(String name) {
    this.name = name;31
    }

    public String getName() {
    return name;35}
}

// ShoppingCart class representing the shopping cart for each user
class ShoppingCart {
    private List<Book> books;

    public ShoppingCart() {
    books = new ArrayList<>();44 }

    public void addToCart(Book book) {
    books.add(book);48
    }

    public List<Book> getBooks() {
    return books;52
    }

    public double getTotalPrice() {
        double totalPrice = 0.0;
        for (Book book : books) {
        totalPrice += book.getPrice();58
        }
        return totalPrice;
    }
}

```

```java
// OnlineBookstore class representing the online bookstore
class OnlineBookstore {
    private List<Book> books;

    public OnlineBookstore(List<Book> books) {
    this.books = books;69 }

    public synchronized void purchase(User user, Book book) {
        System.out.println(user.getName() + " is purchasing: " + book.getTitle());
        // Simulate some processing time for purchasing the book
        try {
            Thread.sleep(200);
        } catch (InterruptedException e) {
        e.printStackTrace();78
        }
        System.out.println("Purchase completed for: " + book.getTitle());
    }
}

public class Main {
    public static void main(String[] args) {
        // Create some books for the online bookstore
        List<Book> books = new ArrayList<>();
        books.add(new Book("Book 1", 10.0));
        books.add(new Book("Book 2", 15.0));
        books.add(new Book("Book 3", 20.0));

        // Create the online bookstore
        OnlineBookstore bookstore = new OnlineBookstore(books);

        // Create some users
        User user1 = new User("User 1");
        User user2 = new User("User 2");
        User user3 = new User("User 3");

        // Create a thread pool to simulate multiple users
        ExecutorService executorService = Executors.newFixedThreadPool(3);

        // Simulate multiple users making purchases concurrently
        executorService.submit(() -> bookstore.purchase(user1, books.get(0)));
        executorService.submit(() -> bookstore.purchase(user2, books.get(1)));
        executorService.submit(() -> bookstore.purchase(user3, books.get(2)));

        // Shutdown the thread pool
        executorService.shutdown();
    }
}

```

Result

CPU Time: 0.22 sec(s), Memory: 36120 kilobyte(s) **compiled and executed in 1.54 sec(s)**

```
User 1 is purchasing: Book 1
Purchase completed for: Book 1
User 3 is purchasing: Book 3
Purchase completed for: Book 3
User 2 is purchasing: Book 2
Purchase completed for: Book 2
```