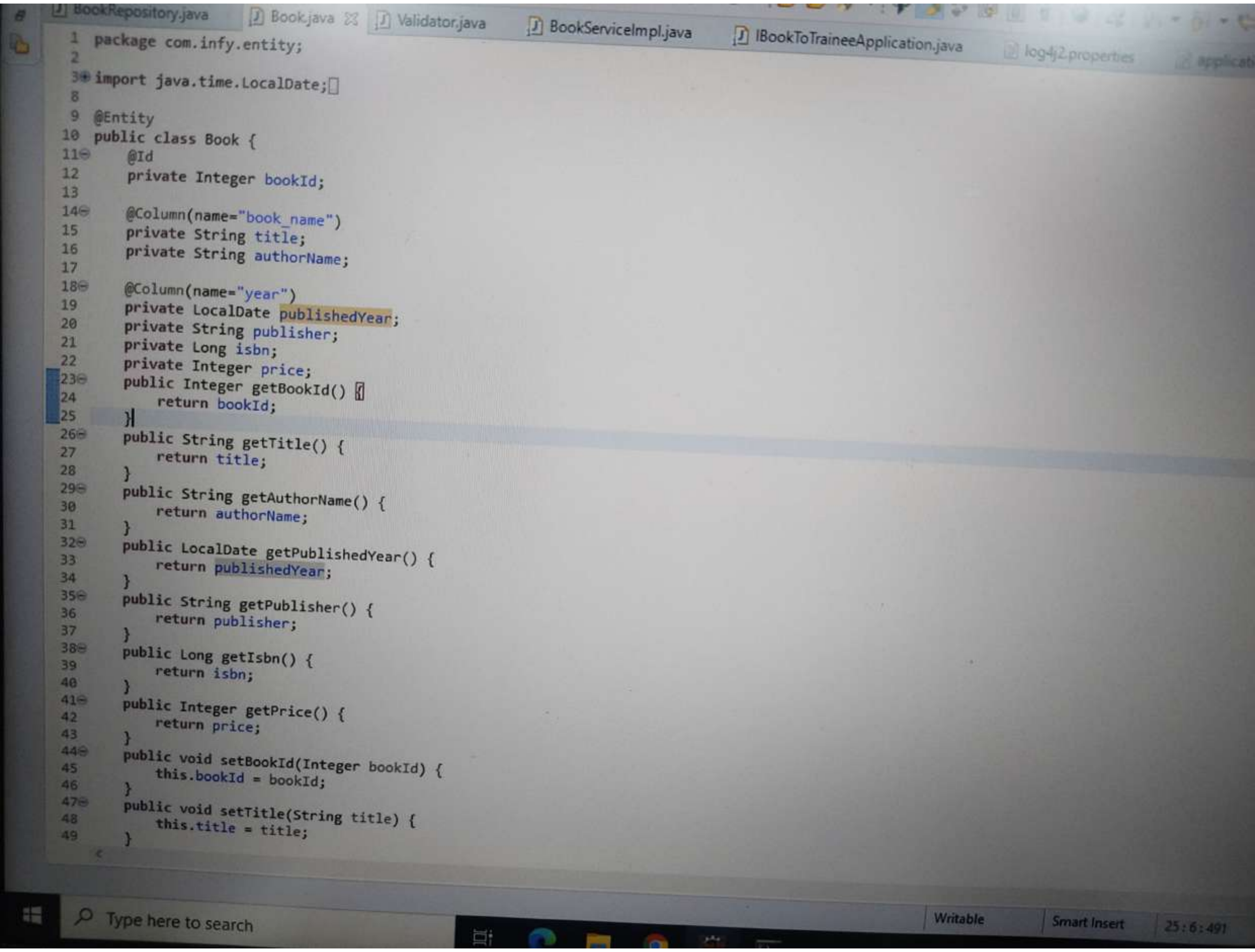
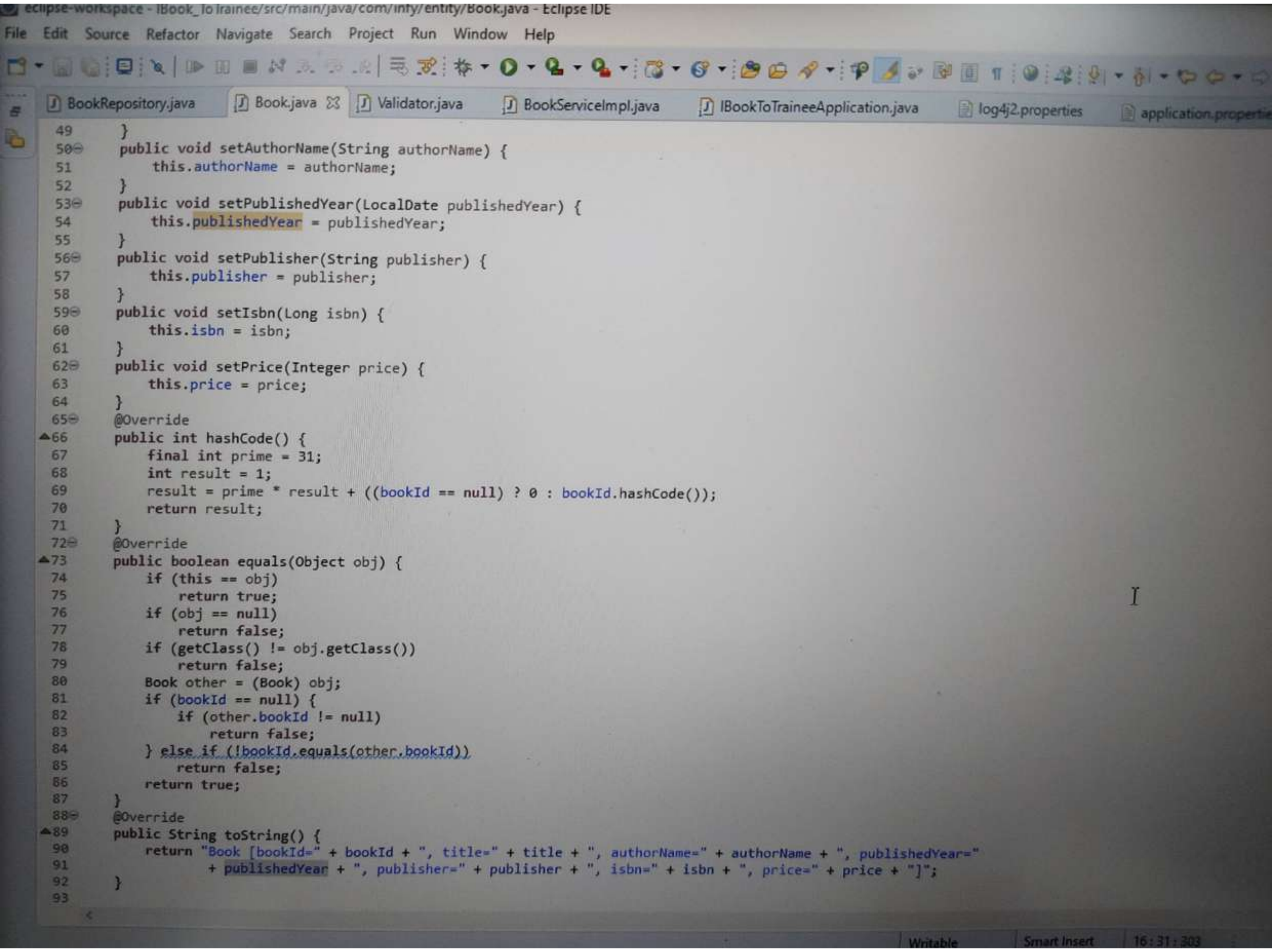
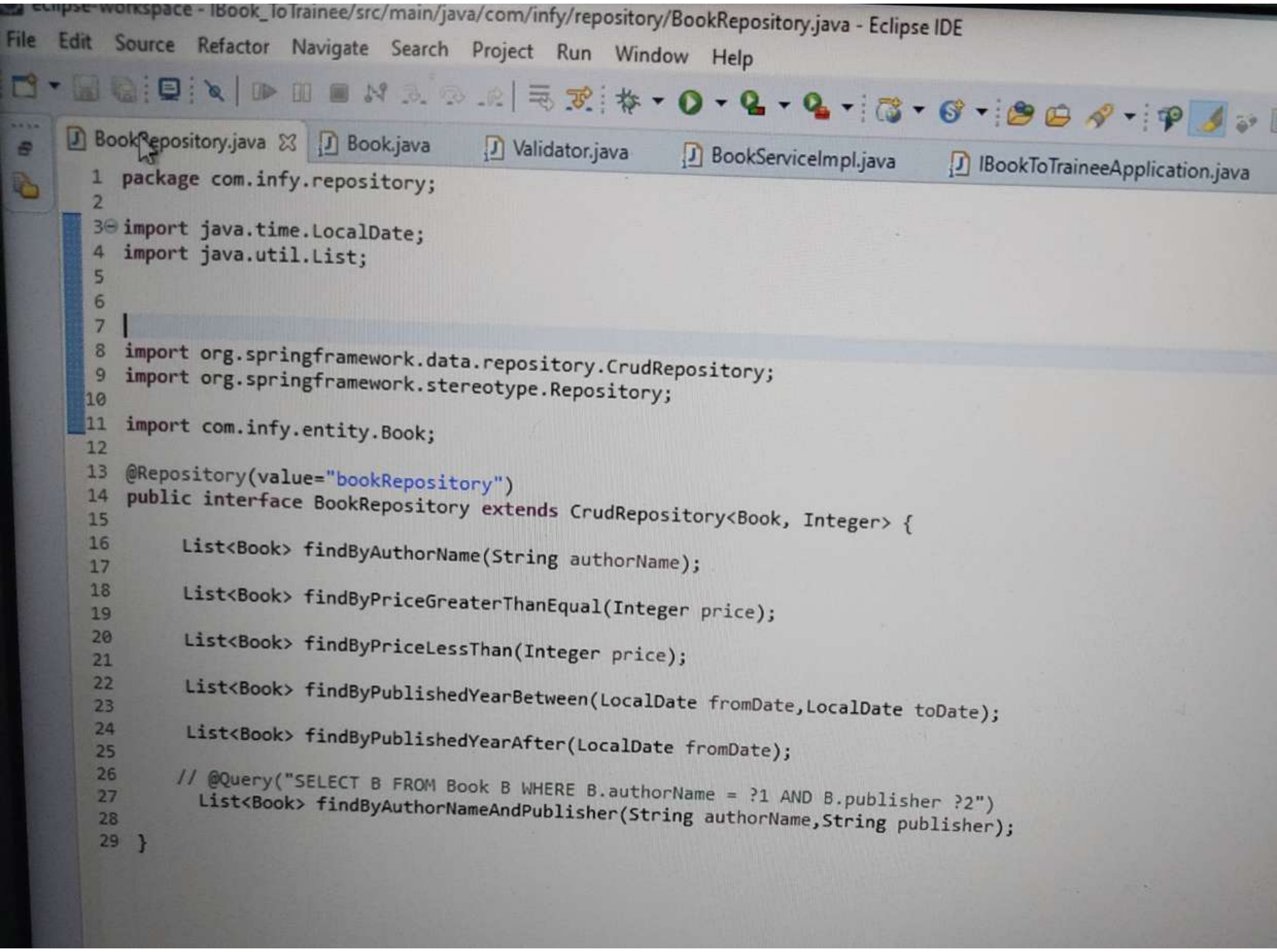


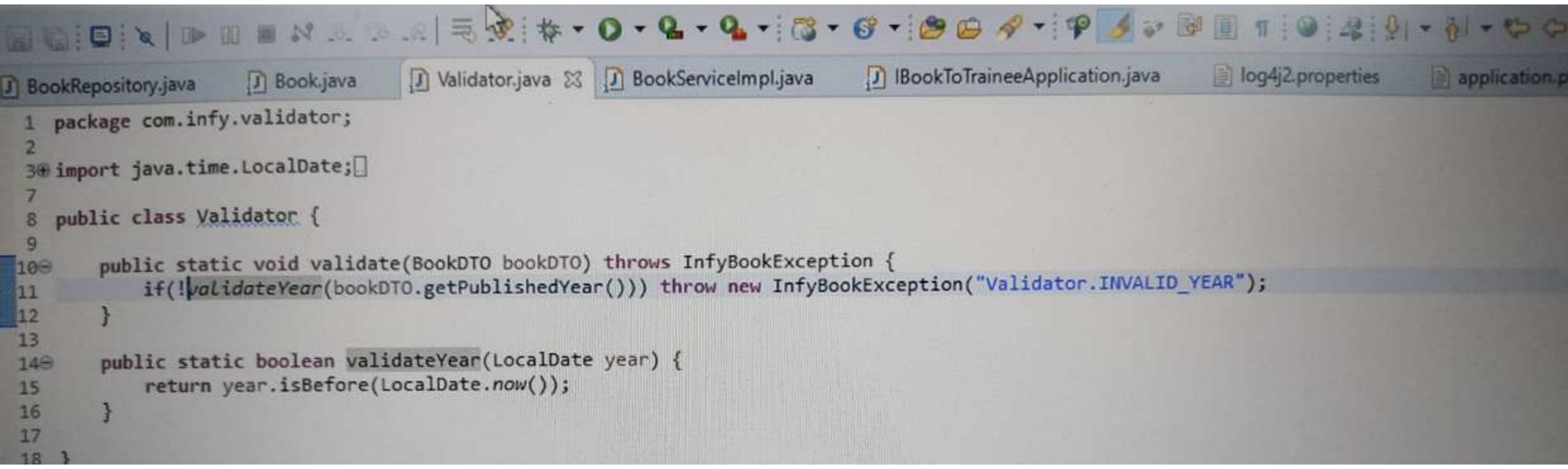
```java
package com.infy.entity;

import java.time.LocalDate;□

@Entity
public class Book {
    @Id
    private Integer bookId;

    @Column(name="book_name")
    private String title;
    private String authorName;

    @Column(name="year")
    private LocalDate publishedYear;
    private String publisher;
    private Long isbn;
    private Integer price;
    public Integer getBookId() {
        return bookId;
    }
    public String getTitle() {
        return title;
    }
    public String getAuthorName() {
        return authorName;
    }
    public LocalDate getPublishedYear() {
        return publishedYear;
    }
    public String getPublisher() {
        return publisher;
    }
    public Long getIsbn() {
        return isbn;
    }
    public Integer getPrice() {
        return price;
    }
    public void setBookId(Integer bookId) {
        this.bookId = bookId;
    }
    public void setTitle(String title) {
        this.title = title;
    }
```



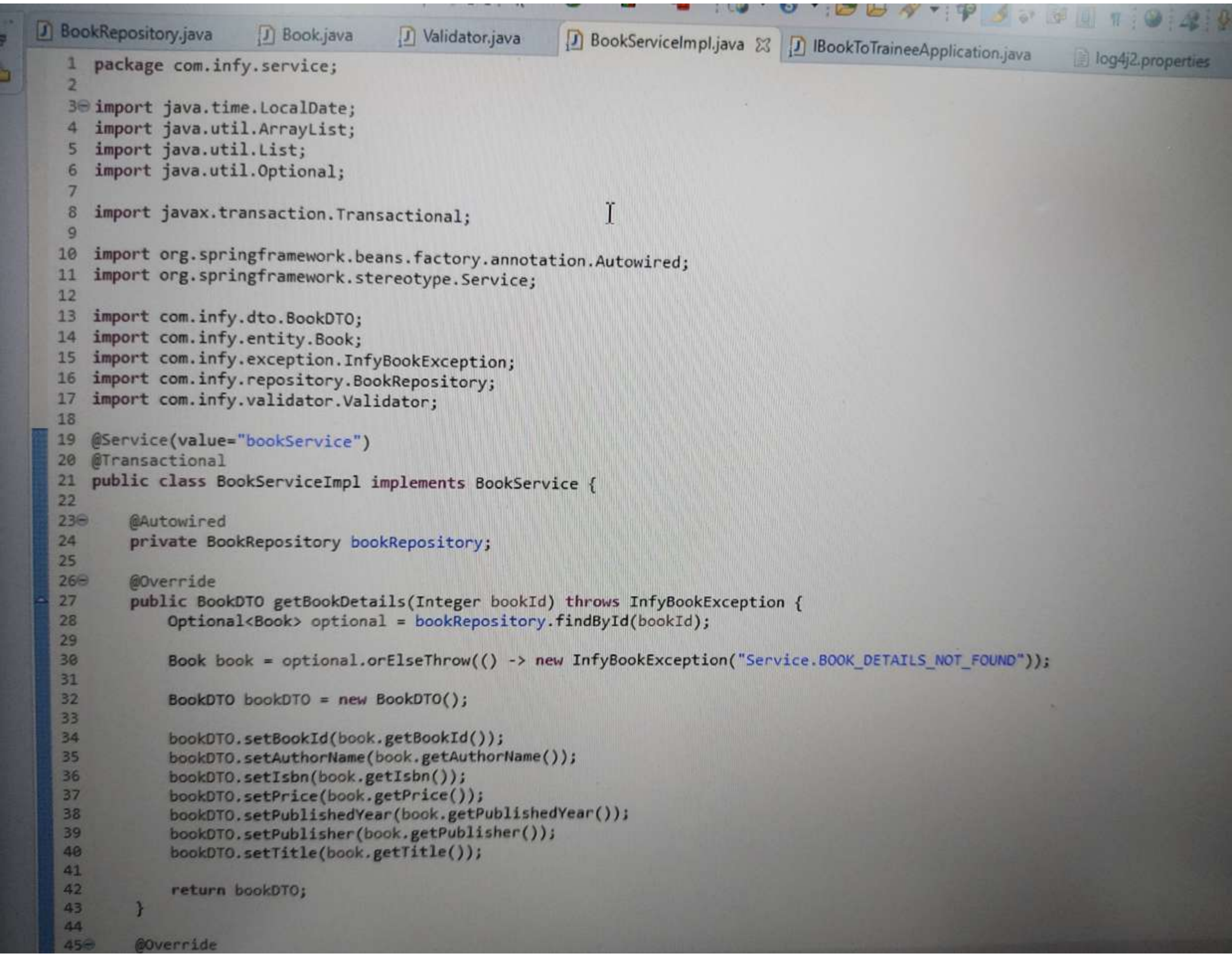
```java
    }
    public void setAuthorName(String authorName) {
        this.authorName = authorName;
    }
    public void setPublishedYear(LocalDate publishedYear) {
        this.publishedYear = publishedYear;
    }
    public void setPublisher(String publisher) {
        this.publisher = publisher;
    }
    public void setIsbn(Long isbn) {
        this.isbn = isbn;
    }
    public void setPrice(Integer price) {
        this.price = price;
    }
    @Override
    public int hashCode() {
        final int prime = 31;
        int result = 1;
        result = prime * result + ((bookId == null) ? 0 : bookId.hashCode());
        return result;
    }
    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        Book other = (Book) obj;
        if (bookId == null) {
            if (other.bookId != null)
                return false;
        } else if (!bookId.equals(other.bookId))
            return false;
        return true;
    }
    @Override
    public String toString() {
        return "Book [bookId=" + bookId + ", title=" + title + ", authorName=" + authorName + ", publishedYear="
                + publishedYear + ", publisher=" + publisher + ", isbn=" + isbn + ", price=" + price + "]";
    }
```


eclipse-workspace - IBook_ToTrainee/src/main/java/com/infy/repository/BookRepository.java - Eclipse IDE
File
Edit
Source
Refactor
Navigate
Search
Project
Run
Window
Help
BookRepository.java
Book.java
Validator.java
BookServiceImpl.java
IBookToTraineeApplication.java

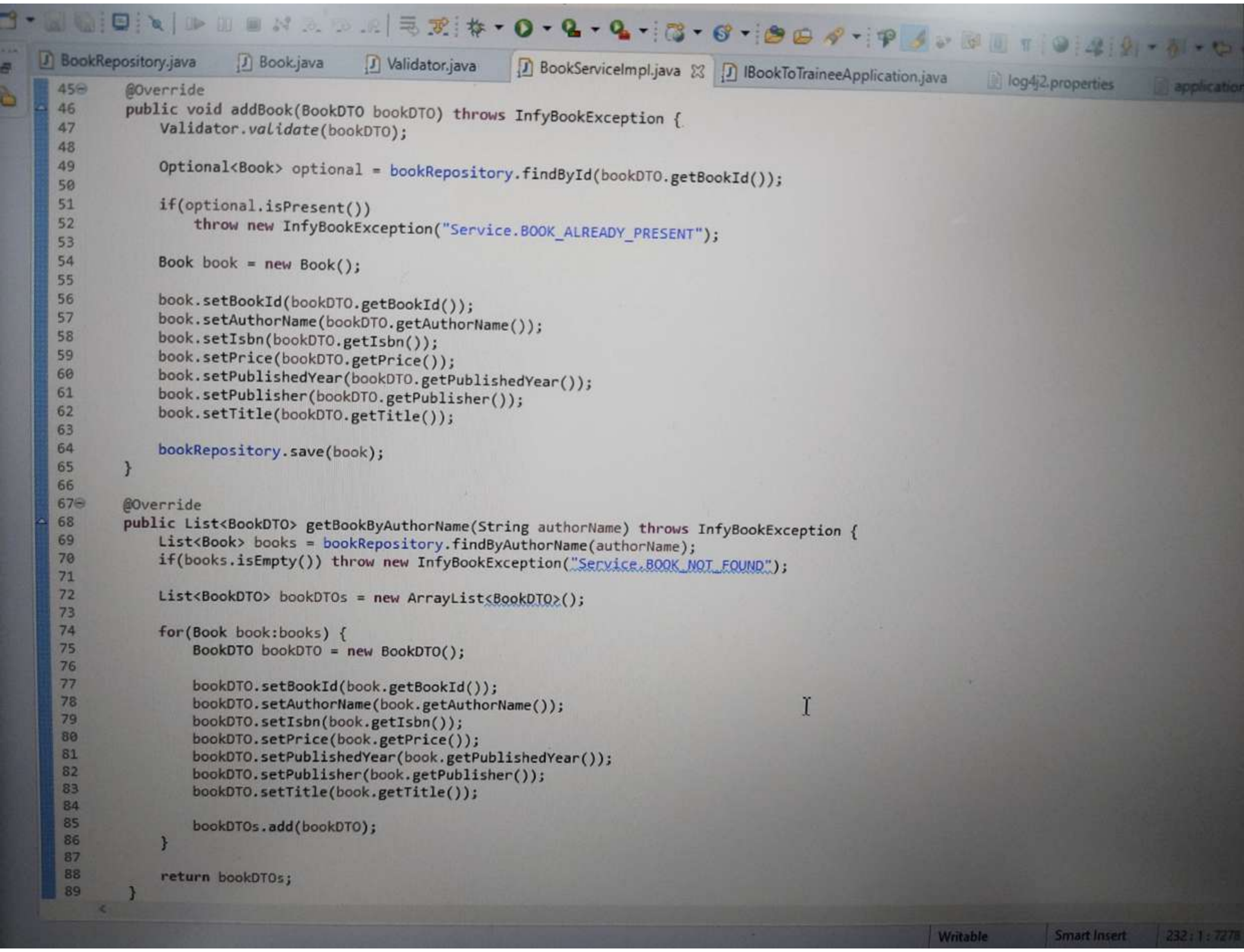
```java
package com.infy.repository;

import java.time.LocalDate;
import java.util.List;



import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;

import com.infy.entity.Book;

@Repository(value="bookRepository")
public interface BookRepository extends CrudRepository<Book, Integer> {

	List<Book> findByAuthorName(String authorName);

	List<Book> findByPriceGreaterThanEqual(Integer price);

	List<Book> findByPriceLessThan(Integer price);

	List<Book> findByPublishedYearBetween(LocalDate fromDate,LocalDate toDate);

	List<Book> findByPublishedYearAfter(LocalDate fromDate);

	// @Query("SELECT B FROM Book B WHERE B.authorName = ?1 AND B.publisher ?2")
	  List<Book> findByAuthorNameAndPublisher(String authorName,String publisher);
}
```



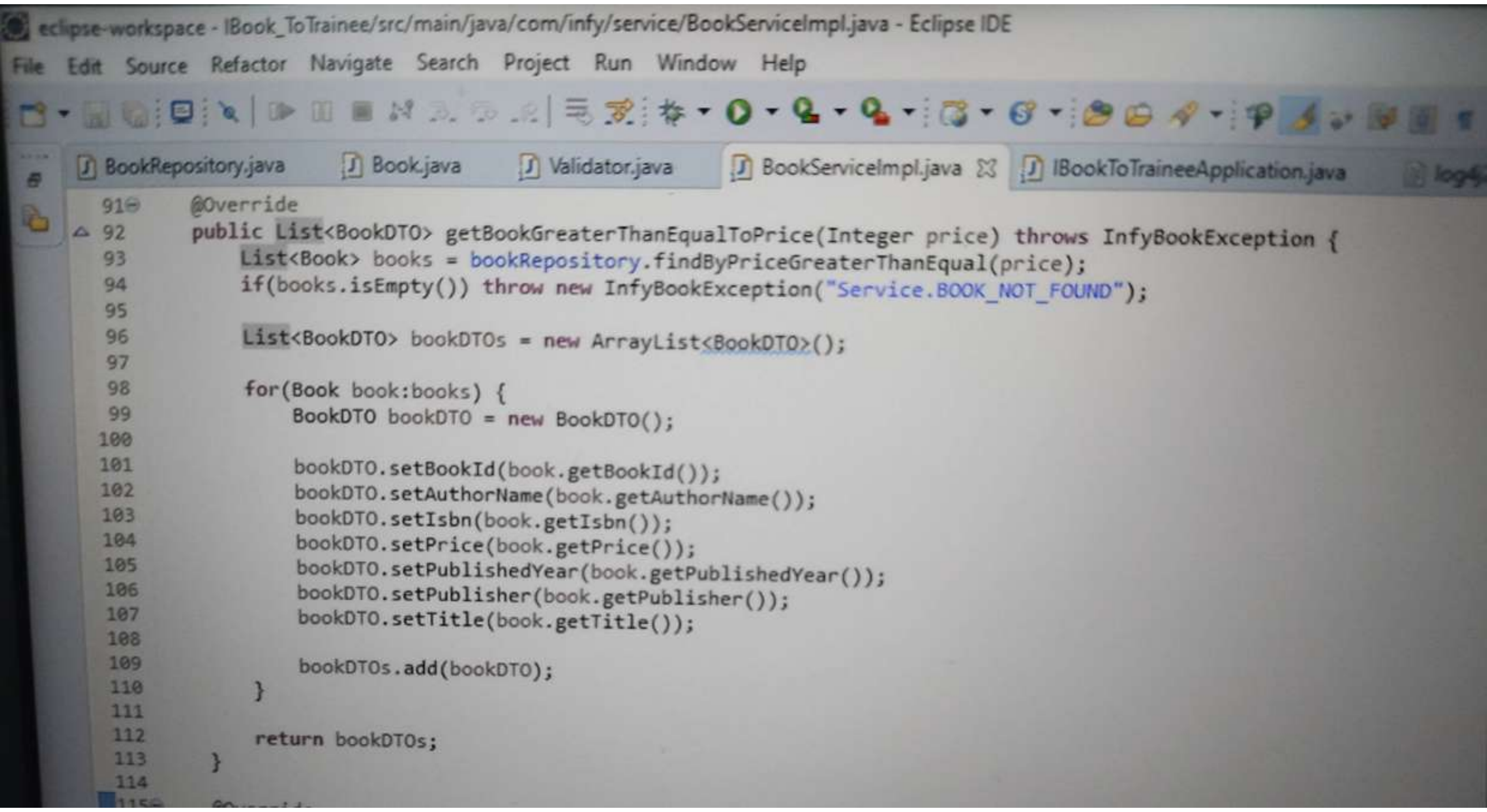
```java
package com.infy.validator;

import java.time.LocalDate;

public class Validator {

    public static void validate(BookDTO bookDTO) throws InfyBookException {
        if(!validateYear(bookDTO.getPublishedYear())) throw new InfyBookException("Validator.INVALID_YEAR");
    }

    public static boolean validateYear(LocalDate year) {
        return year.isBefore(LocalDate.now());
    }

}
```



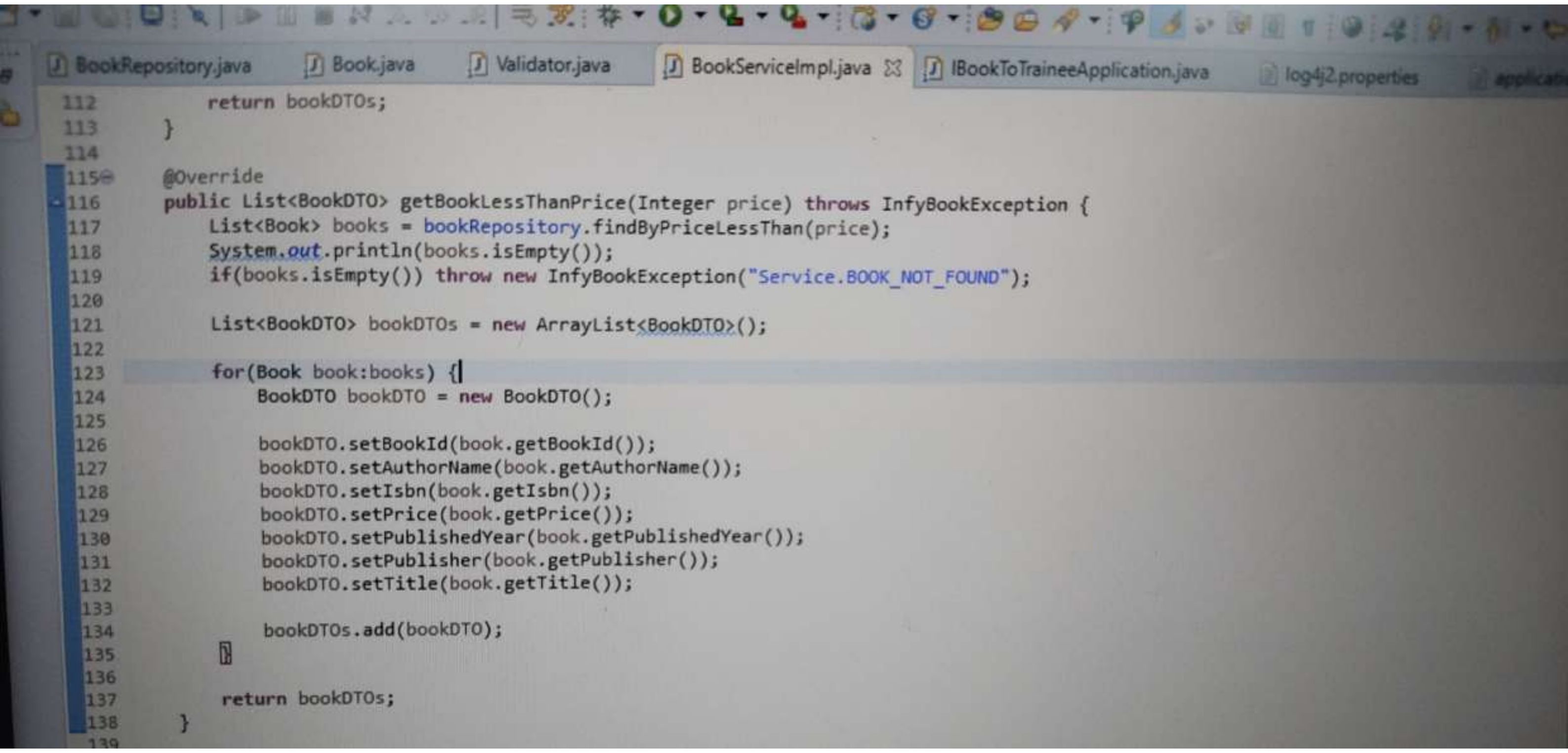
```java
package com.infy.service;

import java.time.LocalDate;
import java.util.ArrayList;
import java.util.List;
import java.util.Optional;

import javax.transaction.Transactional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.infy.dto.BookDTO;
import com.infy.entity.Book;
import com.infy.exception.InfyBookException;
import com.infy.repository.BookRepository;
import com.infy.validator.Validator;

@Service(value="bookService")
@Transactional
public class BookServiceImpl implements BookService {

    @Autowired
    private BookRepository bookRepository;

    @Override
    public BookDTO getBookDetails(Integer bookId) throws InfyBookException {
        Optional<Book> optional = bookRepository.findById(bookId);

        Book book = optional.orElseThrow(() -> new InfyBookException("Service.BOOK_DETAILS_NOT_FOUND"));

        BookDTO bookDTO = new BookDTO();

        bookDTO.setBookId(book.getBookId());
        bookDTO.setAuthorName(book.getAuthorName());
        bookDTO.setIsbn(book.getIsbn());
        bookDTO.setPrice(book.getPrice());
        bookDTO.setPublishedYear(book.getPublishedYear());
        bookDTO.setPublisher(book.getPublisher());
        bookDTO.setTitle(book.getTitle());

        return bookDTO;
    }

    @Override
```



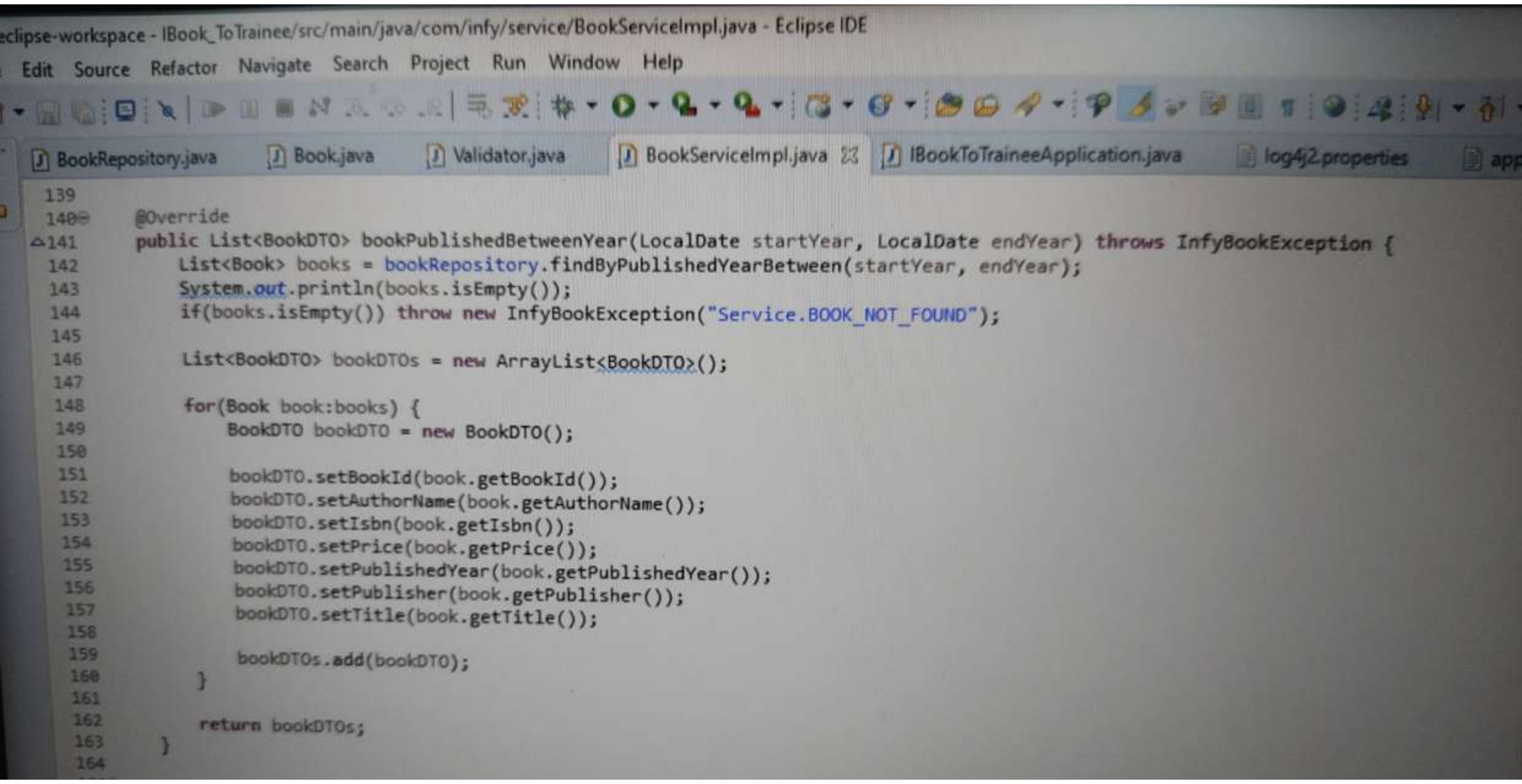
```java
    @Override
    public void addBook(BookDTO bookDTO) throws InfyBookException {
        Validator.validate(bookDTO);

        Optional<Book> optional = bookRepository.findById(bookDTO.getBookId());

        if(optional.isPresent())
            throw new InfyBookException("Service.BOOK_ALREADY_PRESENT");

        Book book = new Book();

        book.setBookId(bookDTO.getBookId());
        book.setAuthorName(bookDTO.getAuthorName());
        book.setIsbn(bookDTO.getIsbn());
        book.setPrice(bookDTO.getPrice());
        book.setPublishedYear(bookDTO.getPublishedYear());
        book.setPublisher(bookDTO.getPublisher());
        book.setTitle(bookDTO.getTitle());

        bookRepository.save(book);
    }

    @Override
    public List<BookDTO> getBookByAuthorName(String authorName) throws InfyBookException {
        List<Book> books = bookRepository.findByAuthorName(authorName);
        if(books.isEmpty()) throw new InfyBookException("Service.BOOK_NOT_FOUND");

        List<BookDTO> bookDTOs = new ArrayList<BookDTO>();

        for(Book book:books) {
            BookDTO bookDTO = new BookDTO();

            bookDTO.setBookId(book.getBookId());
            bookDTO.setAuthorName(book.getAuthorName());
            bookDTO.setIsbn(book.getIsbn());
            bookDTO.setPrice(book.getPrice());
            bookDTO.setPublishedYear(book.getPublishedYear());
            bookDTO.setPublisher(book.getPublisher());
            bookDTO.setTitle(book.getTitle());

            bookDTOs.add(bookDTO);
        }

        return bookDTOs;
    }
```


eclipse-workspace - IBook_ToTrainee/src/main/java/com/infy/service/BookServiceImpl.java - Eclipse IDE
File
Edit
Source
Refactor
Navigate
Search
Project
Run
Window
Help
BookRepository.java
Book.java
Validator.java
BookServiceImpl.java
IBookToTraineeApplication.java

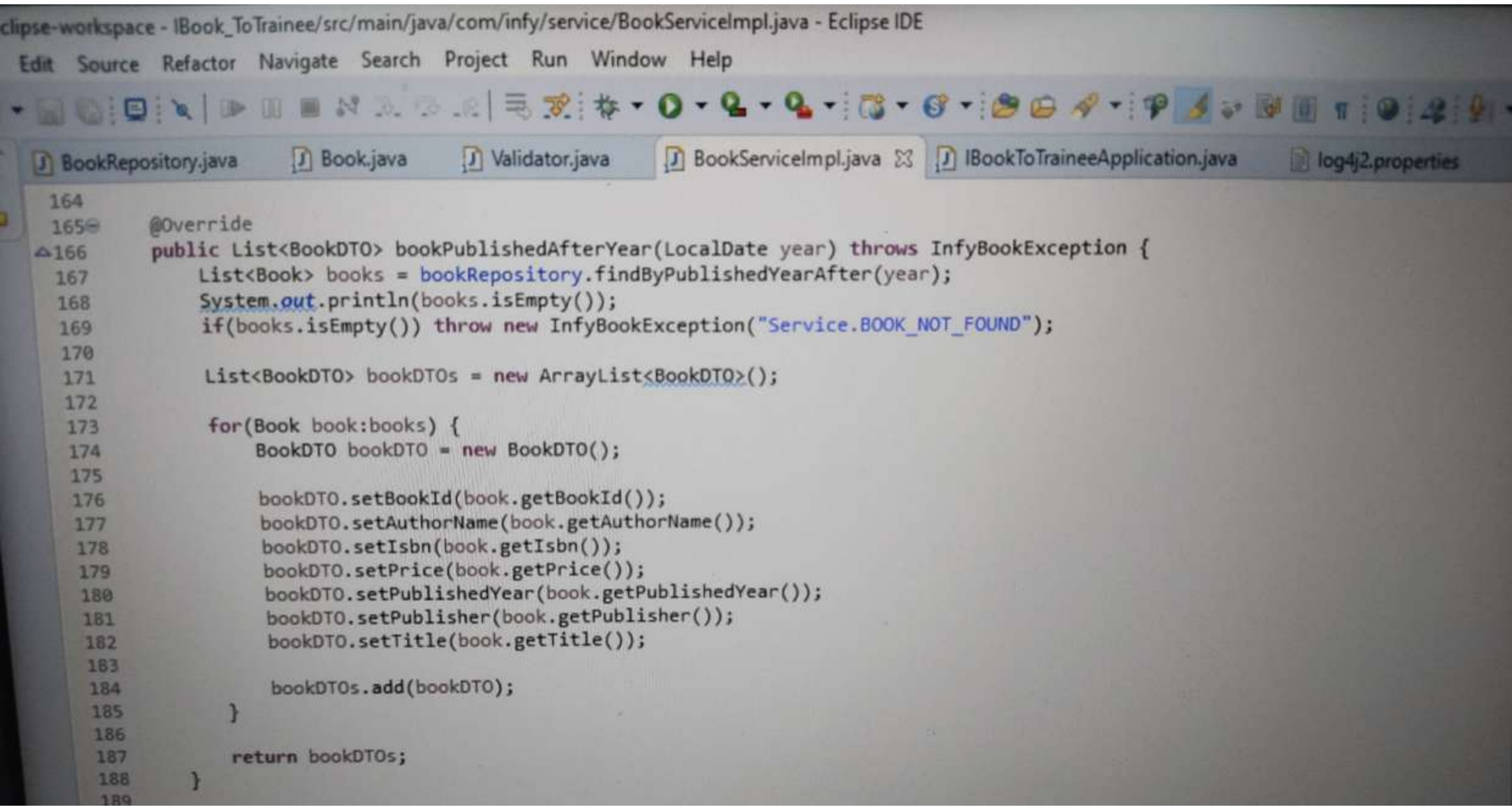
```java
    @Override
    public List<BookDTO> getBookGreaterThanEqualToPrice(Integer price) throws InfyBookException {
        List<Book> books = bookRepository.findByPriceGreaterThanEqual(price);
        if(books.isEmpty()) throw new InfyBookException("Service.BOOK_NOT_FOUND");

        List<BookDTO> bookDTOs = new ArrayList<BookDTO>();

        for(Book book:books) {
            BookDTO bookDTO = new BookDTO();

            bookDTO.setBookId(book.getBookId());
            bookDTO.setAuthorName(book.getAuthorName());
            bookDTO.setIsbn(book.getIsbn());
            bookDTO.setPrice(book.getPrice());
            bookDTO.setPublishedYear(book.getPublishedYear());
            bookDTO.setPublisher(book.getPublisher());
            bookDTO.setTitle(book.getTitle());

            bookDTOs.add(bookDTO);
        }

        return bookDTOs;
    }

```



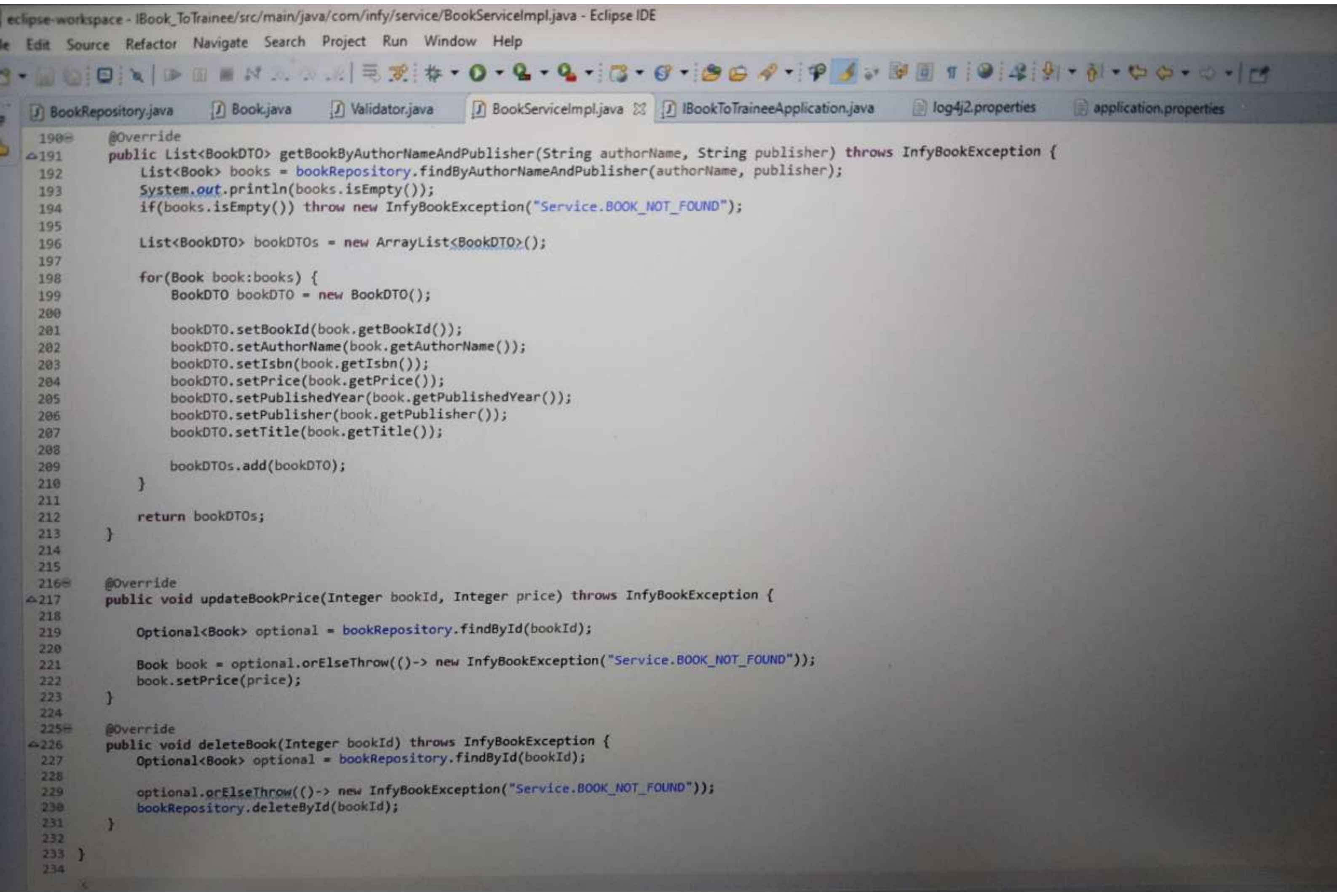
```java
        return bookDTOs;
    }

    @Override
    public List<BookDTO> getBookLessThanPrice(Integer price) throws InfyBookException {
        List<Book> books = bookRepository.findByPriceLessThan(price);
        System.out.println(books.isEmpty());
        if(books.isEmpty()) throw new InfyBookException("Service.BOOK_NOT_FOUND");

        List<BookDTO> bookDTOs = new ArrayList<BookDTO>();

        for(Book book:books) {
            BookDTO bookDTO = new BookDTO();

            bookDTO.setBookId(book.getBookId());
            bookDTO.setAuthorName(book.getAuthorName());
            bookDTO.setIsbn(book.getIsbn());
            bookDTO.setPrice(book.getPrice());
            bookDTO.setPublishedYear(book.getPublishedYear());
            bookDTO.setPublisher(book.getPublisher());
            bookDTO.setTitle(book.getTitle());

            bookDTOs.add(bookDTO);
        }

        return bookDTOs;
    }
```



```java

    @Override
    public List<BookDTO> bookPublishedBetweenYear(LocalDate startYear, LocalDate endYear) throws InfyBookException {
        List<Book> books = bookRepository.findByPublishedYearBetween(startYear, endYear);
        System.out.println(books.isEmpty());
        if(books.isEmpty()) throw new InfyBookException("Service.BOOK_NOT_FOUND");

        List<BookDTO> bookDTOs = new ArrayList<BookDTO>();

        for(Book book:books) {
            BookDTO bookDTO = new BookDTO();

            bookDTO.setBookId(book.getBookId());
            bookDTO.setAuthorName(book.getAuthorName());
            bookDTO.setIsbn(book.getIsbn());
            bookDTO.setPrice(book.getPrice());
            bookDTO.setPublishedYear(book.getPublishedYear());
            bookDTO.setPublisher(book.getPublisher());
            bookDTO.setTitle(book.getTitle());

            bookDTOs.add(bookDTO);
        }

        return bookDTOs;
    }

```



```java

    @Override
    public List<BookDTO> bookPublishedAfterYear(LocalDate year) throws InfyBookException {
        List<Book> books = bookRepository.findByPublishedYearAfter(year);
        System.out.println(books.isEmpty());
        if(books.isEmpty()) throw new InfyBookException("Service.BOOK_NOT_FOUND");

        List<BookDTO> bookDTOs = new ArrayList<BookDTO>();

        for(Book book:books) {
            BookDTO bookDTO = new BookDTO();

            bookDTO.setBookId(book.getBookId());
            bookDTO.setAuthorName(book.getAuthorName());
            bookDTO.setIsbn(book.getIsbn());
            bookDTO.setPrice(book.getPrice());
            bookDTO.setPublishedYear(book.getPublishedYear());
            bookDTO.setPublisher(book.getPublisher());
            bookDTO.setTitle(book.getTitle());

            bookDTOs.add(bookDTO);
        }

        return bookDTOs;
    }
```


eclipse-workspace - IBook_ToTrainee/src/main/java/com/infy/service/BookServiceImpl.java - Eclipse IDE
le Edit Source Refactor Navigate Search Project Run Window Help
BookRepository.java
Book.java
Validator.java
BookServiceImpl.java
IBookToTraineeApplication.java
log4j2.properties
application.properties

```java
    @Override
    public List<BookDTO> getBookByAuthorNameAndPublisher(String authorName, String publisher) throws InfyBookException {
        List<Book> books = bookRepository.findByAuthorNameAndPublisher(authorName, publisher);
        System.out.println(books.isEmpty());
        if(books.isEmpty()) throw new InfyBookException("Service.BOOK_NOT_FOUND");

        List<BookDTO> bookDTOs = new ArrayList<BookDTO>();

        for(Book book:books) {
            BookDTO bookDTO = new BookDTO();

            bookDTO.setBookId(book.getBookId());
            bookDTO.setAuthorName(book.getAuthorName());
            bookDTO.setIsbn(book.getIsbn());
            bookDTO.setPrice(book.getPrice());
            bookDTO.setPublishedYear(book.getPublishedYear());
            bookDTO.setPublisher(book.getPublisher());
            bookDTO.setTitle(book.getTitle());

            bookDTOs.add(bookDTO);
        }

        return bookDTOs;
    }


    @Override
    public void updateBookPrice(Integer bookId, Integer price) throws InfyBookException {

        Optional<Book> optional = bookRepository.findById(bookId);

        Book book = optional.orElseThrow(()-> new InfyBookException("Service.BOOK_NOT_FOUND"));
        book.setPrice(price);
    }

    @Override
    public void deleteBook(Integer bookId) throws InfyBookException {
        Optional<Book> optional = bookRepository.findById(bookId);

        optional.orElseThrow(()-> new InfyBookException("Service.BOOK_NOT_FOUND"));
        bookRepository.deleteById(bookId);
    }

}
```