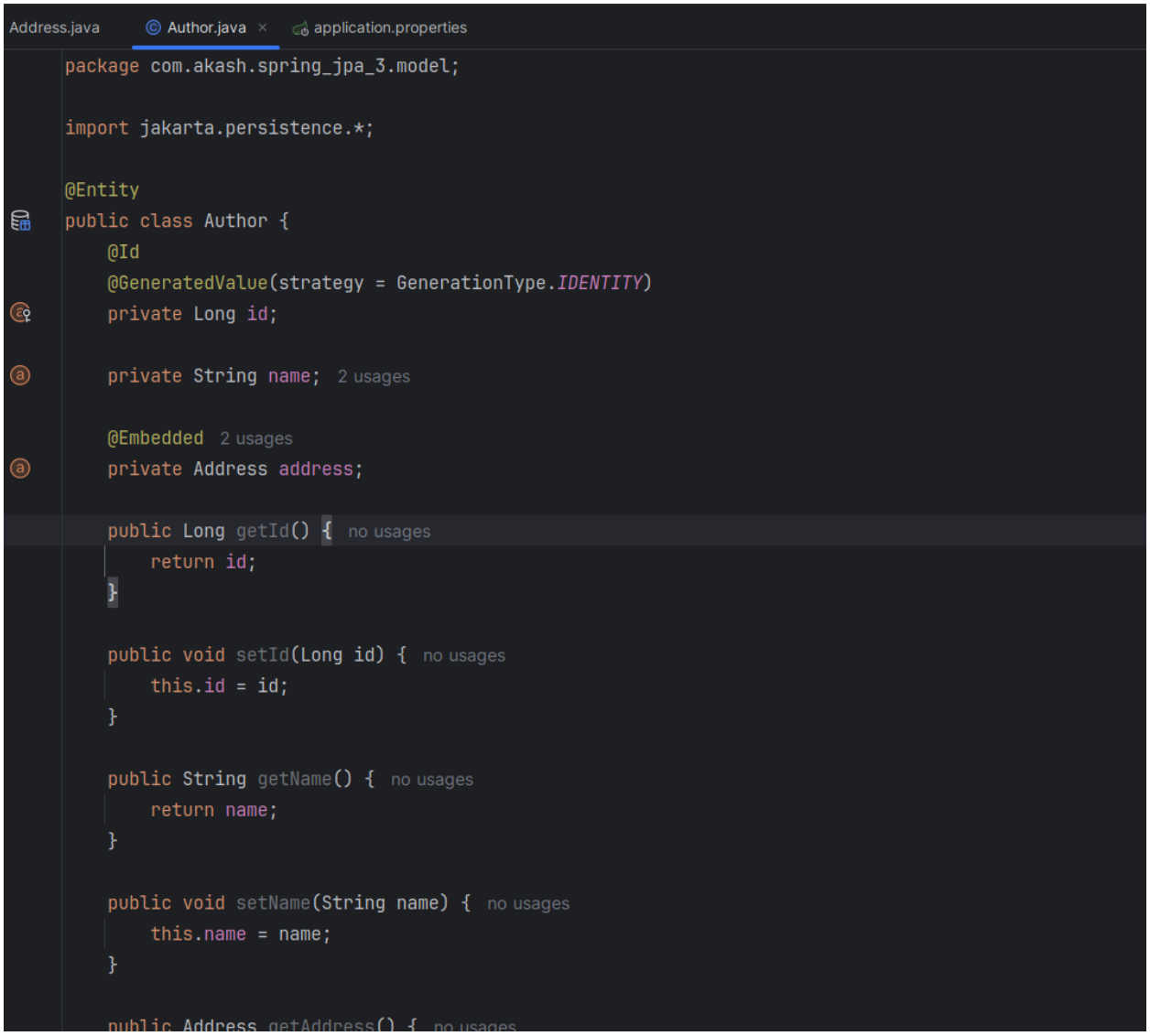
Assignment - Spring Data JPA - Part 3

Q1)Create a class Address for Author with instance variables streetNumber, location, State.

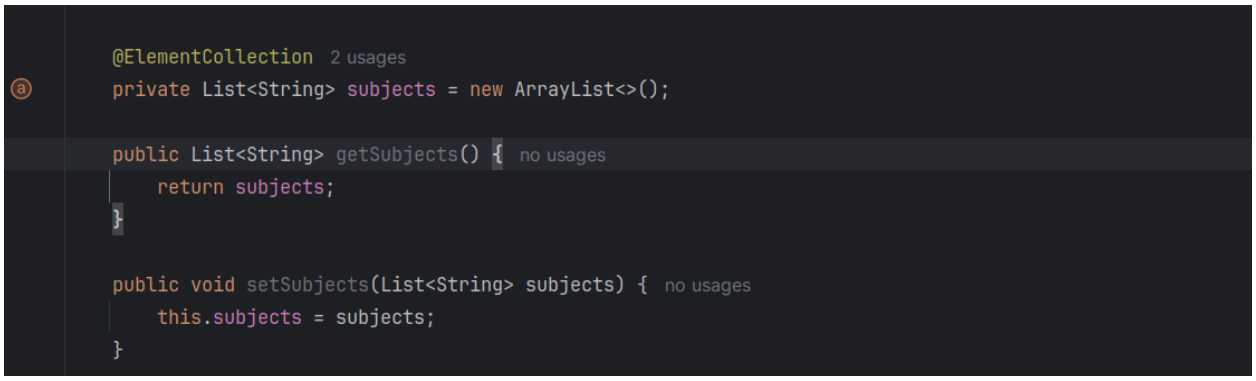
Address.java

```java
package com.akash.spring_jpa_3.model;

public class Address {  no usages
    private String streetNumber;  2 usages
    private String location;  2 usages
    private String state;  2 usages

    public String getStreetNumber() {  no usages
        return streetNumber;
    }

    public void setStreetNumber(String streetNumber) {  no usages
        this.streetNumber = streetNumber;
    }

    public String getLocation() {  no usages
        return location;
    }

    public void setLocation(String location) {  no usages
        this.location = location;
    }

    public String getState() {  no usages
        return state;
    }

    public void setState(String state) {  no usages
        this.state = state;
    }
}
```

Q2)Create instance variable of Address class inside Author class and save it as embedded object.

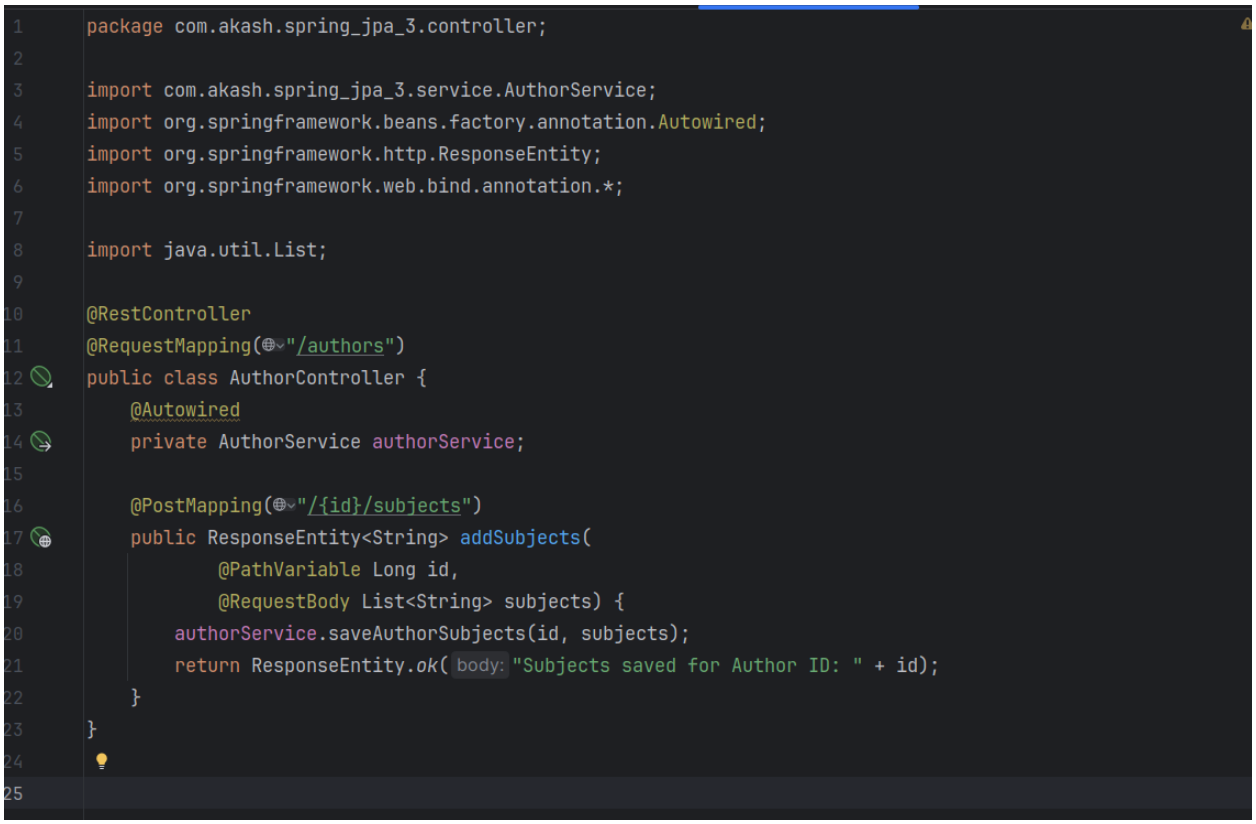


Q3)Introduce a List of subjects for author.



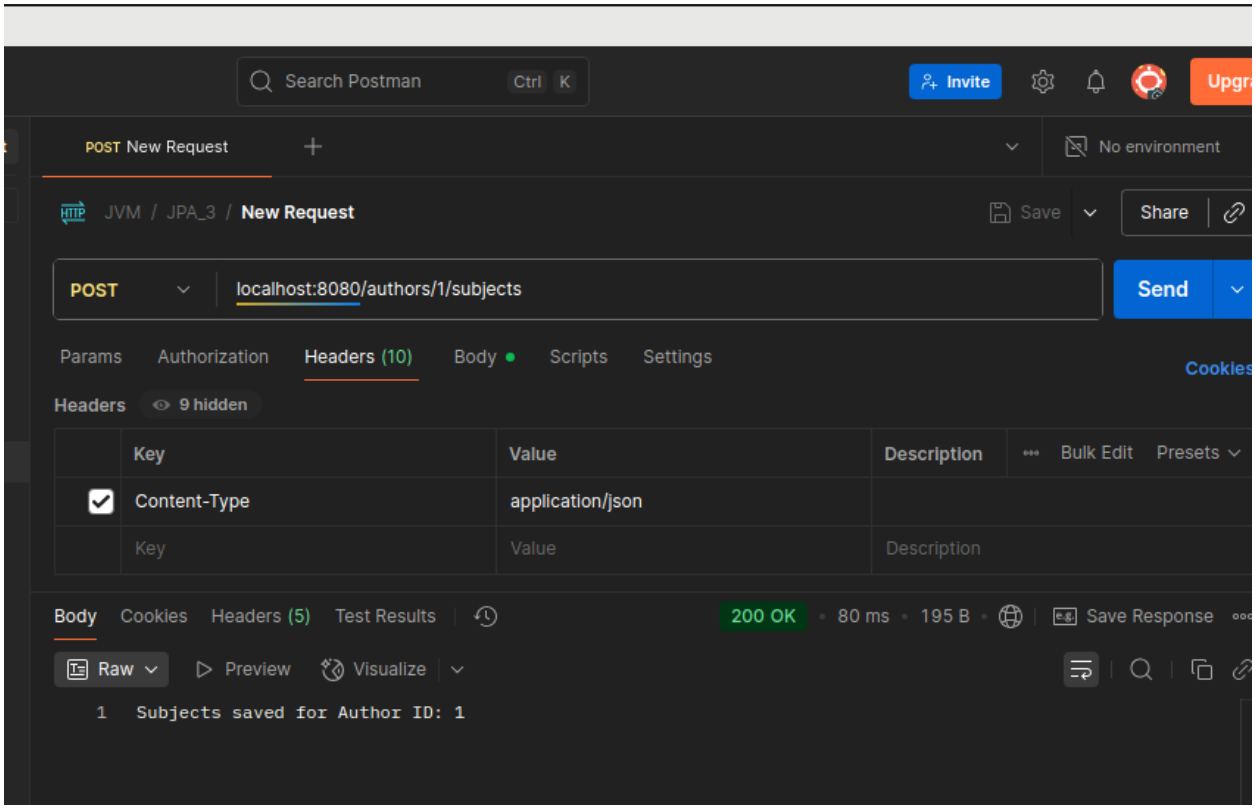
```
@ElementCollection  2 usages
private List<String> subjects = new ArrayList<>();

public List<String> getSubjects() {  no usages
    return subjects;
}

public void setSubjects(List<String> subjects) {  no usages
    this.subjects = subjects;
}
```

Q4)Persist 3 subjects for each author.

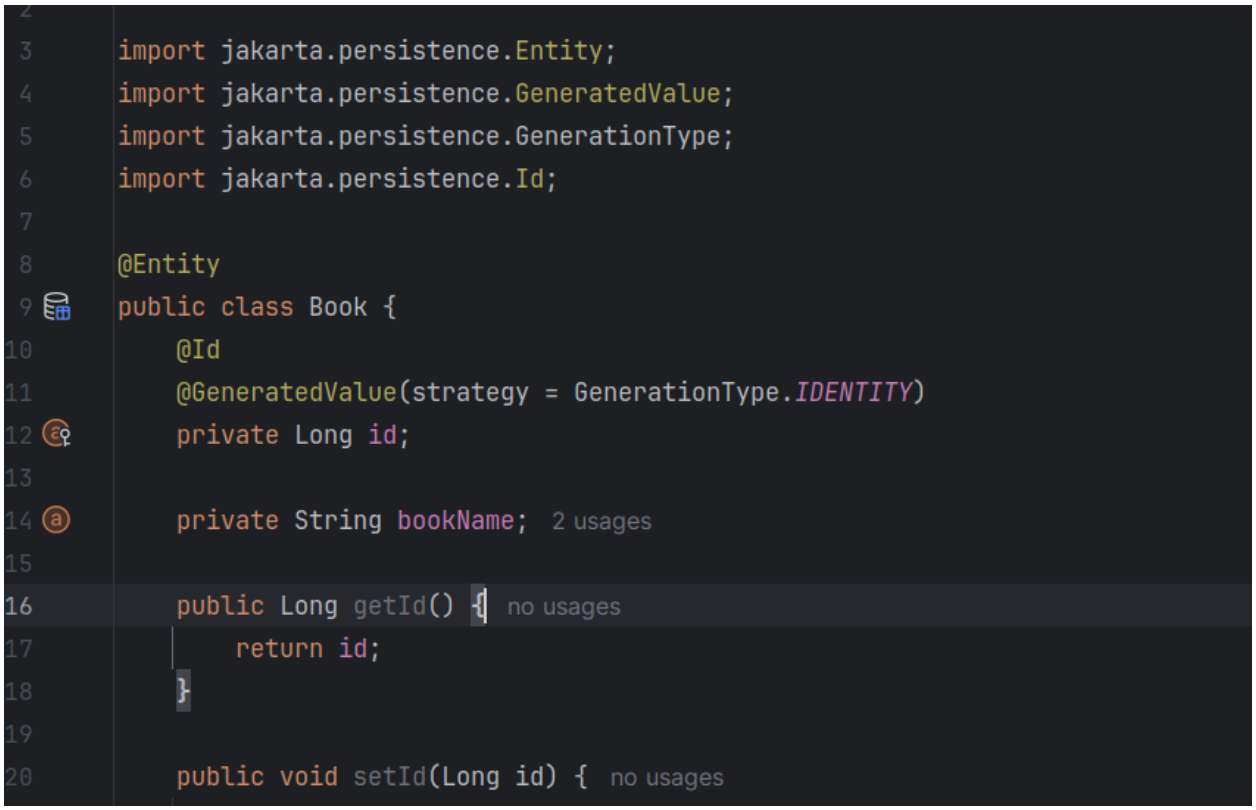


```java
package com.akash.spring_jpa_3.controller;

import com.akash.spring_jpa_3.service.AuthorService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/authors")
public class AuthorController {
    @Autowired
    private AuthorService authorService;

    @PostMapping("/{id}/subjects")
    public ResponseEntity<String> addSubjects(
            @PathVariable Long id,
            @RequestBody List<String> subjects) {
        authorService.saveAuthorSubjects(id, subjects);
        return ResponseEntity.ok( body: "Subjects saved for Author ID: " + id);
    }
}

```

```java
package com.akash.spring_jpa_3.service;

import com.akash.spring_jpa_3.model.Author;
import com.akash.spring_jpa_3.repository.AuthorRepo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;

@Service 2 usages
public class AuthorService {
    @Autowired
    private AuthorRepo authorRepo;

    @Transactional 1 usage
    public void saveAuthorSubjects(Long Id, List<String> subjects) {
        Author author = authorRepo.findById(Id).orElseThrow(() -> new RuntimeException("Author not found"));

        author.setSubjects(subjects);
        authorRepo.save(author);
    }
}
```

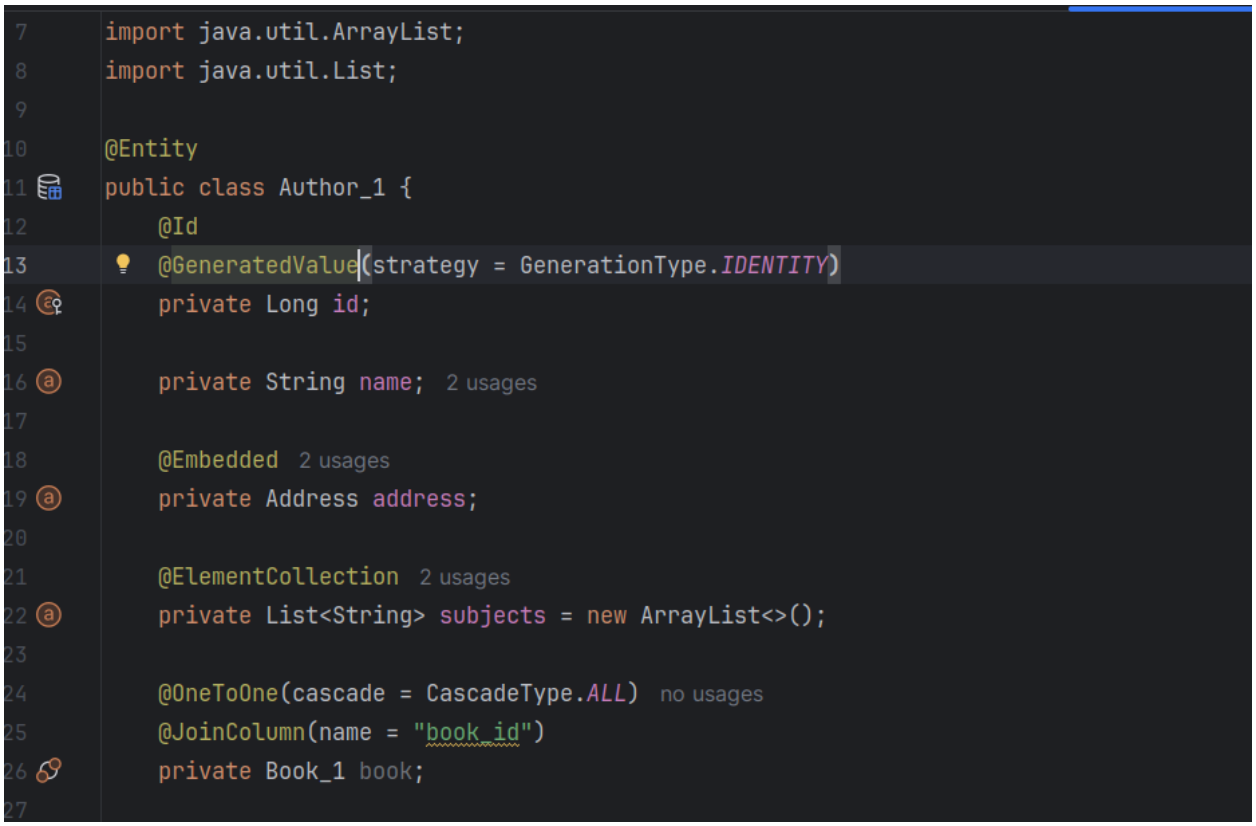


Q5)Create an Entity book with an instance variable bookName.



```java
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;

@Entity
public class Book {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String bookName;  2 usages

    public Long getId() {  no usages
        return id;
    }

    public void setId(Long id) {  no usages
```

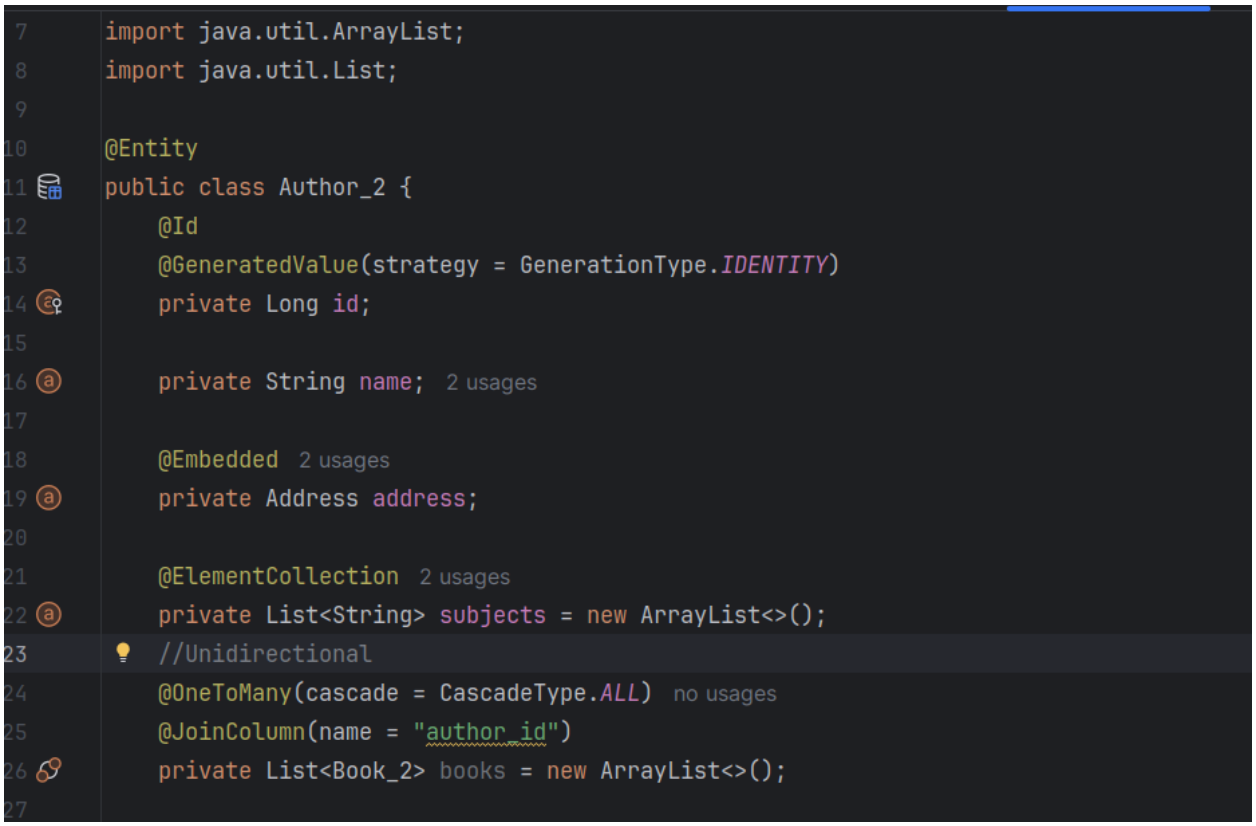
Q6)Implement One to One mapping between Author and Book.



```java
import java.util.ArrayList;
import java.util.List;

@Entity
public class Author_1 {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;  2 usages

    @Embedded  2 usages
    private Address address;

    @ElementCollection  2 usages
    private List<String> subjects = new ArrayList<>();

    @OneToOne(cascade = CascadeType.ALL)  no usages
    @JoinColumn(name = "book_id")
    private Book_1 book;

```

Q7)Implement One to Many Mapping between Author and Book(Unidirectional, BiDirectional and without additional table) and implement cascade save.

a) Unidirectional



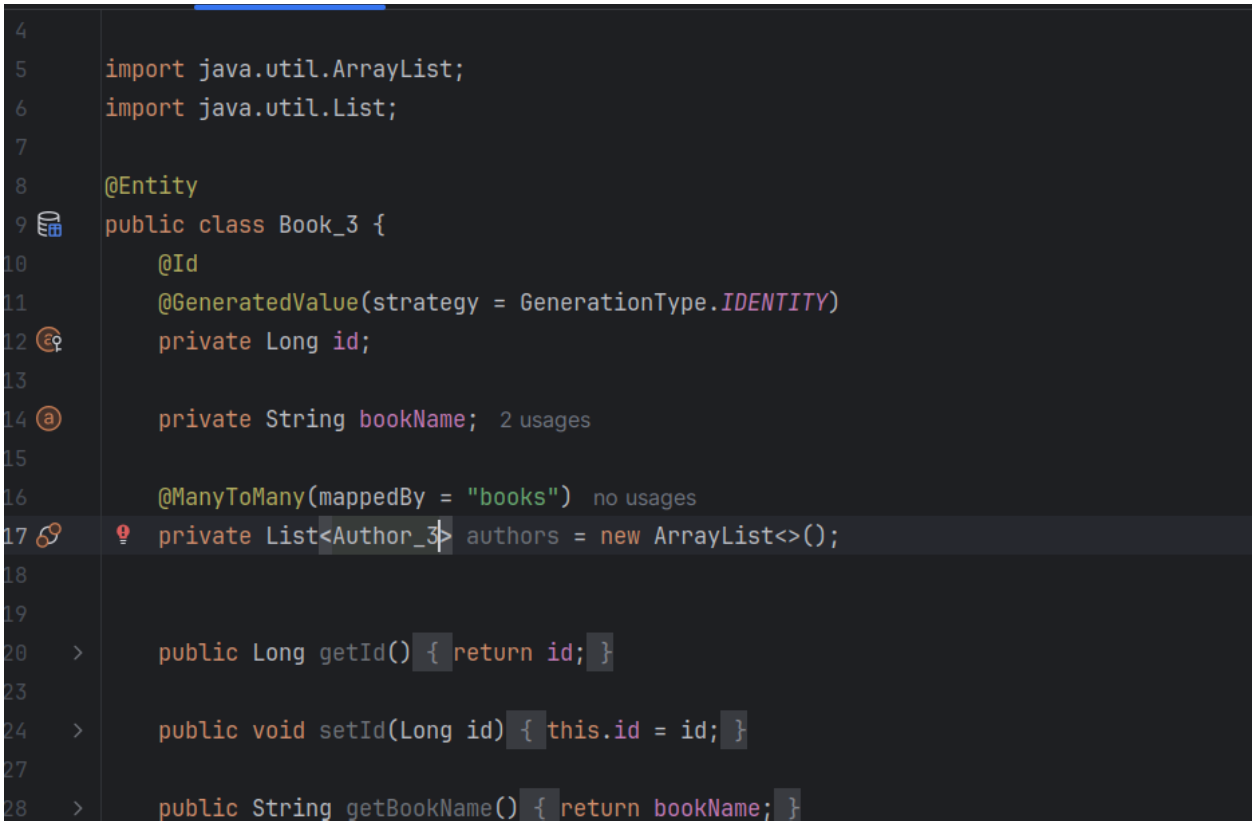
```java
import java.util.ArrayList;
import java.util.List;

@Entity
public class Author_2 {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;  2 usages

    @Embedded  2 usages
    private Address address;

    @ElementCollection  2 usages
    private List<String> subjects = new ArrayList<>();
    //Unidirectional
    @OneToMany(cascade = CascadeType.ALL)  no usages
    @JoinColumn(name = "author_id")
    private List<Book_2> books = new ArrayList<>();

```

b) Bidirectional

```java
package com.akash.spring_jpa_3.model.mapping.one_to_many;

import jakarta.persistence.*;

@Entity
public class Book_2 {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String bookName;  2 usages


    @ManyToOne  no usages
    @JoinColumn(name = "author_id")
    private Author_2 author;

    public Long getId() { return id; }

    public void setId(Long id) { this.id = id; }

```

```java
@Entity
public class Author_2 {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;  2 usages

    @Embedded  2 usages
    private Address address;

    @ElementCollection  2 usages
    private List<String> subjects = new ArrayList<>();
//    //Unidirectional
//    @OneToMany(cascade = CascadeType.ALL)
//    @JoinColumn(name = "author_id")
//    private List<Book_2> books = new ArrayList<>();

    @OneToMany(mappedBy = "author", cascade = CascadeType.ALL)  no usages
    private List<Book_2> books = new ArrayList<>();

```

Q8)Implement Many to Many Mapping between Author and Book.



```java
import java.util.ArrayList;
import java.util.List;

@Entity
public class Book_3 {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String bookName;  2 usages

    @ManyToMany(mappedBy = "books")  no usages
    private List<Author_3> authors = new ArrayList<>();


    public Long getId() { return id; }

    public void setId(Long id) { this.id = id; }

    public String getBookName() { return bookName; }
```

```java
package com.akash.spring_jpa_3.model.mapping.many_to_many;


import ...

@Entity
public class Author_3 {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;  2 usages

    @Embedded  2 usages
    private Address address;

    @ElementCollection  2 usages
    private List<String> subjects = new ArrayList<>();

    @ManyToMany(cascade = CascadeType.ALL)  no usages
    @JoinTable(
            name = "author_book",
            joinColumns = @JoinColumn(name = "author_id"),
            inverseJoinColumns = @JoinColumn(name = "book_id")
    )
    private List<Book_3> books = new ArrayList<>();

```

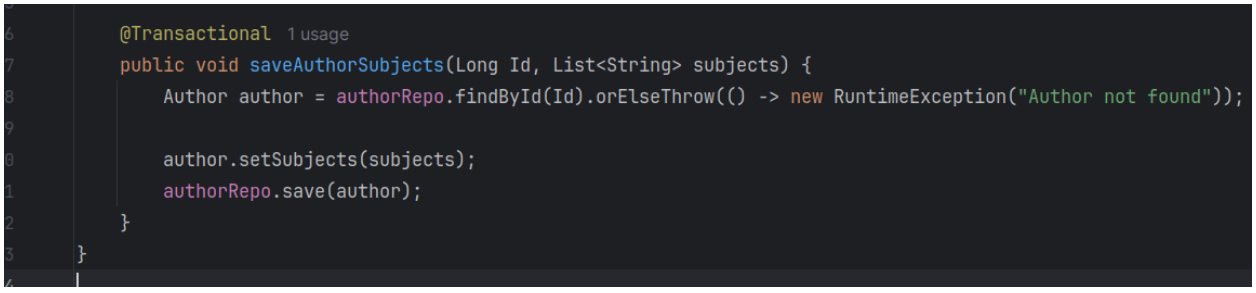
Q9)Which method on the session object can be used to remove an object from the cache?

Ans -> session.evict(object);

Q10)What does @transactional annotation do?

Ans - >

- @Transactional ensures that the method runs inside a database transaction.
- If any operation inside the method fails, all changes will be rolled back.
- Without it, partial data might be persisted.



```
    @Transactional  1 usage
    public void saveAuthorSubjects(Long Id, List<String> subjects) {
        Author author = authorRepo.findById(Id).orElseThrow(() -> new RuntimeException("Author not found"));

        author.setSubjects(subjects);
        authorRepo.save(author);
    }
}
```