**Task SpringData solution**

**Task:** Implement AuthorDao with findByName, find author books (if not familiar with JPA introduce booksAmount)

**1) Create Author in model**

@Entity  
**public class** Author {  
 **private int id**;  
 **private** String **name**;  
 **private** List<Book> **books** = **new** ArrayList<>();

**public** Author() {  
}  
  
**public** Author(String name) {  
 **this**.**name** = name;  
}

// id, name getters and setters

@OneToMany(mappedBy = **"author"**, cascade = CascadeType.***ALL***)  
**public** List<Book> getBooks() {  
 **return books**;  
}  
  
**public void** setBooks(List<Book> books) {  
 **this**.**books** = books;  
}  
  
**public void** addBook(Book book) {  
 **books**.add(book);  
 book.setAuthor(**this**);  
}

}

**2) Add to Book**

**private** Author **author**;

@ManyToOne  
**public** Author getAuthor() {  
 **return** author;  
}  
  
**public void** setAuthor(Author author) {  
 **this**.author = author;  
}

**2) Create AuthorDao**

**public interface** AuthorDao **extends** JpaRepository<Author, Integer>{  
 **public** Author findByName(String name);  
}

**3) Add method to BookDao**

**public** List<Book> findByAuthorName(String name);

**4) Create test method in BookDaoImplTest**

@Test  
**public void** testAuthor() {  
 Author author = **new** Author(**"Sonkin"**);  
 author.addBook(  
 **new** Book(**"Spring"**, Date.*valueOf*(LocalDate.*of*(2015, 5, 1))));  
 **authorDao**.save(author);  
  
 List<Book> books = **bookDao**.findByAuthorName(**"Sonkin"**);  
 books.stream().forEach(System.***out***::println);  
}