In Task 4, we were required to implement a package `finalexam.task4` that includes several classes and an interface to simulate a simple bookstore system. The main goal of this task is to demonstrate the ability to create, manage, and persist objects representing a bookstore and its authors, and to ensure that the bookstore class adheres to a given interface for legal entities.

Package Structure

The package `finalexam.task4` consists of the following classes and interfaces:

1. `LegalEntity` (Interface)
2. `Author` (Class)
3. `Bookstore` (Class)
4. `BookstoreTester` (Class)

Each class and interface serves a specific purpose, which will be described in detail below.

Class and Interface Descriptions

5. **LegalEntity (Interface)**
    - **Purpose**: This interface defines the contract for any legal entity, which in this context, is implemented by the `Bookstore` class.
    - **Methods**:
        - `String getAddress()`: Returns the address of the legal entity.
        - `String getVatNumber()`: Returns the VAT number of the legal entity.

The `LegalEntity` interface ensures that any class implementing it will provide these two crucial pieces of information.

6. **Author (Class)**
    - **Purpose**: This class represents an author with basic details such as name and nationality.
    - **Fields**:
        - `String name`: The name of the author.
        - `String nationality`: The nationality of the author.
    - **Constructor**:

        - `Author(String name, String nationality)`: Initializes an author with the given name and nationality.
    - **Methods**:
        - Getters and setters for `name` and `nationality`.
        - `String toString()`: Provides a string representation of the author.

The `Author` class is used to store information about individual authors that are associated with the bookstore.

7. **Bookstore (Class)**
    - **Purpose**: This class represents a bookstore that implements the `LegalEntity` interface and maintains a list of authors.
    - **Fields**:
        - `String address`: The address of the bookstore.
        - `String vatNumber`: The VAT number of the bookstore.
        - `List<Author> authors`: A list to store the authors associated with the bookstore.
    - **Constructor**:
        - `Bookstore(String address, String vatNumber)`: Initializes the bookstore with the given address and VAT number.
    - **Methods**:
        - Implementation of `getAddress()` and `getVatNumber()` from `LegalEntity`.
        - Methods to manage the authors list:
            - `void addAuthor(Author author)`: Adds an author to the list.
            - `boolean removeAuthor(Author author)`: Removes an author from the list.
            - `List<Author> getAuthors()`: Returns a copy of the authors list.
        - Methods to save and load authors list to and from a file:
            - `void saveAuthorsToFile(String filename)`: Saves the authors list to a file.
            - `void loadAuthorsFromFile(String filename)`: Loads the authors list from a file.
        - `String toString()`: Provides a string representation of the bookstore.

The `Bookstore` class is central to this task. It not only manages its list of authors but also ensures that it adheres to the `LegalEntity` interface by providing the required address and VAT number.

8. **BookstoreTester (Class)**
    - **Purpose**: This class is used to test the functionality of the `Bookstore` class.
    - **Methods**:
        - `main(String[] args)`: The main method to run the tests.
            - It creates a `Bookstore` instance.
            - Adds authors to the bookstore.
            - Prints the current list of authors.
            - Saves the list of authors to a file.
            - Clears the current list and reloads it from the file.
            - Prints the reloaded list of authors.
            - Removes an author and prints the updated list.

The `BookstoreTester` class serves as a demonstration of how the `Bookstore` class and its methods can be used in a real-world scenario.

Conclusion

The `finalexam.task4` package provides a simple but comprehensive example of managing a bookstore and its authors. It showcases the use of interfaces to define a contract for legal entities, the implementation of classes to represent real-world objects, and methods for managing and persisting data. Through this task, i have demonstrated the ability to create and interact with objects in a structured and meaningful way.