Task#4 Report

General Information

This report details the implementation of a Bookstore Management System, developed as part of the final exam task 4. The system is designed to manage a bookstore's operations, particularly focusing on maintaining relationships with publishers. It demonstrates object-oriented programming principles, file I/O operations, and adherence to interface contracts.

The system is implemented in Java and consists of three main classes: `LegalEntity`, `Publisher`, and `Bookstore`, along with a `BookstoreTester` class for demonstration purposes. The implementation showcases the use of interfaces, serialization for data persistence, and basic data management operations.

LegalEntity Interface

Goal: To define a contract for entities with legal standing, ensuring they have methods to retrieve address and VAT number.

Functionality:

- Declares two methods:
    - `getAddress()`: Returns the entity's address.
    - `getVatNumber()`: Returns the entity's VAT (Value Added Tax) number.

Publisher Class

```
public class Publisher implements LegalEntity, Serializable {
   private static final long serialVersionUID = 1L;

   private String name;
   private String address;
   private String vatNumber;

   public Publisher(String name, String address, String vatNumber) {
       this.name = name;
       this.address = address;
       this.vatNumber = vatNumber;
   }
```

Goal: To represent a book publisher with essential details.

Functionality:

- Implements the `LegalEntity` interface and `Serializable` interface.
- Maintains publisher details: name, address, and VAT number.
- Provides getters and setters for all fields.
- Overrides `toString()` method for string representation.
- Implements `equals()` and `hashCode()` methods based on the VAT number for proper comparison and use in collections.

BookstoreClass

Goal: To represent a bookstore and manage its associated publishers.

Functionality:

- Implements the `LegalEntity` interface.
- Maintains bookstore details: name, address, and VAT number.
- Manages a list of `Publisher` objects.
- Provides methods to add and remove publishers.
- Implements file I/O operations to save and load the list of publishers:
    - `savePublishersToFile(String fileName)`: Serializes and saves the publishers list to a file.
    - `loadPublishersFromFile(String fileName)`: Deserializes and loads the publishers list from a file.
- Overrides `toString()` method to provide a comprehensive string representation of the bookstore and its publishers.

```
public void savePublishersToFile(String fileName) throws IOException {
    try (ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream(fileName))) {
        oos.writeObject(publishers);
    }
}

@SuppressWarnings("unchecked")
public void loadPublishersFromFile(String fileName) throws IOException,
ClassNotFoundException {
    try (ObjectInputStream ois = new ObjectInputStream(new
FileInputStream(fileName))) {
        publishers = (List<Publisher>) ois.readObject();
    }
}

@Override
public String toString() {
    StringBuilder sb = new StringBuilder("Bookstore: " + name + "\n");
```

```
    sb.append("Address: ").append(address).append("\n");
    sb.append("VAT Number: ").append(vatNumber).append("\n");
    sb.append("Publishers:\n");
    for (Publisher p : publishers) {
        sb.append(p).append("\n");
    }
    return sb.toString();
}
```

BookstoreTester Class

Goal: To demonstrate the functionality of the Bookstore Management System.

Functionality:

- Creates a `Bookstore` instance and populates it with sample `Publisher` objects.
- Demonstrates adding publishers to the bookstore.
- Shows how to save the publishers' data to a file and load it back.
- Illustrates the process of removing a publisher from the bookstore.
- Prints the bookstore's state at various stages to showcase the system's operations.

```
public class BookstoreTester {
    public static void main(String[] args) {
        Bookstore bookstore = new Bookstore("My Bookstore", "123 Book St,
Reading, RG1 1AB", "GB123456789");

        Publisher p1 = new Publisher("Penguin Books", "80 Strand, London, WC2R
0RL", "GB876543210");
        Publisher p2 = new Publisher("HarperCollins", "1 London Bridge St,
London, SE1 9GF", "GB987654321");
        Publisher p3 = new Publisher("Random House", "20 Vauxhall Bridge Rd,
London, SW1V 2SA", "GB234567890");

        bookstore.addPublisher(p1);
        bookstore.addPublisher(p2);
        bookstore.addPublisher(p3);
```

Conclusion

The Bookstore Management System provides a robust foundation for managing a bookstore's relationships with publishers. It demonstrates key programming concepts such as interface implementation, data encapsulation, file I/O operations, and basic data management. The system is designed with extensibility in mind, allowing for future enhancements such as inventory management or order processing.