Introduction

In this task, we developed a package `finalexam.task4` for an agricultural company. The package includes classes to represent the company's structure, manage its crops, and ensure legal compliance. The main objective was to demonstrate object-oriented programming principles by implementing appropriate classes and methods, and to incorporate file I/O operations to save and load data. The package consists of four primary components: `LegalEntity` interface, `Crop` class, `AgriculturalCompany` class, and `CompanyTester` class.

Class Descriptions

1. LegalEntity Interface

Functionality: The `LegalEntity` interface defines the contract for any class that represents a legal entity. It ensures that implementing classes provide methods to retrieve the entity's address and VAT number.

Goal: The primary goal of this interface is to standardize the way legal entities are defined within the application, ensuring consistency and allowing for polymorphism. Any class that represents a legal entity must provide concrete implementations for the methods declared in this interface.

```
package finalexam.task4;

public interface LegalEntity {
    String getAddress();
    String getVatNumber();
}
```

2. Crop Class

Functionality: The `Crop` class represents a crop grown by the agricultural company. It includes fields for the crop's name, type, and yield, along with getter and setter methods for these fields.

Goal: The main goal of the `Crop` class is to encapsulate the properties of a crop, providing a clear structure for storing and retrieving crop-related data. This class also implements `Serializable` to allow crop objects to be saved to and loaded from a file.

```java
package finalexam.task4;

import java.io.Serializable;

public class Crop implements Serializable {
    private static final long serialVersionUID = 1L;

    private String name;
    private String type;
    private double yield;

    public Crop(String name, String type, double yield) {
        this.name = name;
        this.type = type;
        this.yield = yield;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getType() {
        return type;
    }
```

```
    public void setType(String type) {
        this.type = type;
    }

    public double getYield() {
        return yield;
    }

    public void setYield(double yield) {
        this.yield = yield;
    }

    @Override
    public String toString() {
        return "Crop{" +
                "name='" + name + '\'' +
                ", type='" + type + '\'' +
                ", yield=" + yield +
                '}';
    }
}
```

3. AgriculturalCompany Class

Functionality: The `AgriculturalCompany` class represents the agricultural company itself. It implements the `LegalEntity` interface and manages a list of `Crop` objects. The class includes methods to add, delete, and retrieve crops, as well as to save the list of crops to a file and load it from a file.

Goal: The goal of the `AgriculturalCompany` class is to encapsulate the company's details and operations related to crop management. It ensures the company adheres to the legal entity requirements by implementing the `LegalEntity` interface, and provides functionality to manage crop data effectively, including persistence through file I/O.

```java
package finalexam.task4;

import java.io.*;
import java.util.ArrayList;
import java.util.List;

public class AgriculturalCompany implements LegalEntity {
    private String address;
    private String vatNumber;
    private List<Crop> crops = new ArrayList<>();

    public AgriculturalCompany(String address, String vatNumber) {
        this.address = address;
        this.vatNumber = vatNumber;
    }

    @Override
    public String getAddress() {
        return address;
    }

    @Override
    public String getVatNumber() {
        return vatNumber;
    }

    public void addCrop(Crop crop) {
        crops.add(crop);
    }
```

```
    public void saveCropsToFile(String filename) throws IOException {
        try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(filename
            oos.writeObject(crops);
        }
    }

    public void loadCropsFromFile(String filename) throws IOException, ClassNotFoundExcept
        try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(filename)))
            crops = (List<Crop>) ois.readObject();
        }
    }
}
```

4. CompanyTester Class

Functionality: The `CompanyTester` class is a test class that demonstrates the usage of the `AgriculturalCompany` class. It creates an instance of `AgriculturalCompany`, adds and deletes crops, and tests saving and loading the crop list to and from a file.

Goal: The primary goal of the `CompanyTester` class is to verify that the `AgriculturalCompany` and `Crop` classes function as expected. It serves as a demonstration of how to use these classes in practice, ensuring that all methods work

correctly and that data persistence is properly handled.

```java
package finalexam.task4;

public class CompanyTester {
    public static void main(String[] args) {
        AgriculturalCompany company = new AgriculturalCompany("123 Farm Lane", "VAT123456

        Crop crop1 = new Crop("Wheat", "Grain", 20.5);
        Crop crop2 = new Crop("Corn", "Grain", 30.0);

        company.addCrop(crop1);
        company.addCrop(crop2);

        System.out.println("Crops after adding:");
        company.getCrops().forEach(System.out::println);

        try {
            company.saveCropsToFile("crops.dat");
            company.getCrops().clear();
            System.out.println("Crops after clearing:");
            company.getCrops().forEach(System.out::println);

            company.loadCropsFromFile("crops.dat");
            System.out.println("Crops after loading from file:");
            company.getCrops().forEach(System.out::println);
        } catch (IOException | ClassNotFoundException e) {
            e.printStackTrace();
        }

        company.deleteCrop(crop1);

        System.out.println("Crops after deleting Wheat:");
        company.getCrops().forEach(System.out::println);
    }
}
```

Conclusion

The `finalexam.task4` package effectively demonstrates object-oriented programming principles by defining classes and interfaces with clear responsibilities and encapsulation.

The `AgriculturalCompany` class, along with its supporting classes, provides a robust framework for managing crop data and ensuring compliance with legal requirements. The `CompanyTester` class verifies the functionality of the implemented methods and showcases how these classes can be utilized in practice.