Cow Farm Management System in Java

Description:
Our project revolves around the creation of a Cow Farm Management System implemented in Java. This system plays a crucial role in efficiently managing a cow farm's operations. We've developed a basic version with fundamental functionalities to showcase its core capabilities.

Features:
1. Storage for Cows: Our system provides a robust storage mechanism for cows, ensuring organised management and easy access to cow-related information.
2. Addition of Cows: Users can seamlessly add new cows to the farm, facilitating the growth and maintenance of the cow population.
3. Removal of Cows: Removing cows from the farm is straightforward, allowing for efficient management of the herd.
4. Printing Farm Information: The system enables users to print detailed information about the cows on the farm directly to the console, aiding in farm management and analysis.

Cow Farm Management System Structure:
To realise our objectives, we've structured the system using the following classes:
1. Cow: Represents an individual cow with properties such as name and breed.
2. CowFarm: The core of our system, responsible for managing cows and farm operations.
3. FarmTester: A tester class designed to validate the functionality of our Cow Farm Management System.

Cow Class:
Cow.java: This class represents a Cow with properties like name and breed. It provides getter and setter methods for these properties and a toString method to represent the Cow object as a string.

```java
package midterm.giorgi_kitiashvili_2.task2;

public class FarmTester {  new *
    public static void main(String[] args) {  new *
        // Creating a CowFarm
        CowFarm farm = new CowFarm();

        // Creating some cows
        Cow cow1 = new Cow();
        cow1.setName("Bessie");
        cow1.setBreed("Holstein");

        Cow cow2 = new Cow();
        cow2.setName("MooMoo");
        cow2.setBreed("Jersey");

        // Adding cows to the farm
        farm.addCow(cow1);
        farm.addCow(cow2);

        // Removing a cow from the farm
        farm.removeCow(cow1);

        // Printing information about cows in the farm
        farm.printStorage();
    }
}
```

CowFarm Class:

CowFarm.java: This class represents a CowFarm which maintains a list of Cow objects. It provides methods to add a cow to the farm, remove a cow from the farm, and print the current state of the farm.

```java
package midterm.giorgi_kitiashvili_2.task2;

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

public class CowFarm {  2 usages  new *
    private final List<Cow> storage;  5 usages

    public CowFarm() {  1 usage  new *
        this.storage = new ArrayList<>();
    }

    public void addCow(Cow cow) {  2 usages  new *
        storage.add(cow);
    }

    public boolean removeCow(Cow cow) {  1 usage  new *
        return storage.remove(cow);
    }

    public void printStorage() {  1 usage  new *
        if (storage.isEmpty()) {
            System.out.println("The farm is empty");
        } else {
            System.out.println("Cows in the farm:");
            for (Cow c : storage) {
                System.out.println(c.toString());
            }
        }
    }
}
```

FarmTester Class :
FarmTester.java: This class contains the main method which is the entry point of the program. It creates a CowFarm, adds some Cow objects to it, removes a Cow, and prints the state of the farm.

```java
package midterm.giorgi_kitiashvili_2.task2;

public class FarmTester {  new *
    public static void main(String[] args) {  new *
        // Creating a CowFarm
        CowFarm farm = new CowFarm();

        // Creating some cows
        Cow cow1 = new Cow();
        cow1.setName("Bessie");
        cow1.setBreed("Holstein");

        Cow cow2 = new Cow();
        cow2.setName("MooMoo");
        cow2.setBreed("Jersey");

        // Adding cows to the farm
        farm.addCow(cow1);
        farm.addCow(cow2);

        // Removing a cow from the farm
        farm.removeCow(cow1);

        // Printing information about cows in the farm
        farm.printStorage();
    }
}
```

Conclusion
Our implementation of the Cow Farm Management System in Java demonstrates the effectiveness and versatility of the software in efficiently managing farm resources. With features for adding, removing, and viewing cow information, our system serves as a foundation for more advanced functionalities tailored to specific farm management needs.