Java module 3

Exercises Day 4

1 - Interface	Uncle Tobia's Old Farm
Instructions	At Uncle Tobia's old farm, we have 3 animals that produce milk: cows, sheep, and goats. In particular, sheep also provide wool, just like alpacas. Create parent classes and child classes (with their respective attributes and methods) for all the animals at Uncle Tobia's farm. Create at least: ● An abstract Animal class ● An interface called MilkProducer and an interface called WoolProducer. These will be implemented by the appropriate animals. ● One class for each different animal that exists in the farm. Each animal should have a method to make a sound. Create a MainProgram class with the following code to test your implementation:

```java
public class MainProgram {
   public static void main(String[] args) {
       //Create some animals
       Animal[] animals = new Animal[5];
       animals[0] = new Alpaca();
       animals[1] = new Cow();
       animals[2] = new Goat();
       animals[3] = new Goat();
       animals[4] = new Sheep();
       //All animals make a sound
       for(int i = 0; i < 5; i++) {
           System.out.print(animals[i].name + " is making
the sound: ");
           animals[i].makeSound();
       }
       //All animals should produce
       for(int i = 0; i < 5; i++) {
           if(animals[i] instanceof MilkProducer) {
               ((MilkProducer) animals[i]).produceMilk();
           }
           if(animals[i] instanceof WoolProducer) {
               ((WoolProducer) animals[i]).produceWool();
           }
       }
   }
}
```

Expected output	Alpaca is making the sound: Hum Cow is making the sound: Moo Goat is making the sound: Bleat Goat is making the sound: Bleat Sheep is making the sound: Baa Producing alpaca wool Producing milk Producing goat milk

```
Producing goat milk
Producing sheep milk
Producing wool
```

Animal.java

```java
public abstract class Animal {

    public String name;

    public Animal(String name) {
        this.name = name;
    }

    public abstract void makeSound();
}
```

MilkProducer.java

```java
public interface MilkProducer {

    void produceMilk();
}
```

WoolProducer.java

```java
public interface WoolProducer {

    void produceWool();
}
```

Cow.java

```java
public class Cow extends Animal implements MilkProducer {

    public Cow() {
        super("Cow");
    }

    public void makeSound() {
        System.out.println("Moo");
    }

    public void produceMilk() {
        System.out.println("Producing cow milk...");
    }
}
```

Alpaca.java

```java
public class Alpaca extends Animal implements WoolProducer{

    public Alpaca() {
        super("Alpaca");
    }

    public void makeSound() {
        System.out.println("Hum");
    }

    public void produceWool() {
        System.out.println("Producing Alpaca wool...");
    }
}
```

Goat.java

```java
public class Goat extends Animal implements MilkProducer{

    public Goat() {
        super("Goat");
    }

    public void makeSound() {
        System.out.println("Bleat");
    }

    public void produceMilk() {
        System.out.println("Producing goat milk...");
    }
}
```

Sheep.java

```java
public class Sheep extends Animal implements WoolProducer,
MilkProducer{

    public Sheep() {
        super("Sheep");
    }

    public void makeSound() {
        System.out.println("Bee");
    }

    public void produceWool() {
        System.out.println("Producing sheep wool...");
    }
    public void produceMilk() {
        System.out.println("Producing sheep milk...");
    }
}
```

2 - Abstract	Animal Caretaking Simulation
Instructions	As a user, you have the option to choose an animal to take to school: an owl, a cat, or a toad (create a class for each). Once the preferred animal is chosen, an instance of that animal is created. Now it's time to take care of it! Give it a name, feed it, play with it, and clean up after it. Don’t forget to create an Abstract class, it can be called Animal. Create another class to host the main method. It should allow the user to choose which animal they will take to school as many times as they want.

Expected output

```
Choose an animal to take to school: 1.Owl 2.Cat 3.Toad or 0 to exit.
1
Hedwig is eating mice.
Hedwig is playing in the night.
Cleaning up Hedwig's feathers.
Choose an animal to take to school: 1.Owl 2.Cat 3.Toad or 0 to exit.
2
Whiskers is eating fish.
Whiskers is chasing a laser pointer.
Cleaning up Whiskers's litter box.
Choose an animal to take to school: 1.Owl 2.Cat 3.Toad or 0 to exit.
3
Trevor is eating bugs.
Trevor is jumping around.
Cleaning up Trevor's aquarium.
Choose an animal to take to school: 1.Owl 2.Cat 3.Toad or 0 to exit.
4
Invalid choice
Choose an animal to take to school: 1.Owl 2.Cat 3.Toad or 0 to exit.
0
Good bye!
```

Animal.java

```java
abstract class Animal {
    protected String name;

    public Animal(String name) {
        this.name = name;
    }
    public abstract void feed();

    public abstract void play();

    public abstract void cleanUp();
}
```

Cat.java

```java
class Cat extends Animal {
    public Cat(String name) {
        super(name);
    }

    public void feed() {
        System.out.println(name + " is eating fish.");
    }

    public void play() {
        System.out.println(name + " is chasing a laser
pointer.");
    }

    public void cleanUp() {
        System.out.println("Cleaning up " + name + "'s
litter box.");
    }
}
```

Owl.java

```java
class Owl extends Animal {
    public Owl(String name) {
        super(name);
    }

    public void feed() {
        System.out.println(name + " is eating mice.");
    }

    public void play() {
        System.out.println(name + " is playing in the
night.");
    }

    public void cleanUp() {
        System.out.println("Cleaning up " + name + "'s
feathers.");
    }
}
```

Toad.java

```java
class Toad extends Animal {
    public Toad(String name) {
        super(name);
    }

    public void feed() {
        System.out.println(name + " is eating bugs.");
    }

    public void play() {
        System.out.println(name + " is jumping around.");
    }

    public void cleanUp() {
        System.out.println("Cleaning up " + name + "'s
aquarium.");
    }
}
```

SchoolPets.java

```java
import java.util.Scanner;

public class SchoolPets {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int choice = 1;
        while(choice != 0) {
            System.out.println("Choose an animal to take to
school: 1.Owl 2.Cat 3.Toad or 0 to exit.");
            choice = scanner.nextInt();
            Animal pet = null;

            switch (choice) {
                case 0:
                    System.out.println("Good bye!");
                    break;
                case 1:
                    pet = new Owl("Hedwig");
                    break;
                case 2:
                    pet = new Cat("Whiskers");
                    break;
                case 3:
                    pet = new Toad("Trevor");
                    break;
                default:
                    System.out.println("Invalid choice");
                    break;
            }
            if (pet != null) {
                pet.feed();
                pet.play();
                pet.cleanUp();
            }
        }
    }
}
```