public interface Mammal {

    public void speak();

    public boolean eat(Mammal other);

    public void sleep();

    public boolean mate(Mammal other);

}

public class Feline implements Mammal {

    @Override

    public void speak() {

        System.out.println("\*Meow..");

    }

    @Override

    public boolean eat(Mammal other) {

        if (!(other instanceof Feline)) {

            System.out.println("\*tears into meat");

            return true;

        }

        return false;

    }

    @Override

    public void sleep() {

        System.out.println("\*Purrrr..");

    }

    @Override

    public boolean mate(Mammal other) {

        if (other instanceof Feline) {

            System.out.println("\*Meow~");

            return true;

        }

        return false;

    }

}

public class Canine implements Mammal {

    @Override

    public void speak() {

        System.out.println("\*Woof..");

    }

    @Override

    public boolean eat(Mammal other) {

        if (!(other instanceof Canine)){

            System.out.println("\*chomps meat");

            return true;

        }

        return false;

    }

    @Override

    public void sleep() {

        System.out.println("Ssssuu..");

    }

    @Override

    public boolean mate(Mammal other) {

        if (other instanceof Canine){

            System.out.println("\*Woof~");

            return true;

        }

        return false;

    }

}

public class Bovine implements Mammal {

    @Override

    public void speak() {

        System.out.println("\*Moo");

    }

    @Override

    public boolean eat(Mammal other) {

        if (!(other instanceof Bovine)) {

            System.out.println("\*Chews flesh");

            return true;

        }

        return false;

    }

    @Override

    public void sleep() {

        System.out.println("\*Zzzzz..");

    }

    @Override

    public boolean mate(Mammal other) {

        if (other instanceof Bovine) {

            System.out.println("\*Moo~");

            return true;

        }

        return false;

    }

}

public class Tiger extends Feline {

    @Override

    public void speak() {

        System.out.println("\*Gurrr..");

    }

    @Override

    public boolean eat(Mammal other) {

        if (!(other instanceof Tiger)){

            System.out.println("\*tears into meat");

            return true;

        }

        return false;

    }

    @Override

    public boolean mate(Mammal other) {

        if (other instanceof Tiger){

            System.out.println("\*Gurrr~");

            return true;

        }

        return false;

    }

}

public class Lion extends Feline {

    @Override

    public void speak() {

        System.out.println("\*ROOOAR");

    }

    @Override

    public boolean eat(Mammal other) {

        if (!(other instanceof Lion)){

            System.out.println("\*tears into meat");

            return true;

        }

        return false;

    }

    @Override

    public void sleep() {

        System.out.println("\*PURRRR...");

    }

    @Override

    public boolean mate(Mammal other) {

        if (other instanceof Lion){

            System.out.println("\*ROAR~");

            return true;

        }

        return false;

    }

}

public class Bull extends Bovine {

    @Override

    public void speak() {

        System.out.println("\*Mouu..");

    }

    @Override

    public boolean eat(Mammal other) {

        if (!(other instanceof Bovine)) {

            System.out.println("\*chews flesh.");

            return true;

        }

        return false;

    }

    @Override

    public boolean mate(Mammal other) {

        if (other instanceof Bovine) {

            System.out.println("\*Moo~");

            return true;

        }

        return false;

    }

}

public class Wolf extends Canine {

    @Override

    public void speak() {

        System.out.println("\*Awoooooo..");

    }

    @Override

    public boolean eat(Mammal other) {

        if (!(other instanceof Wolf)) {

            System.out.println("\*chomps meat");

            return true;

        }

        return false;

    }

    @Override

    public boolean mate(Mammal other) {

        if (other instanceof Wolf) {

            System.out.println("\*Awoooooo~");

            return true;

        }

        return false;

    }

}

import java.util.HashMap;

import java.util.Scanner;

public class Main {

    private static boolean SLEEP = true;

    private static boolean SPEAK = false;

    private static boolean EAT = false;

    private static boolean MATE = true;

    public static void main(String[] args) {

        HashMap<String, Mammal> bad\_farm = new HashMap<String, Mammal>();

        boolean done = false;

        Scanner s = new Scanner(System.in);

        while (!done) {

            System.out.println("What do you want to do? (add, speak, eat, sleep, mate, end)");

            String next = s.nextLine();

            switch (next.strip().toLowerCase()) {

                case "add":

                    add(bad\_farm, s);

                    break;

                case "speak":

                    unary(bad\_farm, s, SPEAK);

                    break;

                case "eat":

                    binary(bad\_farm, s, EAT);

                    break;

                case "sleep":

                    unary(bad\_farm, s, SLEEP);

                    break;

                case "mate":

                    binary(bad\_farm, s, MATE);

                    break;

                case "end":

                    System.out.println("Ending");

                    done = true;

                    break;

                default: {

                }

            }

        }

        s.close();

    }

    private static void unary(HashMap<String, Mammal> bad\_farm, Scanner s, boolean flag) {

        System.out.println("What is this animal's name?");

        String name = s.nextLine();

        Mammal animal = bad\_farm.get(name);

        if (animal != null) {

            if (flag == SLEEP)

                animal.sleep();

            else

                animal.speak();

        } else

            System.out.println(name + " does not exist.");

    }

    private static void binary(HashMap<String, Mammal> bad\_farm, Scanner s, boolean flag) {

        System.out.println("What is the animal's name?");

        String animal1\_s = s.nextLine();

        System.out.println("What is the " + (flag == MATE ? "mate" : "prey") + "'s name?");

        String animal2\_s = s.nextLine();

        Mammal animal1\_m = bad\_farm.get(animal1\_s);

        Mammal animal2\_m = bad\_farm.get(animal2\_s);

        if (animal1\_m != null && animal2\_m != null) {

            if ((flag == MATE ? animal1\_m.mate(animal2\_m) : animal1\_m.eat(animal2\_m)) && (flag == EAT))

                bad\_farm.remove(animal2\_s);

        } else {

            if (animal1\_m == null)

                System.out.println(animal1\_s + " does not exist.");

            if (animal2\_m == null)

                System.out.println(animal2\_s + " does not exist.");

        }

    }

    private static void add(HashMap<String, Mammal> bad\_farm, Scanner s) {

        System.out.println("What species is the animal? (Bull, Lion, Tiger, Wolf)");

        String species = s.nextLine();

        System.out.println("'" + species + "'");

        System.out.println("What is this animal's name?");

        String name = s.nextLine();

        switch (species.strip().toLowerCase()) {

            case "bull":

                bad\_farm.put(name, new Bull());

                break;

            case "lion":

                bad\_farm.put(name, new Lion());

                break;

            case "tiger":

                bad\_farm.put(name, new Tiger());

                break;

            case "wolf":

                bad\_farm.put(name, new Wolf());

                break;

            default:

                System.out.println("Invalid choice.");

                break;

        }

    }

}