```
public interface Mammal {
    public void speak();

    public boolean eat(Mammal other);

    public void sleep();

    public boolean mate(Mammal other);
}
```

```
@Override
public void speak() {
   System.out.println("*Meow..");
@Override
public boolean eat(Mammal other) {
    if (!(other instanceof Feline)) {
       System.out.println("*tears into meat");
@Override
public void sleep() {
   System.out.println("*Purrrr..");
@Override
public boolean mate(Mammal other) {
    if (other instanceof Feline) {
       System.out.println("*Meow~");
       return true;
```

```
@Override
public void speak() {
   System.out.println("*Woof..");
@Override
public boolean eat(Mammal other) {
    if (!(other instanceof Canine)){
       System.out.println("*chomps meat");
@Override
public void sleep() {
   System.out.println("Ssssuu..");
@Override
public boolean mate(Mammal other) {
    if (other instanceof Canine) {
       System.out.println("*Woof~");
       return true;
```

```
@Override
public void speak() {
   System.out.println("*Moo");
@Override
public boolean eat(Mammal other) {
    if (!(other instanceof Bovine)) {
       System.out.println("*Chews flesh");
@Override
public void sleep() {
   System.out.println("*\Zzzzz..");
@Override
public boolean mate(Mammal other) {
    if (other instanceof Bovine) {
       System.out.println("*Moo~");
       return true;
```

```
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;
    }
}
```

```
@Override
public void speak() {
   System.out.println("*ROOOAR");
@Override
public boolean eat(Mammal other) {
    if (!(other instanceof Lion)){
       System.out.println("*tears into meat");
@Override
public void sleep() {
   System.out.println("*PURRRR...");
@Override
public boolean mate(Mammal other) {
    if (other instanceof Lion) {
       System.out.println("*ROAR~");
       return true;
```

```
@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;
                    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());
```