Interface

```
public interface AnimalBehavior {
    void eat();
    void move();
}

public class Dog implements AnimalBehavior {
    public void eat() {
        System.out.println("The dog is eating dog food.");
    }

    public void move() {
        System.out.println("The dog is running in the park.");
    }
}

public class Fish implements AnimalBehavior {
    public void eat() {
        System.out.println("The fish is eating flakes.");
    }

    public void move() {
        System.out.println("The fish is swimming in the tank.");
    }
}
```

#1

```
public class Main {
```

```
    public static void main(String[] args) {
        AnimalBehavior myDog = new Dog();
        AnimalBehavior myFish = new Fish();

        System.out.println("Dog's Actions:");
        myDog.eat();
        myDog.move();

        System.out.println("\nFish's Actions:");
        myFish.eat();
        myFish.move();
    }
}
```

1#Output

```
|Dog's Actions:
|The dog is eating dog food.
|The dog is running in the park.
|Fish's Actions:
|The fish is eating flakes.
|The fish is swimming in the tank.
```

#2

```
public class Main {
    public static void main(String[] args) {
        // Create an array of AnimalBehavior
        AnimalBehavior[] animals = new AnimalBehavior[2];
        animals[0] = new Dog();
        animals[1] = new Fish();

        // Iterate over the array and invoke methods
        for (AnimalBehavior animal : animals) {
            animal.eat();
            animal.move();
        }
    }
}
```

2#Output

```
|The dog is eating dog food.
|The dog is running in the park.
|The fish is eating flakes.
|The fish is swimming in the tank.
```