





```
public abstract class Animal {
    protected String color;
    protected int numOfLegs;
    public Animal(String color, int numOfLegs) {
        this.color = color;
        this.numOfLegs = numOfLegs;
    }
    public abstract void makeNoise();
    public String toString() {
        return getClass().getName() + ": color=" +
            color + ", numOfLegs=" + numOfLegs + ", ";
    }
    public final String getColor() {
        return color;
    }
    public final int getNumOfLegs() {
        return numOfLegs;
    }
}
```



```
public class Horse extends Animal {
    public static final int NUM_OF_LEGS = 4;
    private int tailLen;
    public Horse(String color, int tailLen) {
        super(color, NUM_OF_LEGS);
        this.tailLen = tailLen;
    }
    @Override
    public void makeNoise() {
        System.out.println("Hi Yahah!");
    }
    public void ride() {
        System.out.println("I'm riding!");
    }
    @Override
    public String toString() {
        return super.toString() + ", tailLen=" +
tailLen;
    }
}
```



```
public abstract class Cat extends Animal {
    public static final int NUM_OF_LEGS = 4;
    protected int whiskersLen;
    public Cat(String color, int whiskersLen) {
        super (color, NUM_OF_LEGS);
        this.whiskersLen = whiskersLen;
    }
    public void scratch () {
        System.out.println(getClass().getName() + " is scrathin
    }
    @Override
    public void makeNoise () {
        System.out.println("Miyaoooooo!");
    }
    @Override
    public String toString() {
        return super.toString() + ", whiskersLen=" +
        whiskersLen;
    }
}
```



```
public class SiamiCat extends Cat {  
    private String favoriteFood;  
    public SiamiCat(String color, int whiskersLen, String  
        favoriteFood) {  
        super(color, whiskersLen);  
        this.favoriteFood = favoriteFood;  
    }  
    @Override  
    public void makeNoise() {  
        super.makeNoise();  
        System.out.println("I'm so spoiled!");  
    }  
    @Override  
    public String toString() {  
        return super.toString() + ", favoriteFood: " +  
            favoriteFood;  
    }  
}
```



```
public class StreetCat extends Cat {  
    private int numOfFights;  
    public StreetCat(String color, int whiskersLen, int  
numOfFights) {  
        super(color, whiskersLen);  
        this.numOfFights = numOfFights;  
    }  
    @Override  
    public void makeNoise() {  
        super.makeNoise();  
        System.out.println("I want to see a dog!");  
    }  
    public void fight() {  
        System.out.println("I want to have a good fight!");  
    }  
    @Override  
    public String toString() {  
        return super.toString() + ",  numOfFights=" +  
numOfFights;  
    }  
}
```



```
public class Fish extends Animal {
    public static final int NUM_OF_LEGS = 0;
    private int swimSpeed;
    public Fish(String color, int swimSpeed) {
        super(color, NUM_OF_LEGS);
        this.swimSpeed = swimSpeed;
    }
    @Override
    public void makeNoise() {
        System.out.println("Blu-Blu");
    }
    @Override
    public String toString() {
        return super.toString() + ", swimSpeed: " + swimSpeed;
    }
}
```



```
public static void main(String[] args) {  
    Animal[] animals = new Animal[4];  
    animals[0] = new Horse("brown", 120);  
    animals[1] = new SiamiCat("gray", 12, "RoyalCAT");  
    animals[2] = new StreetCat("gingi", 15, 34);  
    animals[3] = new Fish("gold", 2);  
    for (int i=0 ; i < animals.length ; i++) {  
        System.out.println(animals[i].toString());  
        animals[i].makeNoise();  
    }  
}
```

Horse: color=brown, numOfLegs=4, , tailLen=120

Hi Yahah!

SiamiCat: color=gray, numOfLegs=4, , whiskersLen=12,  
favoriteFood: RoyalCAT

Miyaoooooo!

I'm so spoiled!

StreetCat: color=gingi, numOfLegs=4, , whiskersLen=15,  
numOfFights=34

Miyaoooooo!

I want to see a dog!

Fish: color=gold, numOfLegs=0, , swimSpeed: 2

Blu-Blu





```
public static void main(String[] args) {  
    Animal[] animals = new Animal[4];  
    animals[0] = new Horse("brown", 120);  
    animals[1] = new SiamiCat("gray", 12, "RoyalCAT");  
    animals[2] = new StreetCat("gingi", 15, 34);  
    animals[3] = new Fish("gold", 2);  
    for (int i=0 ; i < animals.length ; i++) {  
        if (animals[i] instanceof Cat) {  
            Cat temp = (Cat)animals[i];  
            temp.scratch();  
        }  
    }  
}
```



