```java
package com.publicept;

/**
 * Dog extends Animal
 * "extends" keyword: inherit from a super class (=parent-class)
 *
 * Dog(String name, int size, int weight, int eyes, int legs, int tail, int teeth, String coat)
 *
 * chew()
 * eat()
 * walk()
 * run()
 * moveLegs(int speed)
 * move(int speed)
 */
public class Dog extends Animal {


    // == fields ==
    private int eyes;
    private int legs;
    private int tail;
    private int teeth;
    private String coat;



    // == constructors ==
    /**
     * Dog.Dog()
     * @param name
     * @param size
     * @param weight
     * @param eyes
     * @param legs
     * @param tail
     * @param teeth
     * @param coat
     */
    public Dog(String name, int size, int weight, int eyes, int legs, int tail, int teeth, String coat) {
        /* super() must be called first in the extended constructor! */
        super(name, 1, 1, size, weight);
        this.eyes = eyes;
        this.legs = legs;
        this.tail = tail;
        this.teeth = teeth;
        this.coat = coat;
    }



    // == override methods ==
    /**
     * Dog.eat()
     * -> Override
     */
    @Override
    public void eat() {
        System.out.println("Dog.eat() called");
        chew();
        super.eat();
    }


    /**
     * Dog.move()
     * @param speed
     */
    @Override
    public void move(int speed) {
        System.out.println("Dog.move() called");
        moveLegs(speed);
        super.move(speed);
    }



    // == public methods ==
    /**
     * Dog.chew()
     */
    public void chew() {
        System.out.println("Dog.chew() called");
    }


    /**
     * Dog.walk()
     * -> move();
     */
    public void walk() {
        System.out.println("Dog.walk() called -> move(5)");
        /**
         * Generally, it's recommended to use move() instead of super.move():
         * If the move()-function in the Dog-class is overridden at a later time,
         * it's will correctly call that move() method.
         * -> see walk() and run() examples below...
```

```
 98          */
 99         move(5);
100     }
101
102
103     /**
104      * Dog.run()
105      * -> super.move();
106      */
107     public void run() {
108         System.out.println("Dog.run() called -> super.move(10) -> move()-override is skipped!");
109         super.move(10);
110     }
111
112
113     /**
114      * moveLegs()
115      * @param speed
116      */
117     private void moveLegs(int speed) {
118         System.out.println("Dog.moveLegs() called");
119     }
120 }
121
```