```
// This program is copyright VUW.
// You are granted permission to use it to construct your answer to a COMP102 assignment.
// You may not distribute it in any other way without permission.

/* Code for COMP102 - 2024T3, Assignment 2
 * Name:
 * Username:
 * ID:
 */

import ecs100.*;

/**
 * An Animal object is an animal character, displayed on the screen
 * that can
 *   go left,
 *   go right,
 *   jump,
 *   speak, "think" or shout a phrase,
 *   introduce itself.
 */

public class Animal {
    public static final double STEP = 30;

    /* Fields representing the state of a Animal */
    private String animal;
    private String name;
    private double imageX = -100;   // top left corner of image
    private double imageY = -100;
    private String direction = "left";

    /* Fields containing dimensions of Animals */
    private int imageHeight = 125;
    private int imageWidth = 140;

    private int wordsWidth = 160;
    private int wordsHeight = 45;
    private int wordSize = 12;

    /**
     * Constructor requires
     *  - the type of animal,
     *  - the name of the animal,
     *  - and the coordinates (left, top) of where it should be placed.
     *    For example
     *    new Animal("dog", "Scruff", 100, 50);
     */
    public Animal(String typeOfAnimal, String nameOfAnimal, double x, double y ){
        this.animal=typeOfAnimal;
        this.name=nameOfAnimal;
        this.imageX = x;
        this.imageY = y;
        UI.setFontSize(wordSize);
        this.draw();
    }

    /**
     * Make the Animal face to the left and
     * then move the specified distance to the left
     */
    public void goLeft(double dist) {
        this.erase();
        this.direction="left";
        this.draw();
        UI.sleep(50);
        while (dist > STEP){
            this.erase();
            this.imageX = this.imageX - STEP;
```

```
            dist = dist - STEP;
            this.draw();
            UI.sleep(50);
        }
        if (dist > 0){
            this.erase();
            this.imageX = this.imageX - dist;
            this.draw();
            UI.sleep(50);
        }
    }

    /**
     * Make the Animal face to the right and
     * then move the specified distance to the right
     */
    public void goRight(double dist) {
        this.erase();
        this.direction="right";
        this.draw();
        UI.sleep(50);
        while (dist > STEP){
            this.erase();
            this.imageX = this.imageX + STEP;
            dist = dist - STEP;
            this.draw();
            UI.sleep(50);
        }
        if (dist > 0){
            this.erase();
            this.imageX = this.imageX + dist;
            this.draw();
            UI.sleep(50);
        }
    }

    /**
     * move the Animal up then down
     */
    public void jump(double dist) {
        this.erase();
        this.imageY = this.imageY - dist*0.7;
        this.draw();
        UI.sleep(100);
        this.erase();
        this.imageY = this.imageY - dist*0.3;
        this.draw();
        UI.sleep(100);
        this.erase();
        this.imageY = this.imageY + dist*0.2;
        this.draw();
        UI.sleep(100);
        this.erase();
        this.imageY = this.imageY + dist*0.3;
        this.draw();
        UI.sleep(500);
    }

    /**
     * makes the Animal say something in a speech box
     */
    public void speak(String words) {
        double boxX = this.imageX;
        double boxY = this.imageY - this.wordsHeight - 20;

        if (this.direction.equals("right"))
            boxX += 15 ;
        else
            boxX +=  this.imageWidth  - 15 - this.wordsWidth;
```

```
        UI.eraseRect(boxX, boxY, this.wordsWidth, this.wordsHeight);
        UI.drawRect(boxX, boxY, this.wordsWidth, this.wordsHeight);
        UI.drawString(words, boxX + 5, boxY + this.wordsHeight/2 + 3);

        UI.sleep(3000);

        UI.eraseRect(boxX, boxY, this.wordsWidth+1, this.wordsHeight+1);
    }

    /**
     * makes the Animal introduce itself with a greeting word
     */
    public void introduce(String greeting) {
        this.speak(greeting + " my name is " + name);
        this.speak("I am a " + animal);
    }

    /**
     * makes the Animal shout in big block letter
     */
    public void shout(String words) {
        UI.setFontSize(20);
        this.speak(words.toUpperCase());
        UI.setFontSize(wordSize);
    }

    /**
     * makes the Animal think something in a speech bubble
     */
    public void think(String words) {
        double bubbleX = this.imageX;
        double bubbleY = this.imageY - this.wordsHeight - 2;
        double circleX = this.imageX;
        double circleY = this.imageY;

        if (this.direction.equals("right")){
            bubbleX += 15 ;
            circleX += this.imageWidth + 20;
        }
        else{
            bubbleX +=  this.imageWidth  - 15 - this.wordsWidth;
            circleX -= 30;
        }

        UI.eraseOval(bubbleX, bubbleY, this.wordsWidth, this.wordsHeight);
        UI.drawOval(bubbleX, bubbleY, this.wordsWidth, this.wordsHeight);
        UI.drawString(words, bubbleX + 12, bubbleY + this.wordsHeight/2 + 3);

        UI.drawOval(circleX, circleY, 10, 10);

        UI.sleep(1000);

        UI.eraseRect(bubbleX, bubbleY, this.wordsWidth+1, this.wordsHeight+1);
        UI.eraseOval(circleX, circleY, 10, 10);
    }

    /**
     * Helper method that erases the Animal
     * All the public methods that change the image call erase first
     */
    private void erase() {
        UI.eraseRect(this.imageX, this.imageY, this.imageWidth+1, this.imageHeight+1);
    }

    /**
     * Helper method that draws the Animal
     * All the public methods that change the image call draw.
     */
```

```
    private void draw(){
        String fname = null;
        fname = "animals/" + this.animal +"-"+this.direction+".gif";
        UI.drawImage(fname, this.imageX, this.imageY, this.imageWidth, this.imageHeight);
    }

}
```