```
// 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 4
 * Name:
 * Username:
 * ID:
 */

import ecs100.*;
import java.awt.Color;

/**
 * An Emoji represents an object that can be thrown through the air, or knocked down
 *  onto the ground by another Emoji.
 *  An Emoji can be created at a position, and it will appear at that place.
 *  An Emoji can be launched with a velocity (horizontal and vertical speeds)
 *  The simulation can call the step() method to make the Emoji move one step in its current
direction
 *  If a moving Emoji bumps into a stationary Emoji, it will launch the other Emoji towards the
ground.
 * There are a set of different emoji images, all of size 40x40
 */

public class Emoji{

    // Constants for all Emoji: size, position of the ground
    public static final double SIZE = 40;  // width/length of the emoji images
    public static final double GROUND = EmojiLauncher.GROUND;
    public static final double GRAVITY = 0.25;  // how much to reduce the speed each step.

    // Fields to store state of the Emoji:
    /*# YOUR CODE HERE */
    private double x ;
    private double h;
    private String name;
    private double xSpeed;
    private double ySpeed;




    // Constructor
    /**
     * Construct a new Emoji object.
     *    Parameters are
     *   the initial position (x of the center, and
     *      the height of the bottom of the emoji above the ground),
     *   the name
     *    Stores the parameters into fields
     *  SHOULD NOT DRAW THE EMOJI!
     */
    public Emoji(double x, double h, String name){
        /*# YOUR CODE HERE */
        this.x = x;
        this.h = h;
        this.name = name;

    }

    // Methods
    /**
     * Draw the Emoji on the Graphics Pane centered at its current position
     */
    public void draw(){
        /*# YOUR CODE HERE */
        UI.drawImage("emojis/" + this.name , x - SIZE/2,GROUND - h - SIZE,SIZE,SIZE);
    }

    /**
     * If the emoji's speed is not 0, move the Emoji one step (DO NOT REDRAW IT)
     *    Change its height and x position using the vertical and horizonal steps
     *    Reduce its vertical speed each step (due to gravity),
     *    If it would hit the ground, then change its y position so that it is
     *     resting on the ground and
     *     set its speed (horizontal and vertical) to 0.
     */
    public void step(){
        /*# YOUR CODE HERE */
        if(xSpeed != 0 || ySpeed != 0 ){
            x=x+xSpeed;
            h=h+ySpeed;
            ySpeed = ySpeed - GRAVITY;
        }

        if(h<=0){
            h= 0;
            xSpeed = 0;
            ySpeed = 0;
        }

    }

    /**
     * Return the height of the bottom of the emoji above the ground
     */
    public double getHeight(){
        /*# YOUR CODE HERE */
        return this.h;

    }

    /**
     * Return the horizontal position of the emoji
     */
    public double getX(){
        /*# YOUR CODE HERE */
        return this.x;

    }

    /**
     * Return the speed of the emoji
     */
    public double getSpeed(){
        /*# YOUR CODE HERE */
        return Math.hypot(xSpeed,ySpeed);


    }

    /**
     * Launch the emoji at the specified horizontal and vertical speeds
     */
    public void launch(double xSpeed, double ySpeed){
        /*# YOUR CODE HERE */
        this.xSpeed = xSpeed;
        this.ySpeed = ySpeed;

    }

    /**
     * Return true if this emoji is touching the other emoji (and isn't the same emoji)
     */
    public boolean touching(Emoji other){
        /*# YOUR CODE HERE */
        if(this == other){
            return false;
        }
        if(Math.hypot(this.x - other.x , this.h - other.h) <= SIZE){
            return true;

        }



        return false;


    }

    /**
     * Make this emoji bump the other emoji and make it start moving.
     * Assumes: this != other, this is moving, other is not moving
     * Simple version: just make the other start to move a little bit.
     * Better version: do a proper, elastic collision.
     */
    public void bump(Emoji other){
        /*# YOUR CODE HERE */

        other.xSpeed = 1;
        other.ySpeed = 1;



    }


}
```