Exercise 9 (10 points)

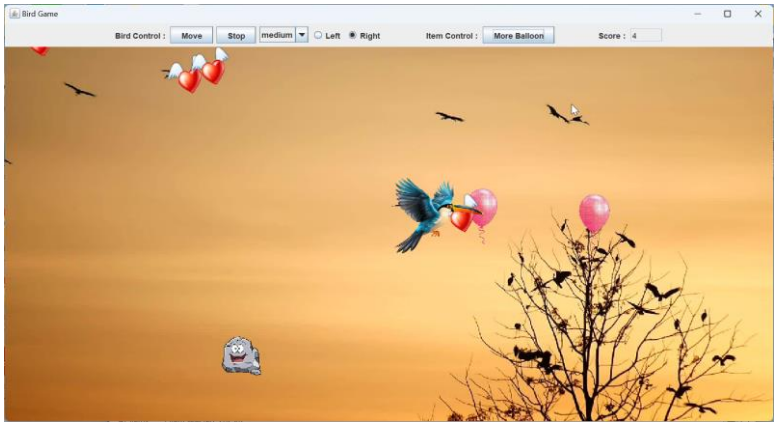
- The first lines of all source files must be comment containing your name & ID
- Put all files (source, input, output) in folder Ex9_xxx where xxx = your full ID. That is, your source files must be in package Ex9_xxx and input/output files (if there is any) must be read from/write to this folder
- Zip Ex9_xxx & submits it to Google Classroom. Email submission is not accepted

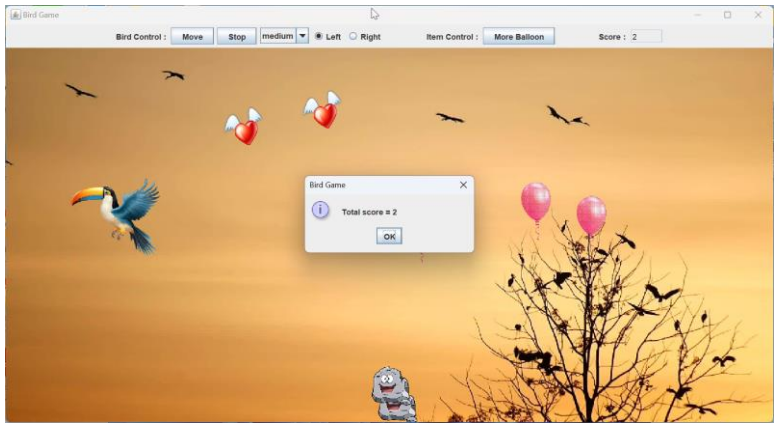
==========

Use the given image/sound files and source file (MainApplication.java). Unzip resources.zip and put this folder in your project folder (Ex9_xxx)

Complete the source file to make program work as follows:

Bird and items (balloons, hearts, rocks) are controlled by separate threads





1. Bird control
   1.1 Move & Stop buttons to move & stop
   1.2 Combo box to set Bird's speed
   1.3 Radio buttons to turn left & move to the left, or turn right & move to the right. When reaching one side of the frame, it'll appear on the other side

2. Item control
   2.1 Balloon button to place 1 Balloon at random location on Bird's path
   2.2 When Bird hits a Balloon, that Balloon will change to either Heart or Rock and the score will increase (for Heart) or decrease (for Rock)
   2.3 Heart will also float up, while Rock will fall down

3. Report total score when closing frame

4. All listener classes must be anonymous classes. Add listeners as follows

4.1 Add ActionListener to Move & Stop buttons, to make Bird move or stop

- Move → create and start birdThread
- Stop → stop birdThread

4.2 Add ItemListener to combo box, to set Bird's speed

- Fast = short sleeping time for birdThread
- Slow = long sleeping time for birdThread

4.3 Add ItemListener to each radio button, to set Bird's direction

4.4 Add ActionListener to Balloon button, to add a Balloon at random location. It can be done by creating & starting a new itemThread (each item is controlled by each thread)

4.5 Add WindowListener to the frame, to show the final score when closing it

5. Use birdThread & itemThread to make all labels move automatically. Anonymous class can also be applied. Complete method setItemThread and class ItemLabel, using example from setBirdThread and BirdLabel

6. Complete method updateScore to increase/decrease score when Bird hits a Balloon. This method requires proper synchronization because it can be called by multiple itemThreads at the same time (if >1 Balloons overlap with Bird)

- Making individual itemThreads check collision with Bird is more efficient than making only 1 birdThread check collisions with all Balloons

All given code can be modified as needed