Exercise 8 (10 points) – can be done individually or in pair

- The first lines of all source files must be comment containing names & IDs of all members. Also create file readme.txt containing names & IDs of all members.

- Put all files (source, input, output) in folder Ex8_xxx where xxx = your full ID. That is, your source files must be in package Ex8_xxx and input/output files (if there is any) must be read from/write to this folder. From now on, you'll get point deduction for wrong package & folder structure.

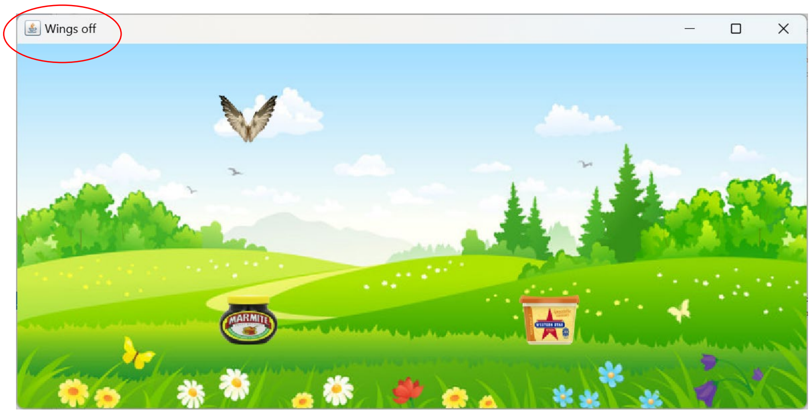
- The group representative zips Ex8_xxx & submits it to Google Classroom. The other members submit only readme.txt. Email submission is not accepted.

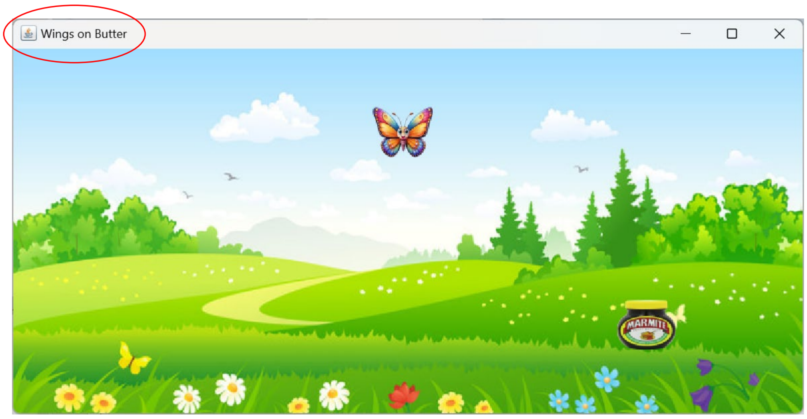
- The exercise is graded only once, and after graded, members can't be added.

==

Use the given image files and source file (MainApplication.java). Unzip resources.zip and put this folder in your project folder (Ex8_xxx).

Complete the source file to make the program work as follows:





There are 2 types of labels.

- CharacterLabels (charLabels[i] keeping Marmite and Butter) can move by keys WSAD or arrow keys.

  But due to event handling, only 1 character will be moved at a time even when pressing both sets of keys simultaneously.

- ItemLabel (i.e. Wing) can be move by mouse drag.

1. Characters without wings
   - Marmite can move left/right by keys A/D, but can't move up/down.
   - Butter can move left/right by arrow keys LEFT/RIGHT, but can't move up/down.
   - When either character reaches one side of the frame, it'll appear on the opposite side.
   - No response to mouse cursor and key ESC.

2. Character with wings (flyingLabel) – only 1 character at a time can have wings
   - Marmite with wings (Crow) can move left/right as before + up/down by keys W/S.
   - Butter with wings (Butterfly) can move left/right as before + up/down by arrow keys UP/DOWN.
   - But it won't appear on the opposite side when reaching the top/bottom of the frame.
   - When mouse cursor touches it, it will jump to a random location inside the frame.
   - Can take off the wings by using key ESC. It must be back on the ground; the wings can be thrown to any location inside the frame.

3. Wings (itemLabel)
   - Can be dragged within the frame by using mouse.
   - When it is dragged on top of either Marmite or Butter, that character will be changed to Crow or Butterfly. Update the frame's title to inform who's having the wings.

4. Complete class MainApplication extends JFrame implements KeyListener

   JLabel cannot hear KeyEvent. We have to make JFrame hear & handle KeyEvent on its behalf. We need different sets of keys for Marmite and Butter, but JFrame will handle KeyEvent to move only 1 of them at a time.

   4.1 Move Marmite/Crow when pressing alphabet keys WASD ➔ e.g. by making it call moveUp/moveDown/moveLeft/moveRight according to move conditions in 1+2.

   4.2 Move Butter/Butterfly when pressing arrow keys ➔ e.g. by making it call moveUp/moveDown/moveLeft/moveRight according to move conditions in 1+2.

   4.3 Take wings off the flyingLabel when pressing ESC and reset the frame's title ➔ e.g. by changing flyingLabel's & itemLabel's icons and updating their move conditions & locations.

5. Complete class CharacterLabel extends BaseLabel extends MouseListener
   5.1 Update its location according to move conditions.
   5.2 Switch between the character with & without wings.
   5.3 Jump to a random location when mouse cursor enters its bound.
   5.3 Add variables/methods or make further modifications as needed.

6. Complete class ItemLabel extends BaseLabel implements MouseMotionListener
   6.1 Update its location upon mouse drag. If it overlaps with a CharacterLabel, switch its and character's labels to alternative labels. Also update their move conditions.

   Note: to check whether 2 labels overlap

```
if ( this.getBounds().intersects(otherLabel.getBounds()) )
{
   otherLabel.doSomething();
   this.doSomething();
}
```

   6.2 Add variables/methods or make further modifications as needed.

7. Add listener objects to proper component objects.