Exercise 8 (10 points)

- The first lines of all source files must be comment containing your name & ID
- 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
- Zip Ex8_xxx & submits it to Google Classroom. Email submission is not accepted

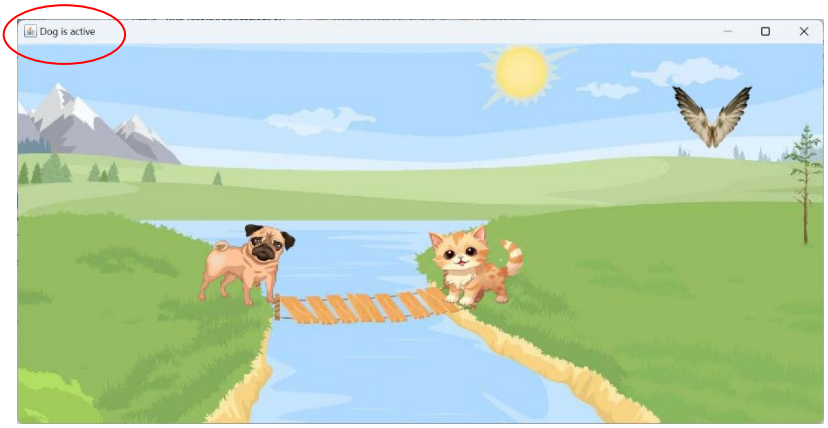
==========

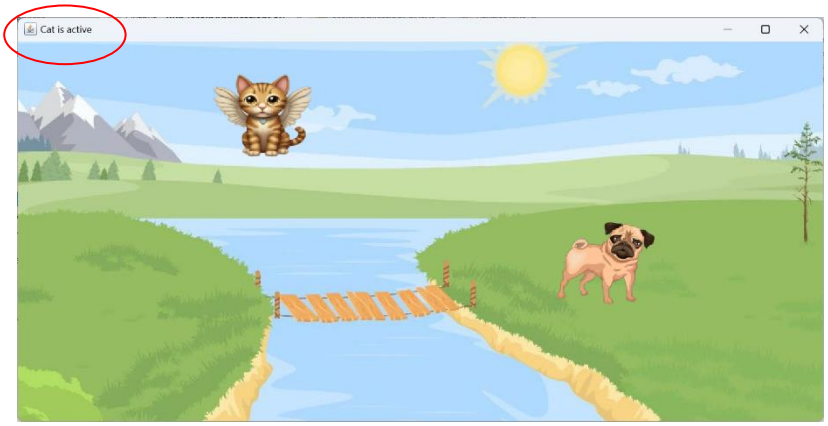
Use the given image files and source file (MainApplication.java). Unzip resources.zip and put this folder in your project folder (Ex8_xxx)

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

There are 2 different types of labels

- CharacterLabels (petLabels[i] keeping Cat and Dog) can be moved by arrow keys
- ItemLabel (wingLabel) can be move by mouse drag





1. Only 1 petLabels[i] can be active (i.e. as activeLabel) and moved by arrow keys at a time.
   - Use alphabet key C to switch the activeLabel to Cat, and key D to switch to Dog. Active character must be shown on title bar.
2. activeLabel without wings
   - Can move left/right by arrow keys LEFT/RIGHT. When it reaches one side of the frame, it'll appear on the opposite side.
   - Can jump to the opposite site of the bridge by alphabet key J. But if it is already on the bridge, it won't jump.
   - No response to arrow keys UP/DOWN and key ESC.
3. activeLabel with wings
   - Can move left/right by arrow keys LEFT/RIGHT. When it reaches one side of the frame, it'll appear on the opposite side.
   - Can also move up/down by arrow keys UP/DOWN, but only within the frame.
   - Can take off the wings by key ESC. After taking off the wings, it must land on the ground, and the wings can be thrown to any location inside the frame.
   - No response to alphabet key J.

4. wingLabel
   - Can be dragged within the frame at any time by using mouse.
   - When it is dragged on top of the activeLabel, the activeLabel will put on the wings.
   - But if it is dragged on top of the non-active label, there won't be any effect.

5. Complete class MainApplication extends JFrame implements KeyListener

   JLabel cannot hear KeyEvent. We have to make JFrame hear & handle KeyEvent on its behalf. And because JFrame can handle one JLabel at a time, we will make it handle activeLabel which is the chosen petLabels[i].

   5.1 Set title & activeLabel when alphabet key C/D is pressed.

   5.2 Move activeLabel when arrow or alphabet key J is pressed ➔ e.g. by making it call moveUp/moveDown/moveLeft/moveRight/jump.

   5.3 Take the wings off the activeLabel when key ESC is pressed ➔ e.g. by changing activeLabel's & wingLabel's icons and updating their move conditions & locations.

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

6. Complete class CharacterLabel extends BaseLabel
   6.1 Add methods to update its location according to move conditions.
   6.2 Add methods to switch between the character with & without wings.
   6.3 Add variables/methods or make further modifications as needed.

7. Complete class ItemLabel extends BaseLabel implements MouseMotionListener
   7.1 Add methods to update its location upon mouse drag. If it overlaps with the activeLabel, add the wings to the activeLabel ➔ e.g. by changing activeLabel's & wingLabel's icons and updating their move conditions.

   Note: to check whether 2 labels overlap

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

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

8. Add listener objects to proper component objects.