ORACLE ACADEMY

Lesson 9: Understanding Abstraction

Try It: Practice Activities

Objectives

- Define abstraction and provide an example of when it is used

Vocabulary

Identify the vocabulary word for each definition below.

	A technique used to command newly-created instances to perform different actions.

Project

1. Continue the project from Section 3 Lesson 8.
2. Open the Project named FrogFlyL8. From the Scenario Menu, choose “Save as” and use the name FrogFlyL9. This will keep a copy from Lesson 8 in case there is a need to refer to it later.
3. Compile the Greenfoot project.
4. In this section, the Frog and Fly subclasses will be modified to allow the Frog to “catch and eat” the fly. That Fly will disappear, then a new Fly will be added to the world.
5. Open the Frog Code Editor.
6. Delete the atWorldEdge() method.
7. Delete the code in the act() method.
8. Replace with the steps to follow.
9. Add moveLeft(), moveRight(), moveUp(), and moveDown() methods to the class and change the code for the act() method to use these.
10. Code for the moveLeft() method would look like this:

```
this.setLocation ( getX() - 5, this.getY());
```

To move the Fly left, the code for the act() method would look like this:

```
if (Greenfoot.isKeyDown("left") ) {
        moveLeft();
}
```

Repeat the if statement for the other three directions.

11. Add the following lines to the act() method for the Frog:

```
Actor getFly = getOneIntersectingObject(Fly.class);
if (getFly != null) {
        Greenfoot.playSound("Your Sound File Name Goes Here");
        getWorld().removeObject(getFly);
        getWorld().addObject(new Fly(), Greenfoot.getRandomNumber(600),
        Greenfoot.getRandomNumber(400));
}
```

12. Open the Fly Code Editor.

13. Delete the atWorldEdge() method.

14. Replace the code in act() with the following:

15. move(Greenfoot.getRandomNumber(5));

    turn(Greenfoot.getRandomNumber(9));

16. Compile and test the new features.

17. Save your project and exit.

Optional Practice

1. Open a previously-created scenario that you are building on, then save a copy for this exercise, with a name that reflects the name of this lesson.
   - In your scenario, create a constructor in the Actor's subclass's source code to store the sound and keyboard key variable names. The code should be written at the top of the source code underneath the class definition. A separate constructor is created to define the two variables and parameters.
2. Open a previously-created scenario that you are building on, then save a copy for this exercise, with a name that reflects the name of this lesson.
   - Use the sound recording tools in Greenfoot to create and save five different sounds to this scenario.
   - Create a scenario with actor objects and a world. Edit the World constructor to create five new instances of an Actor object each time the world is re-initialized. Each instance should have a different keyboard key and sound file passed to them.