

Lesson 3: Working with Source Code and Documentation

Try It: Practice Activities

Objectives

- Demonstrate source code changes to invoke methods programmatically
- Demonstrate source code changes to write an IF decision statement
- Describe a procedure to display object documentation

Vocabulary

Identify the vocabulary word for each definition below.

	Multiple methods executed by Greenfoot in the order in which they are written in the program.
	Tells the object which methods to perform, and in what order.
	Describe what the code does to other humans. Greenfoot does not read it, nor does it impact the functionality of the program.
	The analysis process where Java technology programmers analyze a problem and then create objects to build a system, or more specifically, to solve the problem.
	Statement written to tell your program to execute a set of instructions only if/when a certain condition is true.
	A set of comments that the programmer can modify to let others who view the source code know what the class is, what it does, the author of the code, and the date it was last modified.

Project

1. Continue the project from Section 3 Lesson 2.
2. Open the Project named FrogFlyL2. From the Scenario Menu, choose "Save as" and use the name FrogFlyL3. This will keep a copy from Lesson 2 in case there is a need to refer to it later.
3. Compile the Greenfoot project.
4. Add one frog instance and one fly instance to your world.
5. Open the code editor for the Frog subclass.
6. Delete the move() and turn() statements from the previous section.

7. Write the if statement needed to have the frog move forward one step when the “up” arrow is pressed.
8. Compile the project and test the movement of the Frog.
9. Write the if statement needed to have the frog turn right 5 degrees when the “right” arrow is pressed.
10. Compile the project and test the movement of the Frog. Press both the “up” and the “right” arrow keys simultaneously and individually to test.
11. Open the code editor for the Fly subclass.
12. Delete the move() and turn() statements from the previous section.
13. Write the if statement needed to have the fly move forward one step when the “up” arrow is pressed.
14. Write the if statement needed to have the fly turn left 5 degrees when the “left” arrow is pressed.
15. Compile the project and test the movement of the Fly. Press both the “up” and the “left” arrow keys simultaneously and individually to test.
16. Save the project and exit.

Optional Practice

1. In the Act method for an Actor subclass, program the class to move 3 steps whenever the Act or Run buttons are clicked in the environment.
2. In the Act method for an Actor class, write an if-statement that makes an object turn and/or move if a specific condition is true.