Java Fundamentals
3-9: Abstraction Project

Objective – Use abstraction

- Add a parameter to a constructor
- Add random effect to actor
- Access a method in a world subclass

Open your project from lesson 8 (JF_V02_S03_L08)

Complete the following tasks:

1. Save your scenario as JF_V02_S03_L09PrjStudent
2. Add a property to the Block class called turnspeed that stores an integer value
3. Create a constructor in the Block class with the signature public Block(int maxturnspeed)
4. Add code to the Block constructor that sets the turnspeed field to a random value between –maxturnspeed and up to +maxturnspeed.
5. Modify the Block constructor so that if the turnspeed is set to 0, it is changed to 1.
6. You will now have to modify the RobotWorld class. Update the line - addObject(new Block(), 427, 145); to addObject(new Block(**2**), 427, 145);
7. Modify the turn(2) statement within the **block** class' act method to use the turnspeed variable instead of the value 2.
8. Add a property to the RobotWorld class called int currentMaxTurnSpeed = 2;
9. Add a property to the RobotWorld class called int currentLevel = 1;
10. Create a method in RobotWorld called setUpLevel with the signature public void setUpLevel()
11. Program the following pseudo code in RobotWorld in the method setUpLevel. This will add another rotating block with a random speed to the world every time we increase a level.

    If level equals 2

        Add 1 to currentMaxTurnSpeed

        Add new block at max speed of level

        Add 5 pizza pieces to world

    Else if level equals 3

        Add 1 to currentMaxTurnSpeed

        Add new block at max speed of level

        Add 5 pizza pieces to world

    Else if level equals 4

//Design your own level information here

12. Create a method in RobotWorld called increaseLevel with the signature public void increaseLevel()
13. Modify the increaseLevel method so that it increments the currentLevel field by 1.
14. Add a call to the method setUpLevel within increaseLevel.
15. Add the following code to the Robot class method detectHome when the robot successfully reaches the home.

    RobotWorld myworld = (RobotWorld)getWorld();

    myworld.increaseLevel();

    *note. getWorld() returns a reference to World Class Type, but we have added our own public methods and to access them we must get a reference to RobotWorld. So we cast our World as a RobotWorld type. This then gives us access to its public properties and methods.
16. Create a property in Robot called timer of type int.
17. In Robot Create a constant value for the maximum timer amount. This should be in the form of private final int MAXTIMER with a value of 1000.
18. Modify the constructor in Robot to set the value of the timer to the maxtimer value. (You can increase this if you want more time by increasing the value of MAXTIMER)
19. Create a method in Robot that decreases the timer value by one called updateTimer(). Add the updateTimer() method call to the Robots act method.
20. Modify updateTimer so that it calls the World method showText() to display the time left.
21. Update updateTimer so that if the timer is at 0 we use removeLife() to lose a life.
22. Create a public void method called resetPosition that sets the location of the robot back to its starting position.
23. Add a resetPosition method call to the removeLife() method so that the Robot is always returned to the starting position when it loses a life.
24. Remove the setLocation call from the detectBlockCollision() and detectWallCollision() methods as this will now be handled by the removeLIfe() call.
25. Add a resetPosition method call to the detectHome() method instead of using setLocation().
26. Create a public void resetTimer() method in Robot that sets the timer to the MAXTIMER value.
27. Add a call to the resetTimer() method as the last line of the if statement in the updateTimer() method.
28. Modify detectHome so that it calls resetTimer() after it has increased the level when we get to the Home.
29. Use the following pseudocode:

    Set timer = timer + 200

    To modify eatPizza so that every time a pizza is eaten we add another 200 to the timer.
30. Compile your scenario.
31. Save your scenario as JF_V02_S03_L09PrjStudent

Possible options to extend the project

- Add a random moving actor that will reset the robot when you collide
- Add additional obstacles
- Add your own characters
- Make your own sound files
- Add more walls

e.g.