CS427 Homework Assignment 6

Due date: ________________

General info: Turn in all commented problems by paper and by email (to dbahr@regis.edu with “CS427 Homework” in the subject line).

Problem #1: Alright, let’s have a little fun. Create an abstract class that is a RobotDance. This should contain stuff that is common to all basic dance moves that are made by your robot. You may decide what stuff is appropriate (common to all dances), but I insist on a minimum of the following:

1. A constructor that stores both motors as instance variables. Why? Because I think that the motors will be used in any robot dance. Hence, they are a common feature.
2. Add methods for doTheShuffle(), doTheSpin(), and/or other common dance moves (make them up if you don’t know any dance moves).
3. Finally, add an abstract method called doDance(). This is the method that child classes will implement to do a real dance.

Problem #2: Create an interface that is called Funky. This should contain an abstract method called getFunky().

Problem #3: Create a concrete child class that is a FunkyRobotDance. This class should do the following:

1. “Inherit” from both the Robot dance and the Funky interface.
2. Implement the abstract doDance() and getFunky() methods.
3. When implementing these methods, you should use some of the common dance moves already created in the parent class (RobotDance). You may also add new dance moves if you like.

Problem #4: Create a main class. Instantiate your FunkyRobotDance and call the doDance() method. Watch it get funky.

Problem #5: Now create another concrete child class that is a HipHopRobotDance. This class should implement the doDance() method but with a different sequence of dance moves. Also, I want this doDance() implementation to play its own dance music while it is dancing.
Note: It would require special effort to make it play music at the exact same time that it is dancing. So I am fine with code that does a dance move, then plays a note or two, then dances some more, then plays a note or two, etc.

**Problem #6:** Now for something different. Using the ultrasonic distance sensor, I want your robot to start dancing anytime someone (or something) gets close to the robot.

Details:

1. Download the “robot code” from my web site. For this problem, you will want to take advantage of the DistanceSensor and RobotMotor classes.
2. The DistanceSensor has a method called `getDistance()` that lets you ask for the distance to the nearest object. It returns a number between 0 (close) and 254 (far). A 255 means that the object is too far away to detect. Read the comments in that class for more details.
3. Your code should use a loop that periodically checks if anyone has approached the robot. In each loop, get the distance, then dance if necessary.
4. The ultrasonic sensor works like a sonar at very high frequencies. A burst of sound is sent out, and then it waits for the echo to bounce back. The echo will take a short period to arrive, so the sensor won’t be ready for another ping until the first echo returns. Therefore, to prevent erroneous readings you will need to add a short 20 millisecond delay in your loop (giving the sensor time to hear the echo). You can add a delay by using `Thread.sleep(x)` where x is the delay in milliseconds. `Thread.sleep()` must be in a try-catch clause because it throws an `InterruptedException` (read the Thread API for more details).

**Extra Credit #1 (10 points):** Make your robot dance when either (1) somebody gets too close, or (2) when somebody starts clapping or making any other loud noise. You will want to use the NoiseSensor.

**Extra Credit #2 (10 points):** Make some really cool synchronized dance/music moves. This requires that your music play *at the exact same time* as your dance. To do this you will need to run two “threads”. One
thread is the music, and the other thread is the dance. You can look up threading in the library or online.