

CS 444 - Individual Assignment 3

Deadline

11:59 pm on Thursday, February 26, 2015

How to submit

Submit your files for this individual assignment using `~st10/444submit` on nrs-labs, with a homework number of 3

Purpose

To practice programming as an individual with Java and with some of Project 2's leJOS concepts.

Important notes

- You should be able to do Problems 1 and 2 from the command line, or using the DrJava IDE, or from the command line on nrs-labs, or pretty much any Java IDE if you have another available.
- I do realize that you will not be able to test Problem 3's code before you submit it.

Problem 1 - 30 points

As a gentle warm-up, consider the posted version of the source code for the Java class `GameDie`. You are going to modify this class.

- Add an `"@author adapted by <your name>"` line to this class's javadoc comment.
- Add a selector/"getter" method `getNumSides` that expects nothing and returns the number of sides of the calling `GameDie` instance.
 - Don't forget to include an appropriate javadoc comment for this method!
- Add to class `GameDie` a private data field `numRolls`, whose value is intended to be the number of times the calling `GameDie` instance has been rolled, and a selector/"getter" method `getNumRolls` that expects nothing and returns the number of times the calling `GameDie` has been rolled.
 - be careful -- remember that these changes will require that appropriate changes also be made to the constructors and to the `roll()` method.

Submit your version of `GameDie.java`.

Problem 2 - 30 points

Since we haven't formally covered JUnit unit testing in this class, you need some other way to exercise if not test your new version of `GameDie`. So, you will write a Java application using your modified

GameDie class.

Write a Java application class in `TryMyDie.java` that:

- creates two `GameDie` instances, one using the no-argument constructor and one using the 1-argument constructor.
- prints to the screen the results of calling `getNumSides` for each of your `GameDie` instances in a descriptive message of your choice.
- rolls ONE of the `GameDie` instances to determine how many times to roll the OTHER `GameDie` instance -- and it rolls the OTHER `GameDie` instance precisely that many times, printing the result of EACH roll to the screen in a descriptive message of your choice.
- after that, prints to the screen the results of calling `getNumRolls` for each of your `GameDie` instances in a descriptive message of your choice.

As long as you do the above first, you may exercise your `GameDie` further if you would like.

Submit your version of `TryMyDie.java`.

Problem 3 - 40 points

This problem's purpose is to see if you are understanding the programs your team has been typing in and modifying for Project 2 -- it combines aspects of several of these.

Design and write a leJOS application class `GuideMe.java` that meets the following specifications, assuming you are using the end-of-Project-2 robot design with a touch sensor and an ultrasonic sensor.

The goal: By pushing the appropriate buttons on your robot, you can guide it from point A to point B without picking it up or changing its direction by hand, assuming that the path from point A to point B is not straight (but isn't overly narrow). For example, you could guide your robot around a chair or two in one of the aisles in BSS 313, or from BSS 313 to the Math/CS department office even if, say, several chairs or trash cans or other such obstacles were in that hallway.

MINIMUM SPECIFICATIONS: (up to 30 points)

- It starts by displaying `Guide Me` on the NXT screen.
- When you now press the NXT brick's Enter button, it clears the LCD screen and prints "Going Forward". And when you release the Enter button, it goes forward some constant distance.

When you now press the NXT brick's Left button, it clears the LCD screen and prints "Going Left". And when you release the Enter button, it rotates as needed to go to the left that same constant distance.

When you now press the NXT brick's Right button, it clears the LCD screen and prints "Going Right". And when you release the Enter button, it rotates as needed to go to the right that same constant distance.

When you press the NXT brick's Escape button, the program ends.

- You choose how long the constant distance is that is traveled during each button push, and declare that as a named constant private data field.

- You want this to be long enough that you make progress during each button push, but short enough that you can maneuver successfully just by pushing the buttons.
- You decide if you will use the default robot speed or if you will specify the robot speed in some fashion.
- You decide if you would like your robot to make any sounds during any of this.

NOTE: I have written, tested, and successfully run a version that does the above. I have NOT written a version with the following yet:

REMAINING 10 points: get one or both sensors involved!

- If your robot bumps into something, it could back up some constant amount (and then wait for the next button push).
- If your robot bumps into something, it could back up to where it started at the latest button push (and then wait for the next button push).
- If the ultrasonic sensor senses it is getting "too close" to something, it could stop (not go the entire usual distance) (and then wait for the next button push).
- If the ultrasonic sensor senses it is getting "too close" to something, it could back up (and then wait for the next button push).
- If the ultrasonic sensor senses it is getting "too close" to something, it could go left or right instead (and then wait for the next button push).

Obviously, you can't test your `GuideMe.java` yourself -- SO you will test this as an early stage in Project 3, in the newly-assigned Project 3 teams. I will of course understand that you haven't had a chance to test your `GuideMe.java` code before you submit it for this problem. But all of you on your team will be able to compare and contrast your solutions to this as you are all testing and debugging your versions as a team.

Submit your version of `GuideMe.java`.