CS 444 - Quiz 2 Review Suggestions - Spring 2015

last modified: 2015-03-10

- You are responsible for material covered in class sessions and individual assignments; but, here's a quick overview of especially important material related to this upcoming quiz.
- You are permitted to bring into the quiz a single piece of paper (8.5" by 11") on which you have **handwritten** whatever you wish on one or both sides. This paper must include your name, it must be handwritten by you, and it will **not** be returned.
  - Other than this piece of paper, the quiz is closed-note, closed-book, and closed-computer.
- This is a quiz on leJOS and Lego Mindstorms NXT basics that we have covered thus far, some from the reading, some from individual exercises, and some you should have been using in your projects and programming thus far in this class.
  - You are expected to still follow Java conventions and class coding standards in your quiz answers.

Lego Mindstorms NXT basics

- You should be familiar with the Lego Mindstorms NXT parts and their names from Chapter 8 in the course text -- while I do not expect you to have memorized the exact name of every individual kind of part, you should by this time be able to recognize basic examples of:
  - beams
  - liftarms
  - pins
  - axles
  - axle joiners
  - tires
  - gears
  - cables
  - motors
  - sensors
- Know the basics of following Lego directions (e.g., for each step, first gather all of the step's parts, THEN assemble them as illustrated).
- What, basically, do each of the 4 buttons on the NXT brick do?
- Know which ports on the NXT brick are motor ports, and which ports on the NXT brick are sensor ports; you should know the labels for these ports, also.
- How can you run a leJOS program that has been successfully loaded onto an NXT brick?
- How can you stop a program that is currently running on an NXT brick?
- How can you delete a program from an NXT brick?

leJOS basics

- Given the name of a leJOS public class's source code file, you should be able to give the specific, complete command to compile it.
  - This may result in more than one file, BUT (if successful) it will always result in at LEAST one -- you should be able to give the name of this file.
- Given the name of a collection of successfully-compiled leJOS classes in the current working directory, including the name of the exactly-one of those that is an application class, you should be able to give the specific, complete command to link these.
  - If successful, what is the name of the file that should result from this?
- Assuming that linking was successful, you should be able to give the specific, complete command to upload the linked result onto a Lego Mindstorms NXT brick.
  - You should be able to specify whether it is to be just loaded onto the brick, or whether it is to be loaded and executed.
- Which leJOS package has the most basic leJOS classes and interfaces? What are some of the classes in this package that we have used?
- Which leJOS package has the static method for delaying a specified number of milliseconds? Which class has this method, and what is this method?
- You should know the named constants for the 4 buttons on the NXT brick (and how to write these within a leJOS class).
- You should know the named constants for motors connected to NXT brick motor ports (and how to write these within a leJOS class); these named constants are instances of what class?
- You should know the named constants for sensors connected to NXT brick sensor ports (and how to write these within a leJOS class).
- How can you write a message to the NXT brick's LCD screen (there is more than one way to do this); how can you clear the LCD screen?
- You should know how to write a leJOS statement that will cause the current method to wait for any NXT brick button to be pressed before going on -- you should also know how to write a leJOS statement that will wait for a particular NXT button to be pressed.
- What does a tachometer within an NXT motor do? What can it measure? What are some of the tachometer-related methods in class `NXTRegulatedMotor` that you have used?
- What methods in class `NXTRegulatedMotor` let you cause a motor to go forward, to go backward, and to stop? What method can you use to determine if a motor is currently moving?
- What is the difference between the `NXTRegulatedMotor` methods `rotate` and `rotateTo`? When might you want to use `rotate`, and when might you want to use `rotateTo`?
  - What is the difference between the 1- and 2-argument versions of these methods? When might you want to use the 1-argument versions, and when might you want to use the 2-argument versions?