Program #2 Helper

Prof Bill - Apr 2018

Some help for coders on JavaFX and Program #2.
You guessed it. It's… coming soon!

Fri (the) Apr 13

Here's a nice JavaFX resource that emailed earlier in the week.
docs.oracle.com/javafx/2/ui_controls/overview.htm#

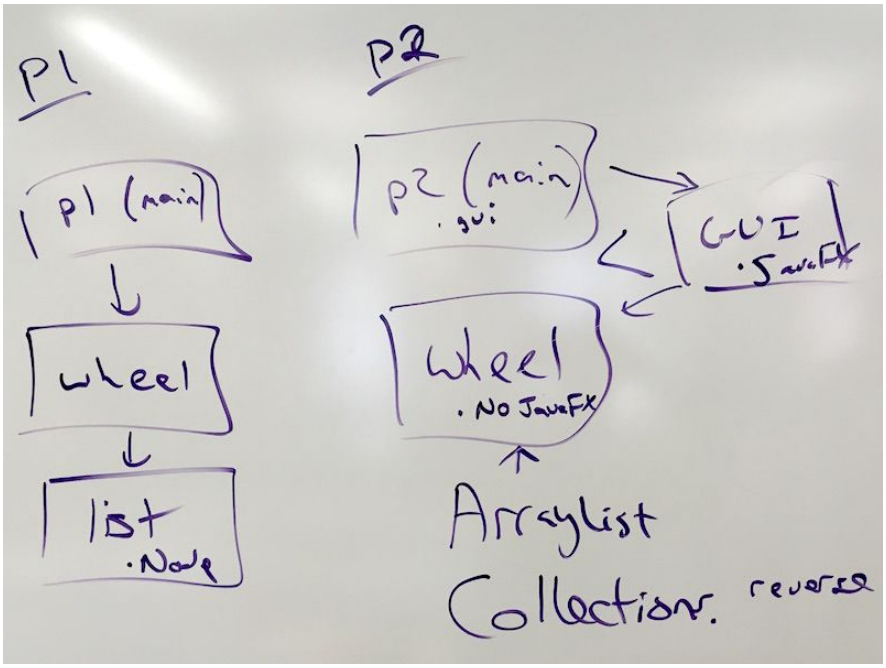
Hey, great brainstorm session in class yesterday. Thank you to Valerie S for the scribe-work. Here are the best and brightest:

P2 Brainstorm Ideas	
Disappearing spin button	Drop-down with library wheels
Flashing boxes instead of spinning wheel	Remove item
Commands in a drop-down	Help button
	Spinning. spinning.. spinning...
Wheel that spins (more on this below)	
Textbox w/ items (gray out as selected)	Pick image: orange or classic Subaru
Elevator animated chooser	Spinning: labels appear randomly on Sooby
Panel of commands/choices	
	Disable some commands when spinning
Spinning arrow of decision	Tooltips!
Command controls always visible	
Changing wheel colors	Show library and item choices on screen
	Load previous wheel
	Text field for input

Some of these ideas are based on sound GUI principles:

- ➔ When possible, show a list of options and select with the mouse, rather than typing
- ➔ Visually disable commands/options that are currently unavailable

Here's our design doodle from class:



P2 design:

- Setup is similar to P1
- Use ArrayList rather than building our own. Use Collector class for sorting and reversing lists. Use the JCF where possible to make yourself hyper-efficient.
- NO JavaFX in your Wheel class (or whatever you call it). Theoretically, we should be able to use your Wheel methods (ctor, add_tiem, spin, first_item, reverse, etc) to build a command-line interface if we wanted.
- Small main() classes are kind of an idiom in GUI. Kust cayin.
- Extra classes are fine… Blinker, Spinner, ProfBillRules, etc. The classes shown here are the core.

Don't forget our coding guidelines: wtkrieger.faculty.noctrl.edu/java_coding_guidelines.pdf
Quality code!
Start early. Small bites. Use the debugger. Email if you need help.
thanks...yow, bill

Sat Apr 14

All hail Nate N for his spinning wheel research. Here's his (excellent) code.
If you use any of Nate's solution, please include a comment with his URL in there!

github.com/nwnorris/NWheelSpinTest/blob/master/src/NWheelSpinTest.java

I'll try to cajole Nate N into a quick review of his approach during Tuesday's lecture.

Sun Apr 14

Wheel class - Start easy on a Sunday morning with my Wheel class.
Current methods include:

ctor, getWheelName(), setWheelName(), getWheelItems(),
size(), isEmpty(), firstItem(), lastItem(), randomItem(),
clear(), reverseItems(),
spin(), reload(),
toString()

Notice: 1) all my methods are wheel operations, 2) this code is super easy with ArrayList and other JCF methods, and 3) there's no GUI or JavaFX anywhere. That comes later.

I added a **Program2** class with **main()** and a simple test method for my new Wheel. I just create a Wheel and call my methods and print the results to confirm that they work.

Running GREEN - NetBeans has a tiny square in the upper right. It's green if your code is correct. I have 2 rules when coding in Java:

- My code is always **GREEN**
- My program is always running

Et tu?
thanks… yow, bill