

Refactoring (and maybe some Recursion)

2-23-2011

Opening Discussion

- Minute essays responses:
 - Why we write both versions of stacks/queues? Why we write basic data structures.
- Making Transforms works in Drawer.

Refactoring

- This is something that you do when you don't want to change the functionality of your code, but you want to change how it does something.
- You typically refactor your code when it “smells.” Here are a few of the many different smells.
 - Long method
 - Large class
 - Duplicate code
 - Shotgun surgery
 - Switch statements
- Scala tools don't yet refactor well, but the language does.

Recursion

- You should have learned about recursive functions in PAD1. A recursive function is simply a function that calls itself.
- You can use recursion to imitate loops, but we won't do that very often in C/Java/Scala. Where recursion comes in really handy is when a function needs to test more than one alternative at a time.
- This works nicely because the call stack remembers where you are in a given function so when you return back, you can take off from that point again.

Maze Solving

- One of my favorite recursive algorithms is maze solving. This is a special case of graph traversals which are common problems in CS.
- We'll use a 2D array of Ints as our maze and we can even put this into our drawing program.
- I want to write code to find the shortest path through a maze or count all paths through a maze.
- We can try to make this nice and graphical as well so it fits properly into our drawing program.

Formula Parsing

- Another one of my favorite recursive algorithms is formula parsing. This allows us to have the user type in a function and our code can evaluate it.
- We do this through “divide and conquer”. We split the formula in two across the lowest precedence operator then recursively evaluate the two halves.
- We can use this to put function plotting into our program if we give it the ability to handle a variable.

Minute Essay

- When is recursion better than loops and what makes it more powerful?
- We have the midterm Friday. Review session from 6:00-7:00.