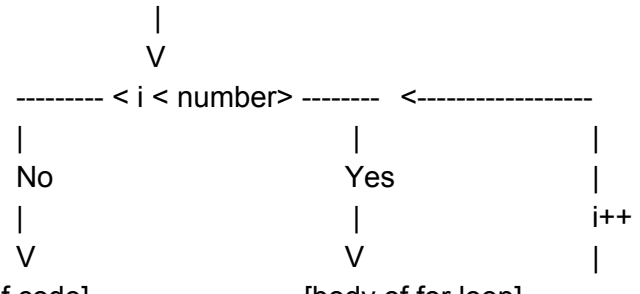
Weebo
Jackie Woo and Nikita Borisov

APCS1 pd01
2017-01-06

Final Project Proposal
Java Flowchart Visualizer

The goal of the Java Flowchart Visualizer is to create a program that reads through java code and converts it into a flowchart. For example, if the program saw a for loop, then, in the terminal, it will print out:



This project will utilize input output interactivity with the user. It will also use string methods in order to break up the code and what we learned about loops and different types of statements (such as declarations, initializations, assignments … etc.) to understand how to interpret the code that the user inputted. In order to display the result we will utilize a 2D array of strings. Different ascii characters will function as arrows and boxes for the question or body as of a loop.

We also want to keep track of the variables in the program and will do this by creating an ArrayList that will contains arrays of length 2 (each of a form [“var”, 5]). This will serve as similarly to a python dictionary and will allow us to update variables and print out a step by step visualization of how the code would execute.

We can also learn a bit about the character primitive and potentially use it in the flowchart to prevent a jagged looking 2D array. Once we are done with coding a loop processor we will move on to coding a general code processor that can have multiple loops and potentially even whole methods. Upon this expansion beyond a single for loop we might run into an issue of limited space in our 2D array. To resolve this we are prepared to use a super 2D array or an ArrayList of arrays.

The class JavaVisualizer will have an instance variable for the super 2D array and the arrayList of variable values. It will included methods for processing the input java code that is retrieved from the main method through the user. It will also have a method for updating the code one step at a time for every time the user enters a specific character, such as “Y”. This will simulate the ability of a user to press a “next step” button in a java visualizer.

We will start with a java for loop visualizer and expand from there.