Software Requirements – Fibonacci Assignment

Assignment Prompt:

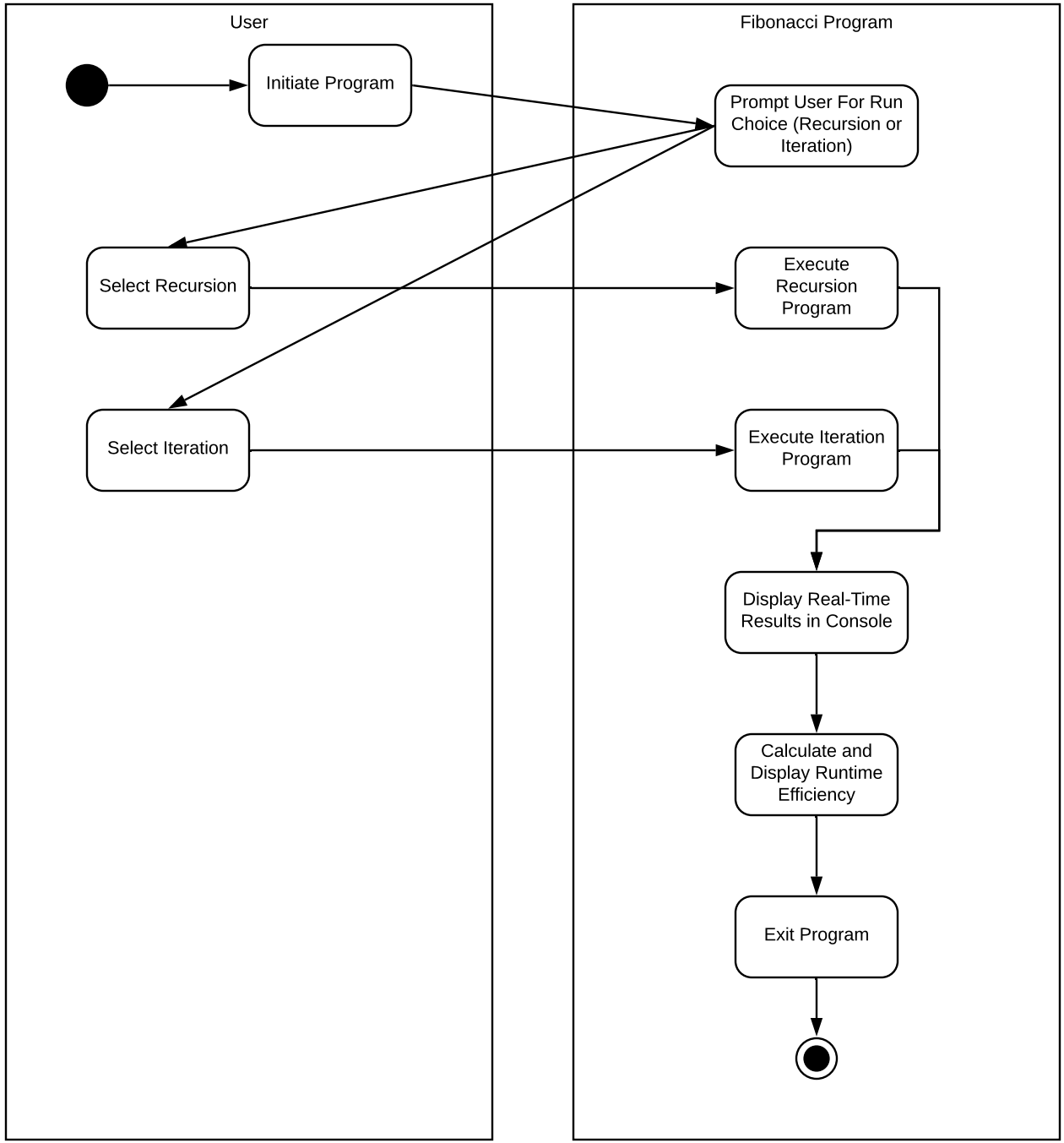
> Part I.
>
> Implement the Fibonacci function in both a recursive and iterative fashion. What's the runtime efficiency of each?
>
> You can look up sample programs - there are many out there - but for your own experience, please type yours from scratch.
>
> Turn in a chart of the results, with time on the Y axis, and input on the X axis, Please use nanosecond. long startTime = System.nanoTime(); This chart must not be hand written.

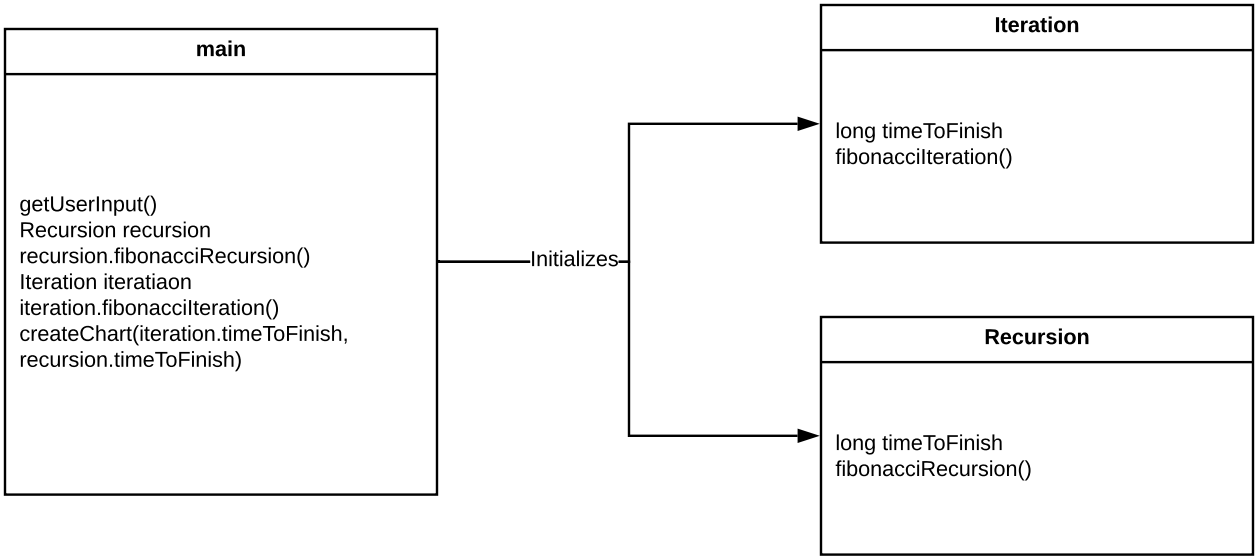
Requirements

- Program must run the Fibonacci function via two independent methodologies: recursion and iteration.
- Program must calculate the runtime efficiency of each, by printing out elapsed run time in nanoseconds.
- Program must produce elapsed run time in the form of a visual aide, comparing the two methods.
- Software must be well documented (non-funcitonal requirement).

Fibonacci Assignment Software Development Plans

1. Draft development plans document (this).
2. Create repository for software and documentation on GitHub.
3. Draft a requirements document for software.
4. Create an activity diagram showing process for how user is intended to interact with software.
5. Create a class-level UML diagram outlining software architecture/design.
6. Implement recursive Fibonacci solution.
7. Implement iterative Fibonacci solution.
8. Add timing calculator.
9. Add chart feature to graph method vs runtime (nanoseconds).
10. Create documentation outlining objects and properties within solution, as well as how to run the software.


User
Fibonacci Program
Initiate Program
Prompt User For Run Choice (Recursion or Iteration)
Select Recursion
Execute Recursion Program
Select Iteration
Execute Iteration Program
Display Real-Time Results in Console
Calculate and Display Runtime Efficiency
Exit Program


main
getUserInput()
Recursion recursion
recursion.fibonacciRecursion()
Iteration iteratiaon
iteration.fibonacciIteration()
createChart(iteration.timeToFinish,
recursion.timeToFinish)
Initializes
Iteration
long timeToFinish
fibonacciIteration()
Recursion
long timeToFinish
fibonacciRecursion()

Fibonacci Program Class Documentation

Classes

Main.java – The main class that initiates execution. This class extends Application (a JavaFX package), and contains the methods start, main and AlgorithmDecider.

FibonacciRecursion.java – The class that contains the logic needed to run the recursive version of the Fibonacci math. This class contains the methods Recursion and fibonacciRecursion, which return a long and int respectively.

FibonacciIterative.java – The class that contains the logic needed to run the iterative version of the Fibonacci math. This class contains the method Iteration, which returns a long.

Methods

Main

The **start** method is a JavaFX method that initializes the stage for the GUI, and populates the scene with the objects needed to show a bar chart with the Fibonacci runtime data. This is where the chart's axes and data series are created, as well as where the scene and stage dimensions are set.

The **main** method is the first method to run when the program is executed. This method immediately passes control to the AlgorithmDecider method, then prints to console, and finally passes control to the start method.

The **AlgorithmDecider** method is responsible for prompting the user for input, processing that input, and then determining which of the two Fibonacci methods to execute. This method uses a switch statement to process the user input, then instantiate either a Recursion object or a Iterative object based off of that input.

FibonacciRecursion

The **Recursion** method inside FIbonacciRecurison.java contains the process that calls math method required to print out the Fibonacci values. This method calculates and prints out the runtime in nanoseconds. This method will call fibonacciRecursion the number of times equal to the user-specified Fibonacci length value. This method returns a long value, which is the runtime in nanoseconds.

The **fibonacciRecursion** method is responsible for performing the math that calculates the Fibonacci values. This method is recursive, in that the method's return statement calls the method. The amount of recursive calls is controlled by the parent method, Recursion. This method returns an int, which is the Fibonacci value.

FibonacciIterative

The **Iteration** method inside of FibonacciIterative.java is responsible for the iteration method for calculating Fibonacci values. This method runs a for-loop for the user-specified number of loops. This method also returns a long, representing the runtime in nanoseconds.

Fibonacci Program Run Instructions

Purpose: This program will output Fibonacci values to the console obtained by two different approaches: one by using a recursive algorithm, and the other by using and iterative algorithm. A bar chart is also produced that compares the runtime of each method. The user can specify which algorithm to implement (or both) and specify how many values of the Fibonacci series they would like to see (up to a predetermined limit). Warning: This program uses JavaFX.

How to run:

After importing the project into Eclipse and making sure that all the appropriate JavaFX configurations have been set, run the program. The console will produce some text prompting the user for input. The input options are either “R” or “I”, for the recursive algorithm and the iterative algorithm, respectively. Entering any other value will cause the program to use the default setting, which runs both algorithms and compares the runtime of each. After entering some text, the console will now prompt the user for the number of Fibonacci values that will be printed to console, up to a predetermined maximum. Once these two pieces of information have been entered, the program will execute. The result should be some Fibonacci values in the console window, along with the number of nanoseconds that each algorithm ran for. A bar chart in a separate window will also pop up, showing the user the same runtime in nanoseconds for the two algorithms.