CSCB07 – Software Design

Lab 1

Objectives

- Getting familiar with Java
- Learning how to compile and run a Java program from a command line
- Getting familiar with Git

Logistics

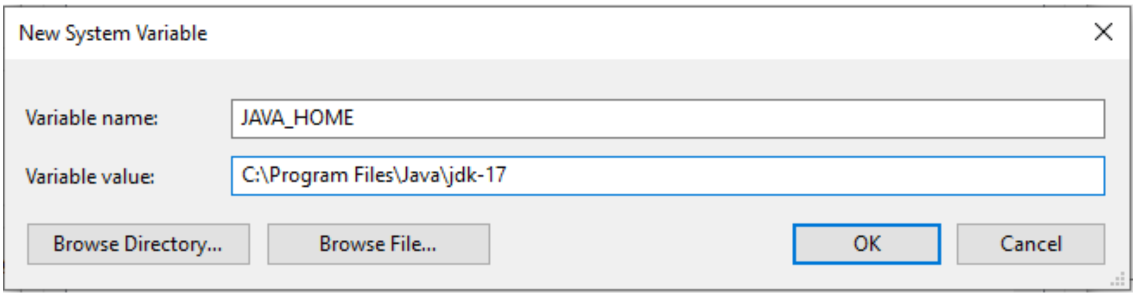
- This lab is worth 3% of the course grade and it will be supervised by your TA during the tutorial session of Week 2. If you encounter any problem while doing the steps listed in the next section, ask the TA for help.
- Attendance will be taken during the tutorial. If you are unable to attend, please send your TA an email explaining why and make sure to submit the deliverables by the due date. Failing to do so might result in a 10% penalty.
- The lab should be done individually.
- The due date is May 19, 2024.

Instructions

1. Download and install JDK:
   https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html

2. Windows users might have to set the JAVA_HOME environment variable as follows:

   Control Panel -> System and Security -> System -> Advanced system settings -> Advanced -> Environment Variables -> System Variables -> New and set JAVA_HOME by specifying the path to the JDK directory (the path on your machine might be different than the one shown in the example below)



New System Variable

Variable name: JAVA_HOME

Variable value: C:\Program Files\Java\jdk-17

3. If Git is not installed on your machine, download it and then install.
4. If you don't have a Github account, create one here.
5. Create a personal access token to be used for authentication. More information could be found here.
6. On Github, create a repo named **b07lab1** as per these instructions and make sure to make it public.
7. Navigate to the repo and create a new file: name it "info", add some information about the lab, and click the "Commit new file" button.
8. On your machine, do the following steps using Command Prompt/Terminal:
    a. Clone the repo using the command **git clone *RepoURL*** (*RepoURL* should be replaced by the actual URL of your repo)
    b. **cd b07lab1**
    c. Windows users:
    notepad Polynomial.java (answer "Yes" when asked if you want to create a new file)

    Mac users:
    touch Polynomial.java
    open Polynomial.java

    d. Develop class *Polynomial* as follows:
        i. It has one field representing the coefficients of the polynomial using an array of *double*. A polynomial is assumed to have the form $a_0 + a_1x^1 + \cdots + a_{n-1}x^{n-1}$. For example, the polynomial $6 - 2x + 5x^3$ would be represented using the array [6, -2, 0, 5]
        ii. It has a no-argument constructor that sets the polynomial to zero (i.e. the corresponding array would be [0])
        iii. It has a constructor that takes an array of *double* as an argument and sets the coefficients accordingly
        iv. It has a method named **add** that takes one argument of type *Polynomial* and returns the polynomial resulting from adding the calling object and the argument
        v. It has a method named **evaluate** that takes one argument of type *double* representing a value of x and evaluates the polynomial accordingly. For example, if the polynomial is $6 - 2x + 5x^3$ and **evaluate(-1)** is invoked, the result should be 3.
        vi. It has a method named **hasRoot** that takes one argument of type *double* and determines whether this value is a root of the polynomial or not. Note that a root is a value of x for which the polynomial evaluates to zero.
    e. Save the file
    f. Windows users:
    notepad Driver.java (answer "Yes" when asked if you want to create a new file)

    Mac users:
    touch Driver.java
    open Driver.java

    g. Add the following code, then save and close the file

```
public class Driver {
    public static void main(String [] args) {
            Polynomial p = new Polynomial();
            System.out.println(p.evaluate(3));
            double [] c1 = {6,0,0,5};
            Polynomial p1 = new Polynomial(c1);
            double [] c2 = {0,-2,0,0,-9};
            Polynomial p2 = new Polynomial(c2);
            Polynomial s = p1.add(p2);
            System.out.println("s(0.1) = " + s.evaluate(0.1));
            if(s.hasRoot(1))
                    System.out.println("1 is a root of s");
            else
                    System.out.println("1 is not a root of s");
    }
}
```

h. **javac Driver.java**

i. **java Driver**

j. Verify that the output you obtain is as follows:

```
0.0
s(0.1) = 5.8041
1    is a root of s
```

k. **git add *.java**
l. **git commit -m “added Polynomial.java and Driver.java”**
m. **git push**

9. Submit the URL of your repo to “Lab 1” on Quercus.

Evaluation

- Git: 1 pt
- Java code: 2 pts
- A 10% penalty will be applied for missing the tutorial with no valid excuse