CSCI 502 – W03 – Computer Programing I
Professor Bhatt

Outline and Description of Final Project - Fall 2020

Robert Trieste
917-806-6333
rtrieste@nyit.edu
https://github.com/bobtreehouse
www.linkedin.com/in/bobtrieste

GitHub link to working project:
https://github.com/bobtreehouse/nyit_csci502_profBhatt/tree/master/PrimeWorker

Project Name: PrimeWorker

Description: PrimeWorker is a Java console application that asks the user to enter a whole number between 2 and 1000. The application verifies that an Integer has been passed and then proceeds to calculate both the number of Prime numbers up to the ceiling of the user input and also to output to the console a list of the actual Prime numbers within the specified range.

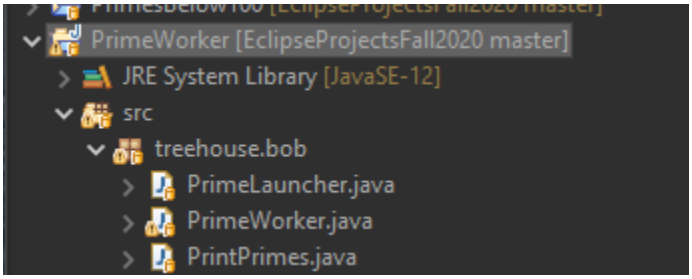
Concepts Implemented: use of Package to group 3 classes and outline Program Flow; Constructors to instantiate instances of the other classes to access their Methods; Private variables; Public Getters/Setters; Java Utility Scanner; Do-While and If/Else Loops to validate input and direct program flow; Logical Operator to check User Input; Parse Int Static Method of Wrapper Class Integer; Try-Catch block for error-handling; ArrayList for dynamically-sized array; Nested For-loop and modulus operator for calculating the Prime numbers; System output to display results to User.

Program Flow and Screenshots: PrimeWorker contains a Package with 3 Classes:

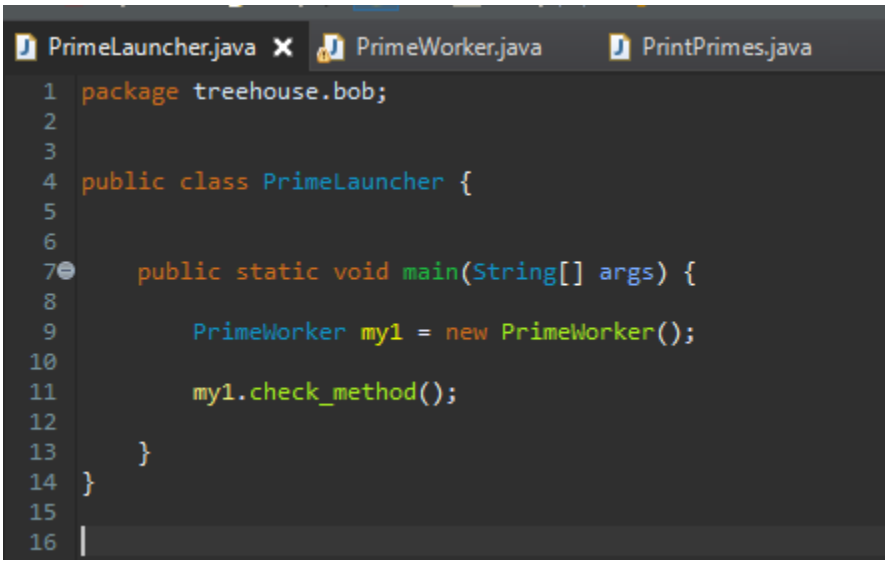
(1) PrimeLauncher Class: contains Main method of only 2 lines for simplicity.
(2) PrimeWorker Class: initializes variables to hold UserInput, has method 'check_method' to verify UserInput and then pass that input to the next Class and method.
(3) PrintPrimes Class: uses the Integer given from user and contains method 'numberofPrimes_method' to calculate the Primes and a dynamically sized ArrayList to hold them.

(cont'd with screenshots next)

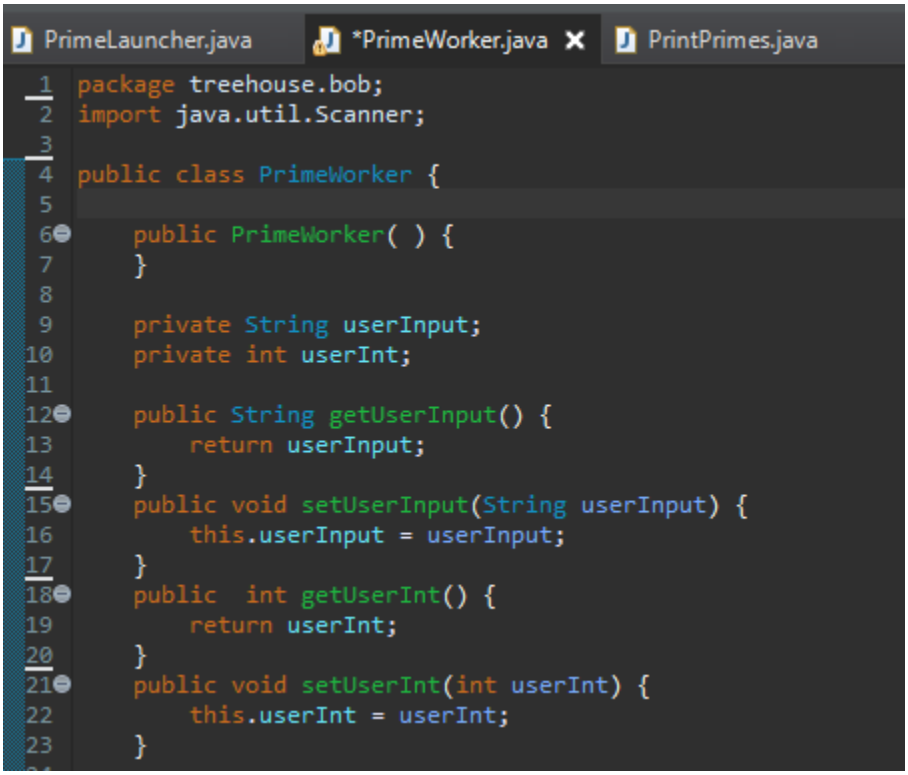
Package with Classes:



Main creates an Instance of the PrimeWorker class in order to make a method call of check_method:



PrimeWorker sets up placeholders in memory for variables of userInput and the parsed userInt and then Getters/Setters for program access to those elsewhere:



(cont'd with screenshots next)

PrimeWorkers' check-method takes UserInput; parses it to Int; simulates a CPU delay; creates an instance of the next object PrintPrimes:

PrimeLauncher.java | *PrimeWorker.java | PrintPrimes.java

```
24
25          void check_method() {
26              Scanner myScanner = new Scanner(System.in);
27
28          do {
29              System.out.printf("Please enter a number between 2 and 1000 and I will tell you the # of Primes "
30                      + "\nin that range and also print out those prime numbers for you.\n>");
31
32              userInput = myScanner.nextLine();
33              //validate
34              if(userInput.isEmpty()) {
35                  System.out.println("Nothing was entered. Please try again.");
36              }else
37
38              //the scanner took in a String - we need to convert that to an Integer:
39              userInt = Integer.parseInt(userInput);
40
41              /*Integer is wrapper class of primitives type int and parseInt() is a static method of wrapper class Integer
42               * which returns equivalent int or integral value of string given as parameter.
43               * parseInt() method would prompt NumberFormatException during runtime if it is become
44               * failed to convert string into int. can have NULLS unlike Strings; better for backend DB
45               * */
46              } while(userInt < 2 || userInt > 1000);
47              /*
48              so with the above we keep going back the "do" during the
49              "while" is IN PLACE
50               */
51          System.out.println("\nYou selected: " + userInput);
52          System.out.println("\nLet me check your input...");
53          //let's add a delay in the console output to simulate long computation or network latency
54          try
55          {
56              Thread.sleep(2000); //insert a pause in milliseconds
57          }
58          catch(InterruptedException ex)
59          {
60              Thread.currentThread().interrupt();
61          }
62          System.out.println("\nOK, looks good..getting your result...");
63            try
64              {
65                  Thread.sleep(2000);
66              }
67              catch(InterruptedException ex)
68              {
69                  Thread.currentThread().interrupt();
70              }
71            PrintPrimes pr1 = new PrintPrimes(userInt);
72            pr1.numberOfPrimes_method();
73
74          }
75      }
```

(cont'd with screenshots next)

PrintPrimes Class:

PrimeLauncher.java | *PrimeWorker.java | *PrintPrimes.java

```java
package treehouse.bob;
import java.util.ArrayList;

public class PrintPrimes   {

    public PrintPrimes(int limit) {
        this.limit = limit;
    }
    private int limit;
    public int getLimit() {
        return limit;
    }
    public void setLimit(int limit) {
        this.limit = limit;
    }
```

Finally the numberOfPrimes_method uses a nested For-Loop to calculate, store in ArrayList and then output results:

```java
    void numberOfPrimes_method() {
    ArrayList<Integer> values = new ArrayList<>();
            //loop through the numbers one by one
            for(int i=2; i < limit; i++){
                    boolean isPrime = true;
                    //check to see if the number is prime
                    for(int j=2; j < i ; j++){
                            if(i % j == 0){
                                    isPrime = false;
                                    break;
                            }
                    }
                    //if each time we loop through 'i' = isPrime then add 'i' to the ArrayList 'values'.
                    if(isPrime)
                        values.add(i);
            }
            //print out the ArrayList names 'values':
            System.out.println("\nThe number of Primes between 1 and " + limit + " are: ");
            System.out.println(values.size());
            System.out.println("\nThe Prime numbers between 1 and " + limit + " are: ");
            System.out.println(values);
    }
}
```

Results from console:

Console | Problems | Debug Shell

```
<terminated> PrimeLauncher [Java Application] C:\Program Files\Java\jdk\bin\javaw.exe (Dec 16, 2020, 6:59:11 AM – 6:59:20 AM)
Please enter a number between 2 and 1000 and I will tell you the # of Primes
in that range and also print out those prime numbers for you.
>125

You selected: 125

Let me check your input...

OK, looks good..getting your result...

The number of Primes between 1 and 125 are:
30

The Prime numbers between 1 and 125 are:
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113]
```

-end.