Creating Twin Primes and Hexagon Crosses

Background of the Project's Client

It is very important to know your client(s). In this case and project, this application you are building below will be for a fictitious mathematics professor that is 60+ years of age. While this may not affect your programming the application and the solutions, these facts greatly influence the GUI interface required for this project.

Background of Twin Primes

```
A "prime number" is any number that is divisible by only 1 and itself.
    Examples: The following are prime numbers:
     2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71,
     73, etc…

A "Twin Prime" consists of 2 prime numbers with a single number in between them.
    Examples: The following are "twin primes":
    3 and 5        (i.e., the single number between them is 4);
    5 and 7        (the number between them is 6.)
    11 and 13;   (the number between them is 12.)
    17 and 19;   (The number between them is 18l)
    29 and 31;   (The number between them is 30.)
    41 and 43;   (The number between them is 42.)
    59 and 61;   (The number between them is 60.)
    71 and 73, etc...

A "Hexagon Cross" twin prime is a twin prime that consists of 2 twin primes with the
following characteristic: If the number between a twin prime is some number N, and
there exist another twin prime such that 2N is the number between a twin prime, then
these two numbers form a Hexagon Cross.

    Examples: The following are "N"s that satisfy this condition (and are the first
couple of Hexagon Crosses that can be found in the number system.:
    6 and 12       (i.e., 5 and 7 is a twin prime AND 11 and 13 is a twin prime)
    30 and 60      (i.e., 29 and 31 is a twin prime AND 59 and 61 is a twin prime)
    660 and 1320  (i.e., 559 and 661 is a twin prime AND 1319 and 1321 is a twin
                              prime).

Obviously, these "N"s are few and far between!!! (And they take a long time to find
without using a computer!)

The Hexagon Cross "N"s that I've found so far (and I've found these "by hand") are:
6, 12;
30, 60;
660, 1320;
810, 1620;
2130, 4260'
```

```
2550, 5100;
3330, 6660;
3390, 6780;
5850, 11700;
6270, 12540;
10530, 21060; and
33180, 66360


The program needs to:
Create a file (result.txt) and display (using a GUI textbox) that contains as many
"N"s as can be found. The client should be able to select the number of "N's"
viewable. (100, 1000, 10000)
```

Requirements of the Project
1. Created in Java
   a. version 13
   b. whatever IDE you wish (suggest Eclipse)
2. Code requirements
   a. A Collection of TwinPrimes and HexCrosses described above
   b. Application interface using JavaFX
      i. display Hexagon Crosses
      ii. the overall display and features are up to you
      iii. can find notes and PlayList here:
         1. [PlayList](#)
         2. [Notes](#)
3. Classes Required
   a. Driver.java
   b. FileIO.java
      i. read in Twin Primes
      ii. write Hexagon Crosses
         1. to results.txt
         2. results printed in the same directory
   c. Prime.java
      i. used in TwinPrimes.java
   d. TwinPrimes.java
      i. think of this as the application
   e. Other classes as you need
4. Documentation for the Client
   a. in PDF form, called Readme.pdf
   b. how to run the program, remembering your audience
5. Documentation for the next set of developers
   a. Assume this project will be further developed
   b. This will be documentation that will explain each working class you developed for your project.
   c. For each file:
      i. Explain how this particular class is a part of the project
      ii. Explain what each non-getter/setter/toString function does
         1. Use this for your normal comments you should be doing anyway!

d.   in PDF form, called Developer.pdf

Submission of the project

The entire project will be zipped and uploaded in Canvas. Please have the PDFs in the base directory: not buried within a directory. (So easy to find.)