Criterion C –Development

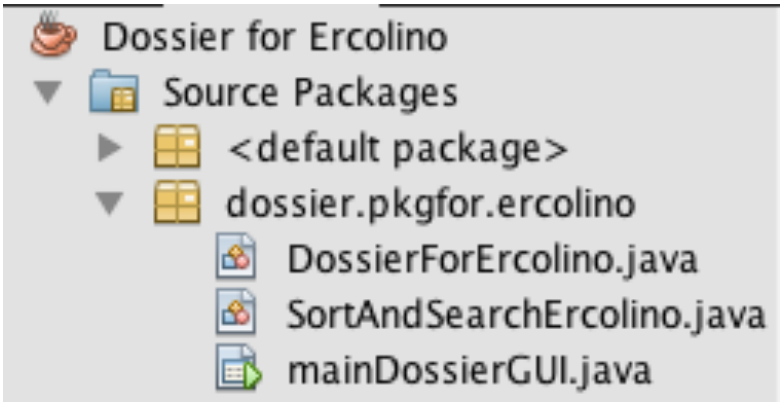
A Short Introduction:
I created my program using Netbeans, which allowed me to design a user-friendly Graphical User Interface (GUI) database using Swing tools. The user is able to input her information on what she has planted into the GUI, save it, and open it when it's needed.

Summary List of All The Techniques
-Parameter passing
-For loop
-While loop
-Do while loop
-Method returning a value
-Arrays List
-Users defined objects made from an OOP "template" class
- Encapsulation of private methods that work on public attribute of a "template" class
-Making an array of objects
-Simple selection (if/else)
-Sorting (bubble sort etc.), and in particular sorting an array of objects based on one key attribute
- searching (linear search, binary search…)
- saving to a file
- opening a file to a table
- error handling (for example catching a divide by 0 error, or a null pointer while using an array of object…)
-Use of a flag value (such as -999, or "not set yet")
-Overloaded constructors, which work differently depending on the parameters sent
-Parsing a file using String Tokenizer
-GUI elements including: Tables, Combo Box, Radio Button, Check Box, Button.
-Logical comparison, compound conditions (&& = and)
-Buffer reader, buffer writer
-Read In
-Try and Catch
-J File Chooser, File Writer, File Reader

Structure the Program:
In my program I divided my tasks into four separate classes. I created my GUI class along my "template class", which is called DoccierForErcolino. I also had my sorting and searching class. It makes sense to divide my classes like this because it made my trouble shooting easier.

The main GUI Class, mainDossierGUI, is where I coded all the important properties and where I created my GUI, which includes a table, text fields, and combo boxes and more. Here you will also find the save and open options, this will be useful for

the user since she can keep her information in a safe place. In the "template class" you will find the attributes, the constructor, the gets and sets. Lastly, I created a class called SortAndSearchErcolino, in this class I had all the methods for searching and sorting. I'm thankful I divided the classes this way since I'm now able to access a new instance of this class rather than copying and pasting a long code into my GUI class. These classes made it extremely helpful to use, for me, and hopefully it will allow the user to enjoy her program, keeping her organized.



Data Structures Used

ArrayList is a class in java that allows you to add on elements to an array as a non-static implementation of an array. An ArrayList is like a "package"; instead of copying and pasting everything inside the "package" every time, we send the "package" name, which is the ArrayList. In this program there are two ArrayLists named VegetableOrPlant: one for Spring and another for the other seasons (Other), each list included vegetables and fruits.

The vegetables and fruits are my temple class, I have two constructors, one is called Spring which consists of 9 attributes, e.g. amount of yield. The other class called Other has 6 attributes such as dateOfPlanting. I chose an ArrayList over an Array since in the beginning I wasn't sure how many vegetables and fruits the user would input, therefore an ArrayList makes sure that the user can add more to the list.

A FileWriter created a Writer object that you can use to write to a file, this is used in my program to allows the user to save and then open her information, using FileReader.

Main Unique Algorithms

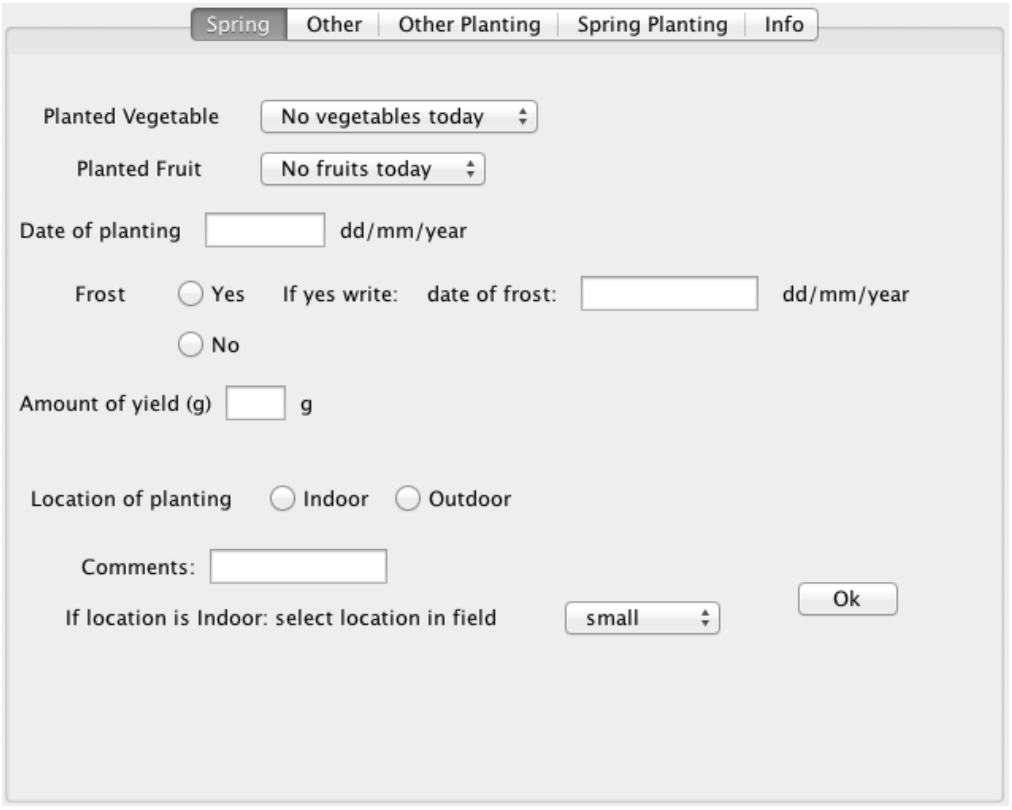
The program I created consists of one algorithm. In the GUI the user inputs the information such as what vegetable she planted on the day, when the information is put in its place, the user presses the button OK and a new instance of the vegetable class is made. The information disappears from the text fields, and it is transferred to the GUI-table. The user has the option to save the information permanently by

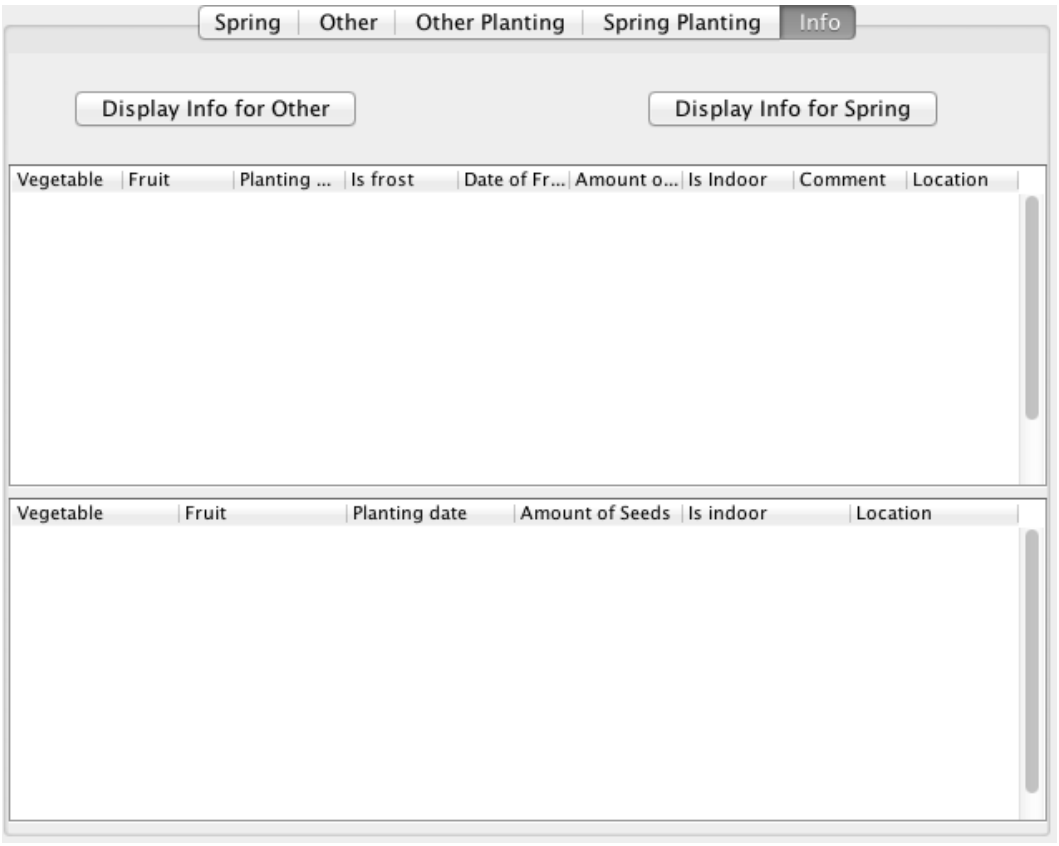
clicking Save As wherever she wants, using FileWriter and BufferedWriter, and open the information when it is needed, using FileReader by clicking the Open tab.

The part of my program that I could not finish but I planned, was determining the specific time which was best to plant a certain fruit or vegetable:
-When the information in inputted, and Spring is set to true
-It arrays through the past years and determinates which one is the closest time, using Searching and Sorting class which I created specifically for this.
-Then, the frost date is subtracted from the past year
-Calculating the closest 3 years with the smallest difference.
-It is sorted by the biggest harvest.
-Picks one of the years with the biggest harvest

User Interface/ GUI Work
The GUI application offered in NetBeans allowed me to create an efficient project and user- friendly. By using Java Swing I was able to add components to my project such as ComboBoxes, Radio Buttons, Text Fields, Tables, Check Boxes and more.

Java Swings allowed me to create a complex GUI, it was very efficient for me since the code was now simple to write, and for the user since it has the properties that the she asked for. I was able to change the amount of vegetables and fruits in each Combo Box by going to properties.
The image below is a screenshot of my GUI, here is where the user inputs the information.

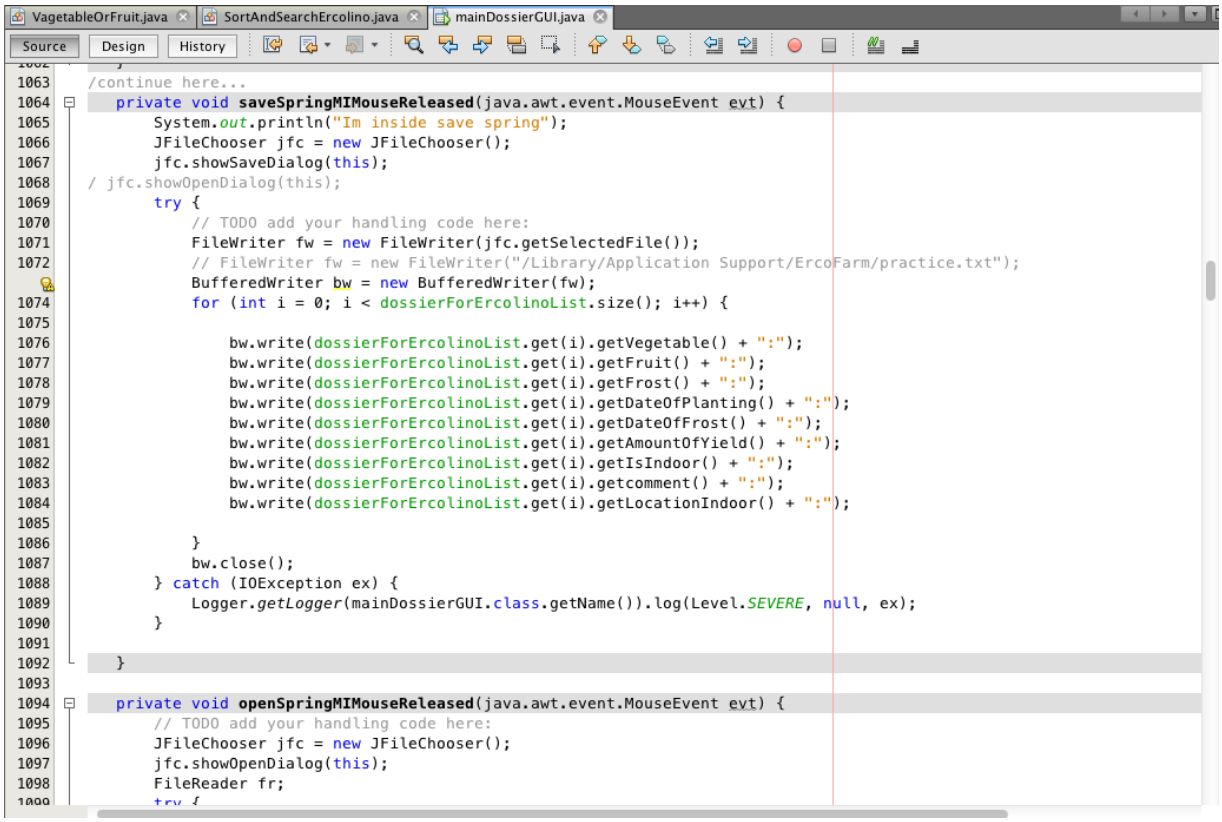




Software Tools Used

The software used to create the project was Netbeans. In our Computer Science IB course we are taught the Java language and how to collaborate with Netbeans. Netbeans is the most popular Integrated Development Environment (IDE) and is used by programmers worldwide. I had a great opportunity to work for a client and build a program. As I mentioned before, Netbeans is very efficient because of the GUI, which allowed me to include Java Swing components.

Below you can see a screenshot of the coding I created in Netbeans and the working environment. This code is when the user wants to save her information.




```
1063  /continue here...
1064      private void saveSpringMIMouseReleased(java.awt.event.MouseEvent evt) {
1065          System.out.println("Im inside save spring");
1066          JFileChooser jfc = new JFileChooser();
1067          jfc.showSaveDialog(this);
1068    / jfc.showOpenDialog(this);
1069          try {
1070              // TODO add your handling code here:
1071              FileWriter fw = new FileWriter(jfc.getSelectedFile());
1072              // FileWriter fw = new FileWriter("/Library/Application Support/ErcoFarm/practice.txt");
              BufferedWriter bw = new BufferedWriter(fw);
1074              for (int i = 0; i < dossierForErcolinoList.size(); i++) {
1075
1076                  bw.write(dossierForErcolinoList.get(i).getVegetable() + ":");
1077                  bw.write(dossierForErcolinoList.get(i).getFruit() + ":");
1078                  bw.write(dossierForErcolinoList.get(i).getFrost() + ":");
1079                  bw.write(dossierForErcolinoList.get(i).getDateOfPlanting() + ":");
1080                  bw.write(dossierForErcolinoList.get(i).getDateOfFrost() + ":");
1081                  bw.write(dossierForErcolinoList.get(i).getAmountOfYield() + ":");
1082                  bw.write(dossierForErcolinoList.get(i).getIsIndoor() + ":");
1083                  bw.write(dossierForErcolinoList.get(i).getcomment() + ":");
1084                  bw.write(dossierForErcolinoList.get(i).getLocationIndoor() + ":");
1085
1086              }
1087              bw.close();
1088          } catch (IOException ex) {
1089              Logger.getLogger(mainDossierGUI.class.getName()).log(Level.SEVERE, null, ex);
1090          }
1091
1092      }
1093
1094      private void openSpringMIMouseReleased(java.awt.event.MouseEvent evt) {
1095          // TODO add your handling code here:
1096          JFileChooser jfc = new JFileChooser();
1097          jfc.showOpenDialog(this);
1098          FileReader fr;
1099          try {
```