Introduction
I programmed an organizational database using Netbeans. I achieved this by using the Java programming language and making us of the Graphical User Interface using the capabilities of Swing.

Summary List of All Techniques:

- Private and public attributes
- Parameter passing
- For loops
- Array lists
- Method returning a value
- User defined objects made from an OOP "template" class
- Encapsulation of private methods that work on public attribute of a "template" class
- Array list of objects
- Simple and compound selection
- Saving to a file
- Opening a file to a table
- Parsing using a StringTokenizer
- GUI tabs
- Use of a flag value
- BufferedReader and readLine
- BufferedWriter
- FileReader
- FileWriter
- FileChooser
- Try and catch phrases
- GUI Swing Objects (e.g. Button, Label, Check Box, Combo Box, Table etc.)
- Template classes
- Use of classes

Structure of the Program
My program consists of five different classes. 1) MainDossierGUI 2) tenantsInformation 3) repairInformation 4) inputPropertyInformation and 5)dueDatesInformation. The first class, MainDossierGUI, has most to do with the design and what it does. This class the where all the 'magic' of the program comes to. Adding, saving, and opening information is all under the MainDossierGUI. Also, every Swing object is coded within this class. The other classes are managing the attributes. The reason they are split into four classes is because each piece of information under the class is in one tab of the GUI. For example, all the tenants' information that is taken from the user is part of one class. The repair information does not have a lot to do with a tenant; hence it is given a separate tab (in the GUI) and class.

Data Structures Used
An ArrayList is a class in Java that allows you to add on elements to an array, and so it is a non-static implementation of an array.

- ArrayList of tenants information: this allows the user to add new tenants. The tenants information template class consists of 8 attributes. E.g. tenants name, tenants address, etc.
- ArrayList of repair information: this allows the user to add each repair done in a house/apartment. The repair information template class consists of 6 attributes. E.g. repair type, description of problem, etc.
- ArrayList of property information: this allows the user to add new property information. The property information template class consists of 4 attributes. E.g. property type, development cost, etc.
- ArrayList of due dates information: this allows the user to have an organizer. The user can add a new task everyday and write the information for it such as start date, due date, and importance of task. The due dates information template class consists of 4 attributes. E.g. start date, due date, etc.

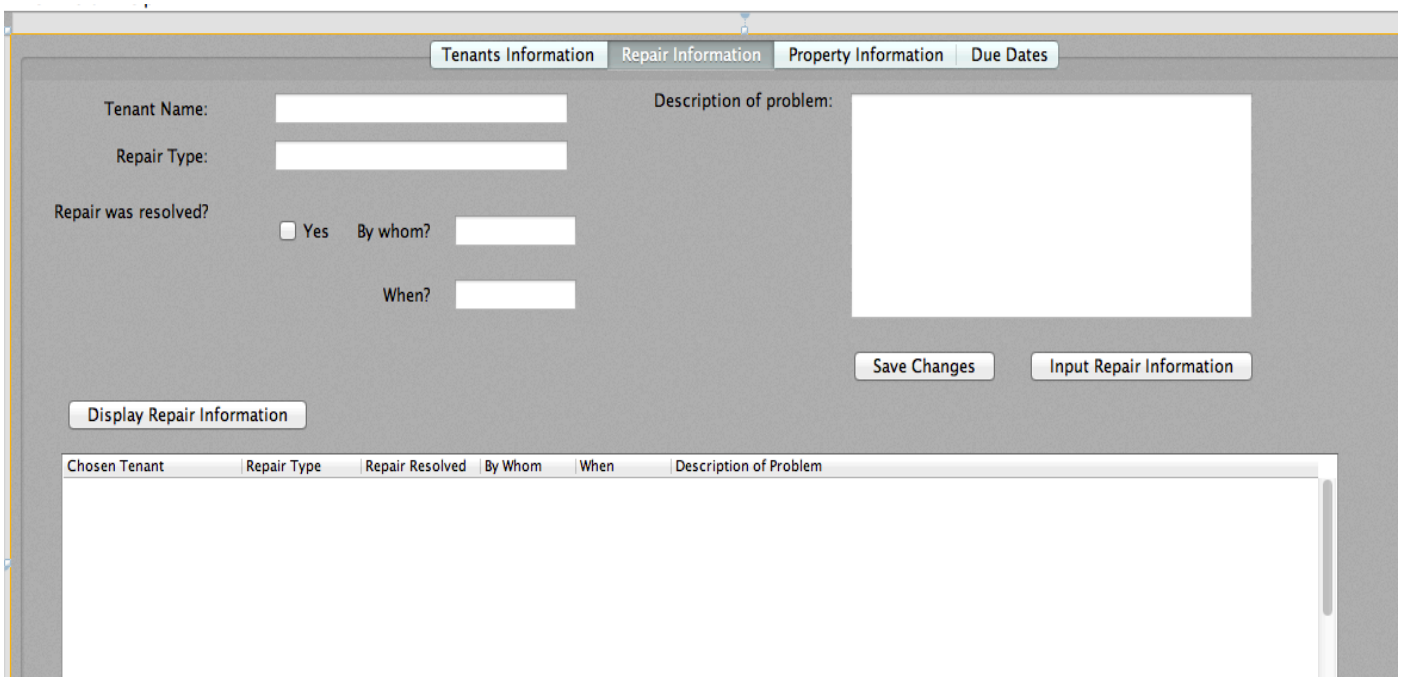
I chose to use an ArrayList instead of an Array because the user is dealing with new information weekly. An Array in a way is limiting the amount of data, while an ArrayList is ongoing. I have also used FileWriter. This allows for the user to save the new information added onto a file permanently.

Main Unique Algorithms

The user the program has been made for requested a program that makes him in charge of many of the functions. For example, in the first tab (tenants information tab), the user inputs all the text fields. When he clicks on 'add tenant', the new information is added onto the ArrayList for tenants information. When the user clicks on 'save changes', the file is automatically saved into a file in the Application Support folder of their computer. Another function this tab can do is opening the file. If the user click on 'File' then 'Open Tenant Info', then the information that had been automatically saved previously will be displayed on the table. The same functions work for the repair information tab and the input property tab. The last tab (due dates information tab) has two functions. After the user inputs all the due date information, he clicks on 'add'. Once done so, the information is displayed, added to the ArrayList of due dates and saved. The second function this tab can do is the same as the others, which is opening the file. Once the user clicks on 'File' then 'Open due dates info', their due dates will be displayed in the table.

User Interface/GUI Work

In my GUI I used different types of Java Swing components. These include text fields, labels, buttons, tabs, Combo boxes, check boxes, and data tables. I used certain components depending on what is most commonly used in applications I use. For example, usually names and addresses are typed into Text Fields, hence that is what I used. Here is a screenshot of a tab that uses the most variety of Java Swing components in my application.



As seen above, not many components were used, but that's because they weren't needed. This tab consists of Text Fields, Check box, buttons, and a data table.

Software Tools Used

To develop my program, I used NetBeans. NetBeans has been helpful throughout this process, mostly because of catching errors. Once it does so, it also provides the user with possible solutions. It not only helped me as a programmer, but also helped me use prior knowledge of computing in order to fix my syntax errors. Here is a screenshot of my NetBeans working environment:

```
    private void saveTenantsInfoButtonMouseReleased(java.awt.event.MouseEvent evt) {

        // JFileChooser jfc = new JFileChooser();
        //jfc.showSaveDialog(this);

        //This method is the event of a button. Once the button is released, the tenants information will be automically,
        //saved onto a file using FileWriter into the computer.
        try {
            //    FileWriter fw = new FileWriter(jfc.getSelectedFile());

            //String userHomeFolder = System.getProperty("user.home");
            FileWriter fw = new FileWriter("/Library/Application Support/MainDossierForBro/Tenants-Information.txt");
            //BufferedWriter bw = new BufferedWriter(fw);
            BufferedWriter bw = new BufferedWriter(fw);
            System.out.println(tenantsInformationList.size());
            for (int i = 0; i < tenantsInformationList.size(); i++) {

                bw.write(tenantsInformationList.get(i).gettenantsName() + ":");
                bw.write(tenantsInformationList.get(i).gettenantsAddress() + ":");
                bw.write(tenantsInformationList.get(i).gettelephoneNumber() + ":");
                bw.write(tenantsInformationList.get(i).getemailAddress() + ":");
                bw.write(tenantsInformationList.get(i).getdateLeaseSigned() + ":");
                bw.write(tenantsInformationList.get(i).getdateLeaseEnds() + ":");
                bw.write(tenantsInformationList.get(i).getrentPaid() + ":");
                bw.write(tenantsInformationList.get(i).getservicesPaid() + ":");

            }
            bw.close();

        } catch (IOException ex) {
            Logger.getLogger(MainDossierGUI.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
```