Criterion C - Development

Introduction

For my Computer Science Internal Assessment, I mainly used the Netbeans program which is a highly integrated environment for Java users to develop my program. In the NetBeans program, I used Java's swing tool to make my Graphical User Interface to further improve the user-friendly interface and environment when entering the program.

Words: 52

Summary list of all Techniques

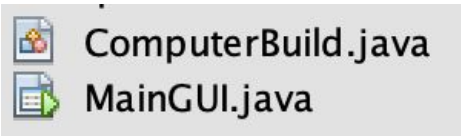
In my program the list of techniques I used was:

- Simple and compound selection (if/else)
- User-Defined Objects made from an OOP "template class"
- ArrayList
- An array of objects
- Error Handling
- GUI Tabs
- GUI Popup menus
- GUI Error menus
- Use of flag a flag value ("null", 0.0)
- Imported Javax.swing jar
- Imported java.io jar
- Imported java.util jar
- Imported org.apache jar

Words: 10

Structure of the program

What:
For my class structures, I only have 2 main classes that I worked with which were ComputerBuild and the MainGUI. In the ComputerBuild class, it consists of all the gets and sets that were needed in my program, with this class it's able to create objects of the ComputerBuild and initialize the variables of the object using the constructor.



Why:
The reason why I split it into 2 classes is that it has great benefits which include encapsulation, inheritance, and polymorphism. With encapsulation, it helps by protecting the group of code by grouping the data attributes together, and their methods together which make the attributes private. While inheritance, the subclasses are able to "inherit" the data and action for a group of objects which in my case is the MainGUI inheriting the ComputerBuild class. Finally polymorphism, a unique method which its actions can have the same name but different lists and parameters. All these 3 amazing features will make the coding experience much better as it improves the stability and reliability of programs, avoids duplication of codes between classes and it just allows only one external class to use the same method on all the subclasses.

Words: 195

Data Structures Used

What:
For my data structures, I mainly used only ArrayList and Arrays of objects which were present in both classes. I used an ArrayList of Computer Build consisting of computer parts which include eight attributes such as CPU, CPU Cooler, GPU, Motherboard, Storage, Case, Power Supply and the budget. Within the attributes, I also created another list of attributes of cost and score of each product excluding case, power supply, and budget.

```
private ArrayList<ComputerBuild> computerparts = new ArrayList<ComputerBuild>();
```

Why:
The reason why I used ArrayList instead of Array is that one of its main features is that it is dynamic in size as it can grow to accommodate more sizes which helps with my program as I have 2 existing presets of lists of the highest spec or lowest spec computer parts, where if I used array I can't change it's length once created. The other reason is that Array is known to take more memory than ArrayList for storing the same amount of elements or objects. Via the

principle of abstraction, I do not have to worry about the implementation of details of the class and this will make things much more convenient as I do not have to implement ADT by myself, reliable since classes are tried, true and tested and finally, consistency throughout the whole program since programmers around the world are using the same method names working in the same expected and understood ways.

Words: 237

Main Unique Algorithms

What:
In my program, I included unique algorithms such as taking the cost and performance table of computer parts to another program called Microsoft Excel and adding JOptionPanes.

Shows Error Pane:
JOptionPane.showMessageDialog(null, "Please fill in the CPU section", "Error", ERROR_MESSAGE);

Shows Confirm? Pane:
JOptionPane.showConfirmDialog(f, "Are you sure? You will have to go back to the information page.") == JOptionPane.YES_OPTION

With that, I had to install apache poi 4.1.2 to import all the necessary .jar files to fully execute this export. I started coding this by

- Creating a new XSSFWorkbook and XSSFSheet

```
XSSFWorkbook wb = new XSSFWorkbook();
XSSFSheet ws = wb.createSheet();
```

- Then I load the data to the treemap by inputting which will make me able to get each cell value in each row in the JTable.

```
TreeMap<String,Object[]> data = new TreeMap<>();
data.put("0", new Object[]{partsTable.getColumnName(0),partsTable.getColumnName(1), partsTable.getColumnN
data.put("1", new Object[]{getCellValue(0,0),getCellValue(0,1), getCellValue(0,2), getCellValue(0,3)});
data.put("2", new Object[]{getCellValue(1,0),getCellValue(1,1), getCellValue(1,2), getCellValue(1,3)});
data.put("3", new Object[]{getCellValue(2,0),getCellValue(2,1), getCellValue(2,2), getCellValue(2,3)});
data.put("4", new Object[]{getCellValue(3,0),getCellValue(3,1), getCellValue(3,2), getCellValue(3,3)});
data.put("5", new Object[]{getCellValue(4,0),getCellValue(4,1), getCellValue(4,2), getCellValue(4,3)});
data.put("6", new Object[]{getCellValue(5,0),getCellValue(5,1), getCellValue(5,2), getCellValue(5,3)});
data.put("7", new Object[]{getCellValue(6,0),getCellValue(6,1), getCellValue(6,2), getCellValue(6,3)});
```

- To retrieve the data from my JTable I created a new public class which consists of:

```
return partsTable.getValueAt(x, y).toString();
```

- Then I write to the sheet by inputting:

```
Set<String> ids = data.keySet();
XSSFRow row;
int rowID = 0;
```

- Afterward, I'll get the data as per key:

```
for(String key: ids){
    row=ws.createRow(rowID++);
    Object[] values = data.get(key);
    int cellID = 0;
    for(Object o: values){
        Cell cell = row.createCell(cellID++);
        cell.setCellValue(o.toString());
    }
}
```

- Finally Ill directly locate the directory path I want the excel file to be. I had to add a few catch clause as well.

```
try{
    FileOutputStream fos = null;
    try {
        fos = new FileOutputStream(new File("/Users/18604/Desktop/computer.xlsx"));
    } catch (FileNotFoundException ex) {
        Logger.getLogger(MainGUI.class.getName()).log(Level.SEVERE, null, ex);
    }
    wb.write(fos);
    fos.close();
} catch (IOException ex) {
    Logger.getLogger(MainGUI.class.getName()).log(Level.SEVERE, null, ex);
}
```

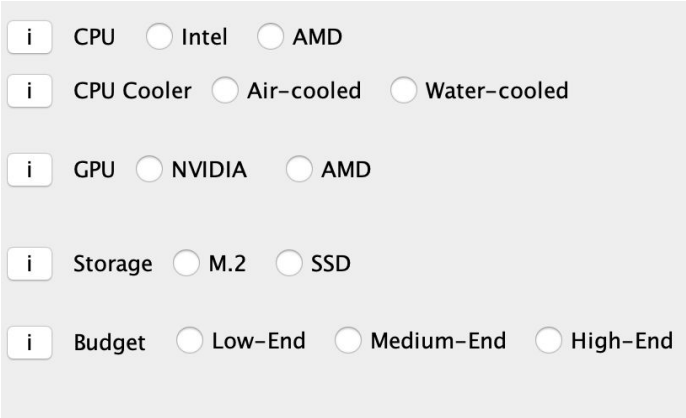
Why:
I included this unique algorithm in my program because it makes users easier to review the parts, cost and performance in a more efficient program, as in this case, excel.

Words: 120

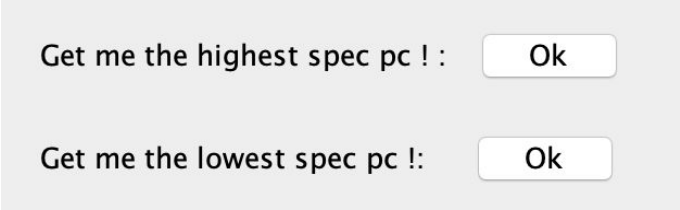
GUI Work

What:
In my program, I used multiple GUI features to fully improve user friendly towards my client and make it really simple to understand.

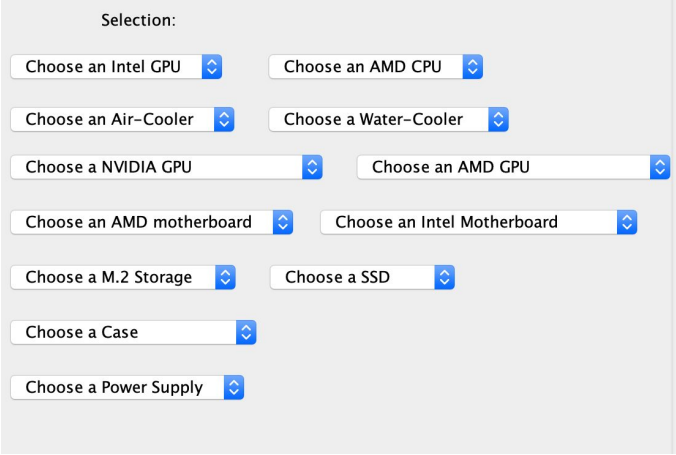
I used Radio Buttons.



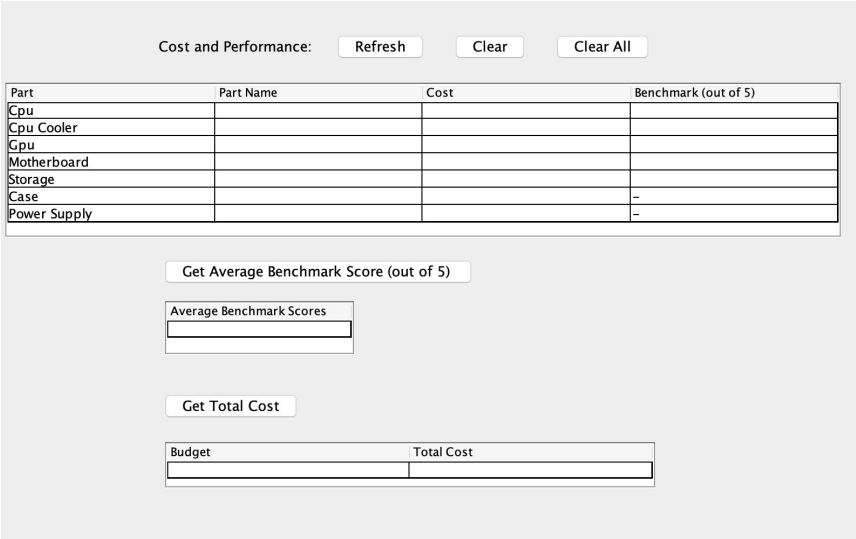
I used Labels and Buttons.



I used Combo Boxes.



I used Tables.



Why:
Having a GUI interface meets my client's expectations of being a really user-friendly program.

Words: 37

Software Tools Used

What:
The programs that were used in this project were both NetBeans and Microsoft Excel. As mentioned above, NetBeans is a highly integrated environment program for Java users and Microsoft Excel is a spreadsheet program developed by Microsoft.

Why:
NetBeans is useful in my project as it is a platform for my client to choose his computer parts and review their costs and performance. Excel serves a purpose in a spreadsheet to review all the chosen parts before purchasing and building the computer.

Words: 81