Final Project Write-up

My project is a User Interface for the periodic table. The Interface allows for a user to click on an element in the Periodic table and receive information on that element, and also allows for a user to search for an element and receive information on that element.

My project has class hierarchy in that there is a parent class called “Element”, which nine subclasses for the different types elements inherit from. Each element object inherits three protected data fields from “Element”: int atomicNumber, String Name, and double molecularWeight. The element objects also inherit get() functions for all of the protected data fields.

There is a class called “Main”. The main() function in Main calls the constructor for Main. When a Main object is created, it first calls the function createElements().

The createElements() function reads from a text file called elements.txt. Each line of this text file has the Atomic Number of an element, the name of that same element, and the molecular weight of that same element. Once a Buffered reader goes through the file line by line, all the Atomic Numbers from the file are added to an Array List called AtomicNumbers, all the element names from the file are added to an Array List called ElementNames, and all the Molecular Weights from the file are added to an Array List called MolecularWeights. All the element objects are then created from the contents of these ArrayLists.

Once all the Elements are created the constructor calls the function createTable(). This function creates JButtons associated with each element and adds them to an ArrayList of JButtons called “Buttons”, and aligns them like the Periodic Table on a JFrame using GridBagLayout. It also creates a title and a search bar (JTextField) at the top of the JFrame, along with a JButton that says “Instructions” next to the JFrame which tells the user how to use the interface when clicked. Once all these components are created, the function calls addSearchListener(JTextField text) which gives function to the search bar. The search bar takes an integer for the Atomic Number of the element that the user is searching for, and then opens up a JFrame which displays the information of that element. The search bar throws an exception if the user enters a number out of range. The function then calls addInstructions() which gives function to the Instructions button, then calls addgb() which aligns the JButtons relative to each other, and ColorButtons() which goes through the ArrayList of buttons and sets their color based on what type of element they’re associated with.

Lastly, after the table is created, the constructor calls a function called addActionListener(). My intent was for this function to walk through the ArrayList of Buttons, have them create a new JFrame that displays information of that Button’s corresponding element. I went about this by creating a for loop that goes through all of the buttons in the ArrayList of Buttons. For each button, I would add an ActionListener and tell it to open up a JFrame and display an element’s information at index “i” in my ArrayList of Elements. However, the problem I ran into was that since I had to enter a new function to implement the ActionListener, I could no longer use the variable “i” that was in my for loop. I tried fixing this by creating a data field integer called “x”, which was initialized to zero just like the for loop, then every time an Action Listener was created for a button to display the information of Elements.get(x), the x variable would increment with the for loop. But when I tried this, it seemed to give every Button the same function in that each button would display the last element’s information in the ArrayList of elements. What I think was happening is that every time the Data Field x was incremented the program would exit the addActionListener() function to change the Data Field, and then go back to the function and start the for loop over and create all the ActionListeners to give the information of element.get(x). What I ended up having to settle for was incrementing x as part of the ActionListener, so every time an element’s button is pressed, it will display the information of next element in the ArrayList of Elements, regardless of which element was pressed.

Overall I was pleased with the project, I feel that even though the final result wasn’t perfect, I learned a lot about creating User Interfaces and became a lot more comfortable with using the Java classes.