

Assignment #6

GUI Payroll Application

Filenames: GUIPayroll.java, Employee.java, Utility.java

This assignment re-uses the Employee class that you created in the previous assignment. You may also want to use the `padString` method from `Utility.java` for formatting some of your output. Hopefully, this assignment will help demonstrate how you can re-use an existing class (such as your `Employee` class) to build a system.

Write a Java GUI application which performs payroll calculations, similar to your previous assignment. The layout of your application is up to you, but you can feel free to mimic the screen shown here. The captured screen shown there uses a 5-row, 2-column `GridLayout` which contains:

- three Label objects
- three TextField objects
- three Button objects
- one TextArea object

The operation of this application should be fairly obvious. Your user is supposed to enter an employee name, hours worked, and a pay rate. When the user clicks the Calculate button, your program should instantiate an `Employee` object, then use methods of the `Employee` class to calculate the employee's pay. The `TextArea` should be cleared, and the employee's information should be shown in the `TextArea`.

The Clear button clears all three `TextFields`, as well as the `TextArea`.

The Quit button terminates the program and closes the window.

1. In Eclipse, create a project named `GUIPayroll`.
2. Create a class named `GUIPayroll`. This class should be a subclass of `java.awt.Frame`. This class will contain your application program logic.
3. The `GUIPayroll` class should contain a main method.
4. The `GUIPayroll` class should also contain a constructor method.

```
public GUIPayroll()  
{  
}
```

5. Your main method should do only one thing: instantiate a GUIPayroll object.

```
new GUIPayroll();
```

6. The GUIPayroll constructor method should:

- set your frame's title
- set your frame's size
- set up your desired layout manager
- instantiate GUI objects and add them to the layout manager
- register your class as the event listener for your buttons
- make your frame visible

Notes:

- To respond to a button click, you need to:

```
import java.awt.event.*;
```

implements ActionListener (in the class header)

override the default actionPerformed method

```
public void actionPerformed(ActionEvent e)  
{  
}
```

- Your logic that actually responds to button clicks (your event responder logic) goes in the actionPerformed method

- To copy Employee.java from a previous project into this one, right-click or control-click on the old Employee.java file, select Copy, then Paste it into your new project.

Note: Because telnet is a text-based program, you will not be able to telnet to your mislab account and run your GUI application. For grading, your teacher will download your files to his computer,

and run it locally. Be sure to upload all required .java files from your project to your mislab account.

Revision #1

Created 2019-01-09 23:36:12 UTC by Aaron Kimbrell

Updated 2019-01-10 16:40:21 UTC by Aaron Kimbrell