Assignment 4: Class Design and GUI

Due Friday September 25, 2009 11:55PM

4.1: Name Class

1. Create a class called `Name`, which will represent a person's name, and do some useful things for us. It will need:
2. 3 string instance variables to store the 3 parts of a person's name: first name, middle name, and last name. They should have the appropriate visibility / protection.
3. A constructor that takes in 3 strings corresponding to the 3 parts of a person's name.
4. 3 getters for the instance variables, following standard Java getter/setter naming conventions.
5. A method named "`normalize`" that will change the names to standard capitalization. For example, if the original name is "jOHn DOe SmiTH", after "`normalize`" is called, it should be "John Doe Smith".
6. A method called "`initials`" that will return the name's initials. For example, if the name is "John Doe Smith", the method should return "JDS".
7. An enumerated type called `NameFormat` that will represent the different ways a full name can be formed. This can be a public enum declared in the `Name` class, or declared in a separate file by itself. It should have the following members:

a.	`firstMiddleLast`	example: "John Doe Smith"
b.	`lastFirstMiddle`	example: "Smith, John Doe"
c.	`firstInitialLast`	example: "John D. Smith"
d.	`initialInitialLast`	example: "J. D. Smith"

8. A method called "`getFullName`" that takes a `NameFormat` parameter. Based on that parameter, it should return the name represented by the `Name` object according to the examples above.
9. A method called "`length`" that returns the number of characters in the full name, not including spaces.

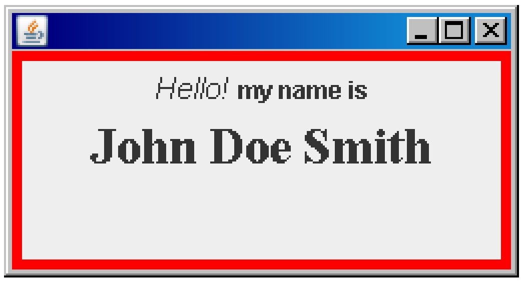
Don't forget to javadoc the class, constructor, and methods, javadocing parameters where applicable (see the previous homework for an example of this)

4.2: NameTag Class

Create a class called `NameTag` that creates a `JFrame` and `JLabels` based upon a `Name` object. It should have:

1. An instance variable to store the `Name`, and an instance variable to store the `JFrame`. They should have the appropriate visibility / protection.
2. A constructor that takes a `Name` parameter for the `NameTag` object to show.
3. In the constructor, create a `JFrame` and store it into the instance variable from step 1. Add a `JPanel`, and put `JLabels` into the `JPanel` that display the name from the `Name` object passed to the constructor. It should use the "`getFullName`" method from part 1, step 7. Don't make the `JFrame` visible here. Don't forget to set the default close operation on the `JFrame`.

   An example would be:

4. Hint: if you declared the enum `NameFormat` in the `Name` class, you can access the enum like this:
   ```
   Name.NameFormat format = Name.NameFormat.firstMiddleLast;
   ```

5. You may use `Font` and `Border` objects to achieve the affects in the above example:

   ```
   label.setFont(new Font("Arial", Font.ITALIC, 15));

   label2.setFont(new Font("Times New Roman", Font.BOLD, 25));

   panel.setBorder(BorderFactory.createLineBorder(Color.red, 5));
   ```

   See: http://java.sun.com/j2se/1.4.2/docs/api/java/awt/Font.html

6. Lastly, it should have a method named "`showNameTag`" that will actually show the `JFrame`.

Don't forget to javadoc the class, constructor, and methods, javadocing parameters where applicable (see the previous homework for an example of this)

4.3: NameTagGenerator

Finally, create a class named NameTagGenerator with a main method that brings together a `Scanner` object, a `Name` object, and a `NameTag` object:

1. Using a `Scanner`, prompt the user for his or her first, middle, and last names.
2. Create a `Name` object, using the values collected above.
3. Normalize the name.
4. Create a `NameTag` object, passing the above `Name` object.
5. Show the name tag.

Turn-in Procedure

Turn in the following files on T-Square. Double-check that you have *submitted* and not just saved as draft (if the submission is successful, you will receive an e-mail from T-Square within a few minutes). Also, be sure all of your files compile and run.

- Name.java
- NameTag.java
- NameTagGenerator.java
- Any other files needed to run your submission

All .java files should have a brief descriptive javadoc comment.

Don't forget your collaboration statement. You should include a statement with every homework you submit, even if you worked alone.