3. LAB C – PROTOTYPE

Purpose

This lab will focus on the prototype design pattern. Students will be required to develop a GUIFactory object which creates objects by cloning a prototype.

1. The GUIFactory will store UI Components.
2. On client request, the factory will clone an existing UI Component and return it.

Lab Scenario (WHAT to do)	Develop a GUIFactory class, which wraps existing UI Components via the constructor, and clones them. **The class has this API** : GUIFactory (Button, ComboBox) createButton() createComboBox() The create methods will duplicate the UI Components stored in the GUIFactory (sent to the constructor on initialization)
Implementation Steps (HOW to do it)	**Code a ComboBox interface** You can add a public Object clone() method, but it is not required. The method should appear on the implementation, not the interface. **Code the Button interface** Again. you can add a public Object clone() method, but it is not required. The method should appear on the implementation, not the interface. **Code the GUIFactory class** - Add a Button member variable and a ComboBox member variable - Implement a constructor taking those values and setting them into the member variables - Implement a createButton method The method will return a clone for the Button member variable. Consider reflection API for this. - Implement a createComboBox method The method will return a clone for the ComboBox member variable. Consider reflection API for this.

Review of Prototype

Three classes will be created :

1. Button interface will serve as base for Concrete Button objects
2. ComboBox interface will serve as base for concrete ComboBox objects
3. GUIFactory will implement the prototype design pattern.

3.1.1. Create a new interface Button

Serves as base for concrete Buttons.:

1. Create the class Button
2. Possibly, implement a public void clone() method

3.1.2. Create a new interface ComboBox

Serves as base for concrete ComboBoxes

1. Create the class ComboBox
2. Possibly, implement a public void clone() method

3.1.3. Create a new class GUIFactory

This class will implement the prototype design pattern. On construction, clients can send UI components to the class (via the class constructor) and then, when asking it for a new UI components , the GUIFactory will respond by cloning an existing object and returning it to the client.

1. Create the class GUIFactory
2. Implement a Button and ComboBox member variables

```
private Button button;
private ComboBox combo;
```

3. Implement the constructor

```
public GUIFactory(Button button, ComboBox combo) {
  this.button = button;
  this.combo = combo;
}
```

4. Implement a createButton() method

```
public Button createButton() {
   return (Button) button.clone();
}
```

5. Implement a createComboBox() method

```
public ComboBox createComboBox() {
   return (ComboBox) combo.clone();
}
```

Note: The solution for this exercise is available in the ‘solutions’ directory