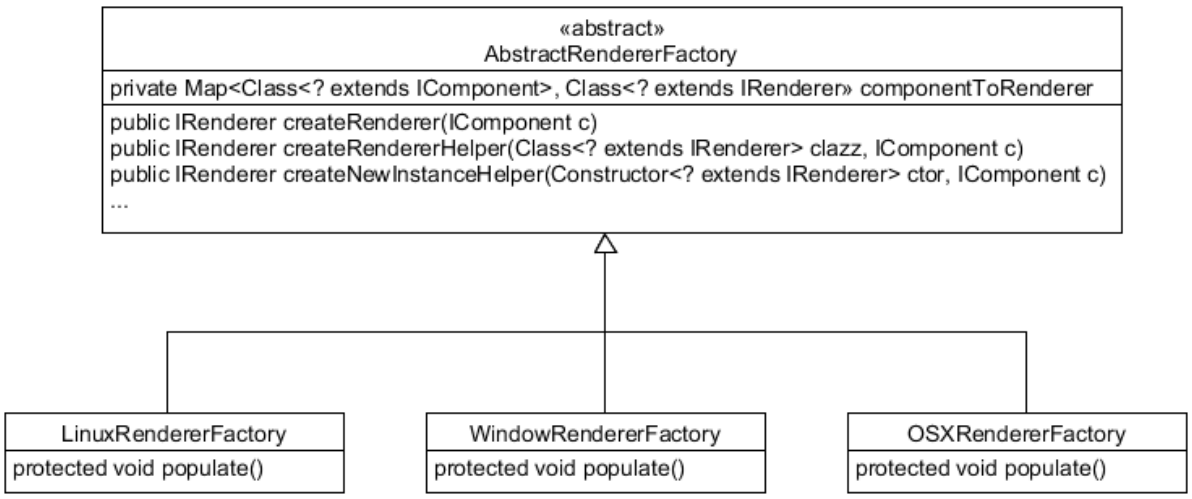
1.

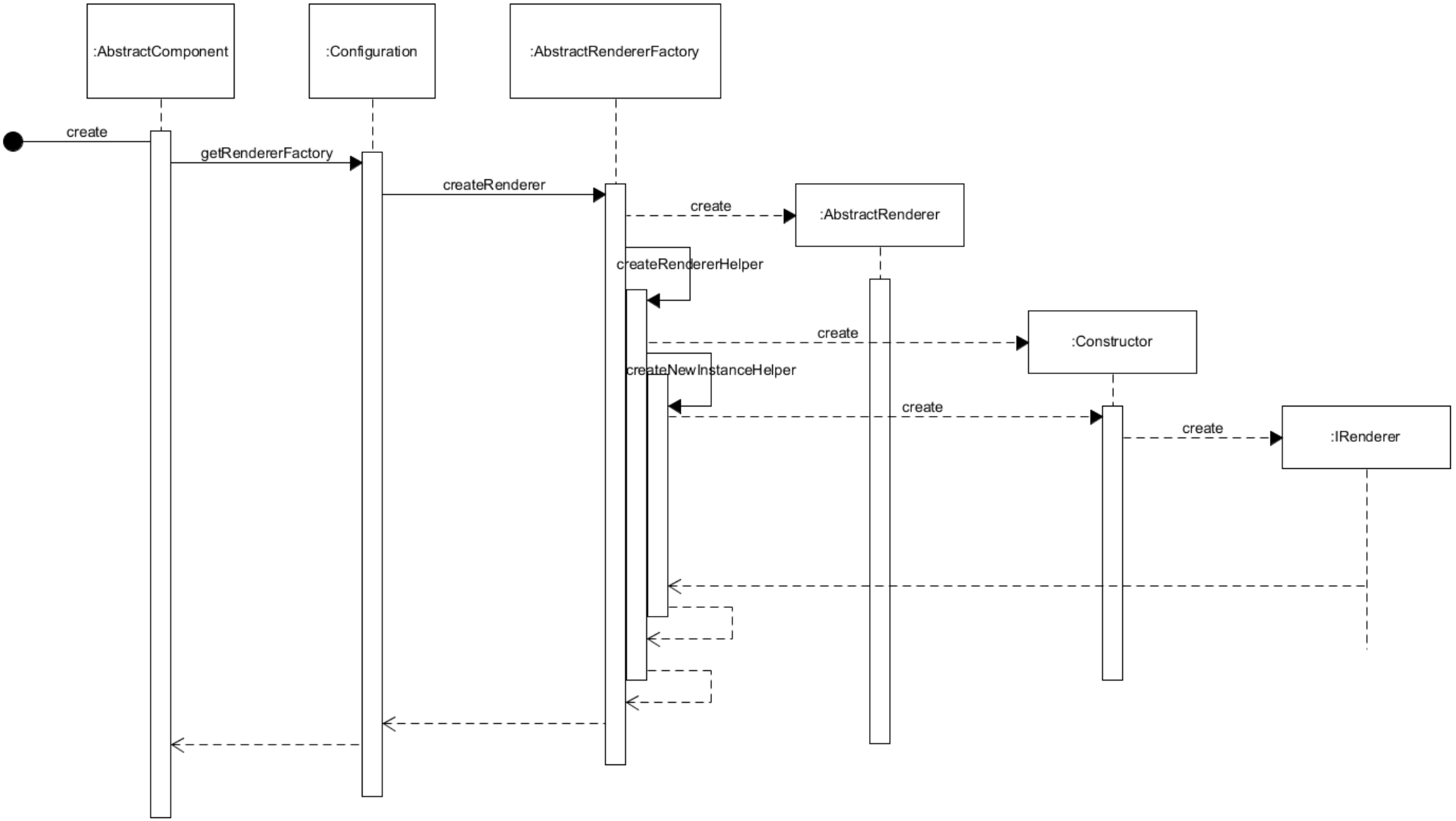
In the AbstractComponent class's public AbstractComponent(IComponent parent, Rectangle bound) constructor on line 40, there is a violation.
The AbstractComponent object's renderer is created from relying on two classes. Instead, you could call a static Configuration.createRenderer(CComponent c) method that returns an IRenderer based on the given IComponent.

In the AbstractRendererFactory class's public IRenderer createRenderer(IComponent c) method on lines 28 and 29 there is a violation.
A concrete implementation of an IRenderer object is created on line 27 before method calls originating from this object occur. To fix the violations, you could create a public IRenderer createRendererHelper(Class<? extends IRenderer> clazz, IComponent c) method inside of the AbstractRendererFactory class. This method would call getDeclaredConstructor(IComponent.class) on the given Class<? extends IRenderer> object storing the returned Constructor<? extends IRenderer> object as a local variable. Another helper method, public IRenderer createNewInstanceHelper(Constructor<? extends IRenderer> ctor, IComponent c) would then be called in the createRendererHelper method by passing the Constructor and IComponent objects. The createNewInstanceHelper method would then return an IRenderer call newInstance on the given Constructor object while also passing the IComponent object as the parameter for the newInstance call. The instance would then get returned. It would look something like this: return ctor.newInstance(c).

2.
Diagrams for violation 2:



Since the AbstractRendererFactory's createRenderer method was changed the only classes affected by the new helper methods are the concrete RendererFactories which are the LinuxRendererFactory, WindowRendererFactory, and OSXRendererFactory.

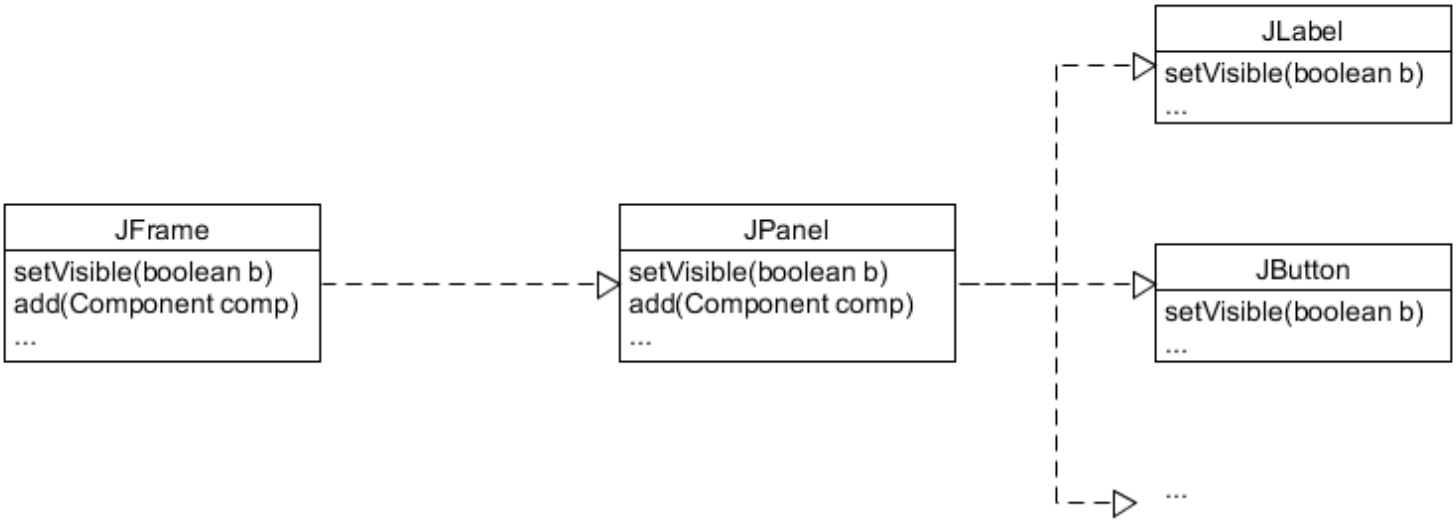


The only class that calls the createRenderer method is the AbstractComponent class. Once called, the AbstractRendererFactory creates a new AbstractRenderer instance before calling the createRendererHelper method. This helper method creates a Constructor from the AbstractRenderer before calling the createNewInstanceHelper method. This method creates a new instance of an IRenderer object using the Constructor and the initial IComponent. This IRenderer object is then returned to the AbstractComponent class.

3.

The javax.swing library is a GUI toolkit that provides many GUI components for developers to use in their implementation of GUIs for their Java applications.

4.



The façade class in this case is the JPanel class. Components can be added to the JPanel by calling the add method. These Components can then be made visible by calling JPanel's setVisible method with true. The façade uses the Components to make them visible on the JFrame after the façade itself is added to the JFrame. The JFrame acts as the client and after adding the JPanel to itself, it calls setVisible with true as the argument. This passes a true Boolean to setVisible on the JPanel which also passes the Boolean to setVisible on the Components. The end result is a JFrame with the JPanel's Components visible to the end user.