**Gof Design Patterns** Nádia Mendes 53175

1. **template method pattern:** package net.sf.freecol.client.gui.action.FreeColAction

In this exemple we can see that FreeColAction serves as an abstract base class that defines a skeleton of behavior for various actions. It implements common methods and defines a shouldBeEnabled method as a hook that is overridden by its subclasses to provide specific behaviors, as is the case with the ChatAction and DebugAction classes.

In the ChatAction class, which is a subclass of the FreeColAction class, this is exactly what happens, it overrides the shouldBeEnabled method to provide specific logic for the chat action.

Uma imagem com texto, captura de ecrã, Tipo de letra, software

Descrição gerada automaticamente

**Class Diagram –**

**Uma imagem com texto, diagrama, file, Tipo de letra

Descrição gerada automaticamente**

1. **Observer pattern:** net.sf.freecol.client.gui.dialog.CaptureGoodsDialog

In this CaptureGoodsDialog class we check the existence of an observer pattern, where the subject of the Observer pattern is the goodsList, which is an instance of JList<GoodsItem>. However, there is no explicit interface called Observer or Observable, because in context the interaction between the subject (such as the goodsList) and the observers (or "listeners") is handled through specific methods and interfaces provided by Java Swing itself.

The interaction between goodsList and observers is carried out through the addMouseListener(MouseListener listener) and removeMouseListener(MouseListener listener) methods. These methods add or remove specific observers that implement the MouseListener interface. The goodsList notifies these observers when mouse events occur.

Therefore, the goodsList acts as the subject and the observers are classes that implement the MouseListener interface.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

**Class Diagram –**

**Uma imagem com texto, diagrama, captura de ecrã, file

Descrição gerada automaticamente**

**3-Abstract Factory Pattern:** src.net.sf.freecol.client.gui.option.LanguageOptionUi

In the LanguageOptionUI class, contained in the src.net.sf.freecol.client.gui.option package, it appears that it acts as an abstract factory to create objects related to the language option (LanguageOption)

The LanguageOptionUI class creates and returns an instance of JComboBox <Language> which is a part of the language option-related family of UI objects.

The LanguageOption class represents the language option, while the Language class represents the available languages.

Therefore, the use of the Abstract Factory Pattern is used to create related objects according to the language choice, and this allows the creation of a family of coherent objects related to the language choice.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, software

Descrição gerada automaticamente

Representation of code of the *LangugeOptionUI* class.

**Class Diagram –**

Uma imagem com texto, captura de ecrã, recibo, Tipo de letra

Descrição gerada automaticamente