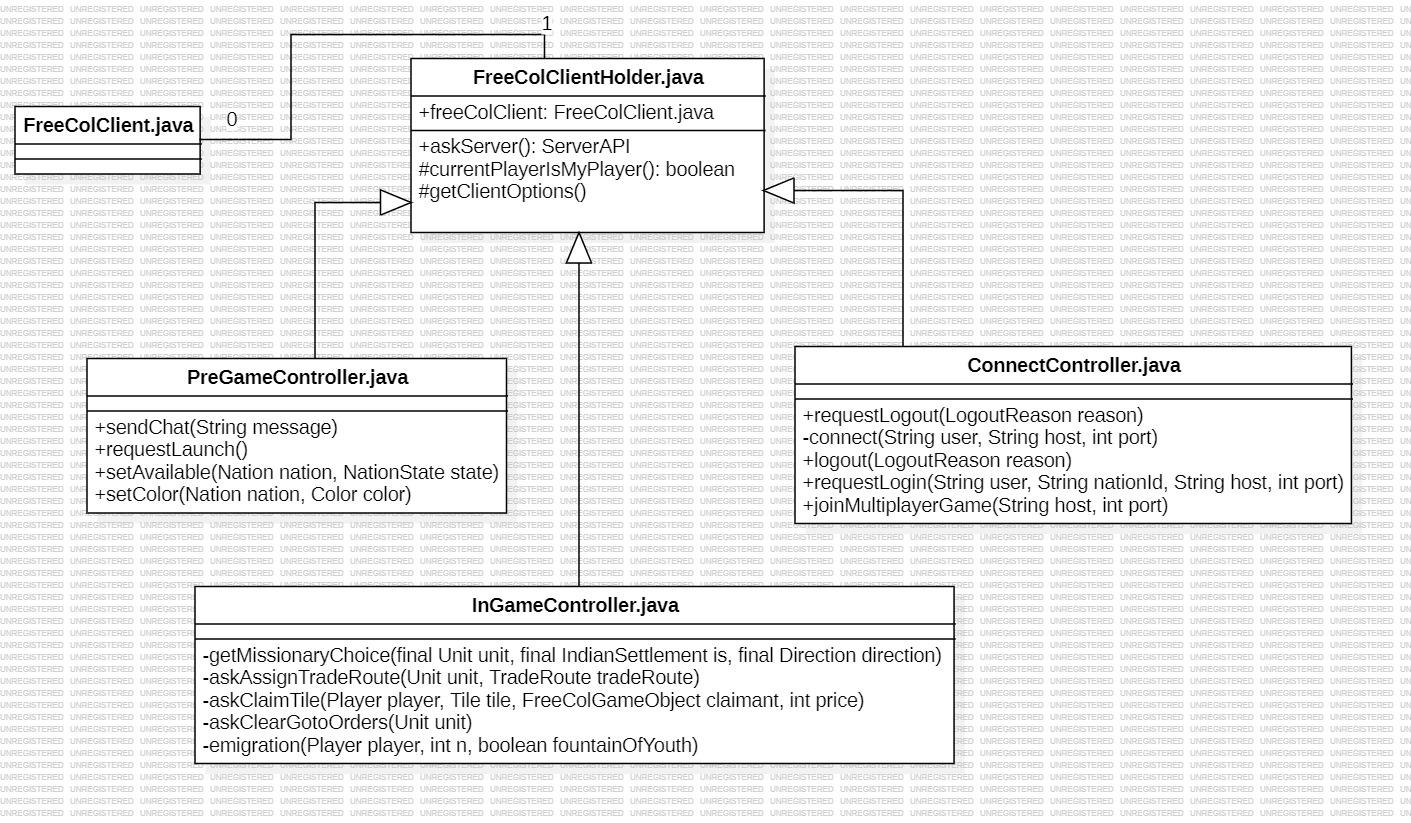
DESIGN PATTERNS

3 patterns identificados durante a visualização do código:

FreeColClientHolder.java - Proxy Pattern

Explicação: FreecolClientHolder é um Proxy pois simplifica a complexidade da classe FreecolClient usando outra classe para manipular os métodos.



Código:

public class FreeColClientHolder {

*/\*\* The main client object. \*/*

private final FreeColClient freeColClient;

*/\*\**

*\* Simple constructor.*

*\**

*\* @param freeColClient The {@code FreeColClient} to hold.*

*\*/*

protected FreeColClientHolder(FreeColClient freeColClient) {

this.freeColClient = freeColClient;

}

*/\*\**

*\* Meaningfully named access to the server API.*

*\**

*\* @return The {@code ServerAPI}.*

*\*/*

public ServerAPI askServer() {

return this.freeColClient.askServer();

}

*/\*\**

*\* Check if the current player is the client player.*

*\**

*\* @return True if the client player is current.*

*\*/*

protected boolean currentPlayerIsMyPlayer() {

return this.freeColClient.currentPlayerIsMyPlayer();

}

*/\*\**

*\* Get the client options.*

*\**

*\* @return The {@code ClientOptions} held by the client.*

*\*/*

protected ClientOptions getClientOptions() {

return this.freeColClient.getClientOptions();

}

*/\*\**

*\* Get the connect controller.*

*\**

*\* @return The {@code ConnectController} held by the client.*

*\*/*

protected ConnectController getConnectController() {

return this.freeColClient.getConnectController();

}

*/\*\**

*\* Get the main client object.*

*\**

*\* @return The {@code FreeColClient} held by this object.*

*\*/*

protected FreeColClient getFreeColClient() {

return this.freeColClient;

}

*/\*\**

*\* Get the server.*

*\**

*\* @return The {@code FreeColServer} held by the client.*

*\*/*

protected FreeColServer getFreeColServer() {

return this.freeColClient.getFreeColServer();

}

*/\*\**

*\* Get the game.*

*\**

*\* @return The {@code Game} held by the client.*

*\*/*

protected Game getGame() {

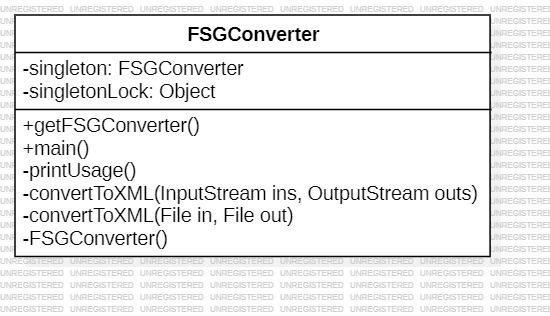
return this.freeColClient.getGame();

}

. . .

FSGConverter.java - Singleton Pattern

Explicação: FSGConverter é um singleton pattern pois, tem seu construtor privado e um método, que só permite a criação de apenas uma instância do objeto dessa classe.



Código:

public class FSGConverter {

*/\*\**

*\* A singleton object of this class.*

*\* @see #getFSGConverter()*

*\*/*

private static FSGConverter *singleton*;

private static Object *singletonLock* = new Object();

*/\*\**

*\* Creates an instance of {@code FSGConverter}*

*\*/*

private FSGConverter() {

// Nothing to initialize;

}

*/\*\**

*\* Gets an object for converting FreeCol Savegames.*

*\* @return The singleton object.*

*\*/*

public static FSGConverter getFSGConverter() {

// Using lazy initialization:

synchronized (*singletonLock*) {

if (*singleton* == null) {

*singleton* = new FSGConverter();

}

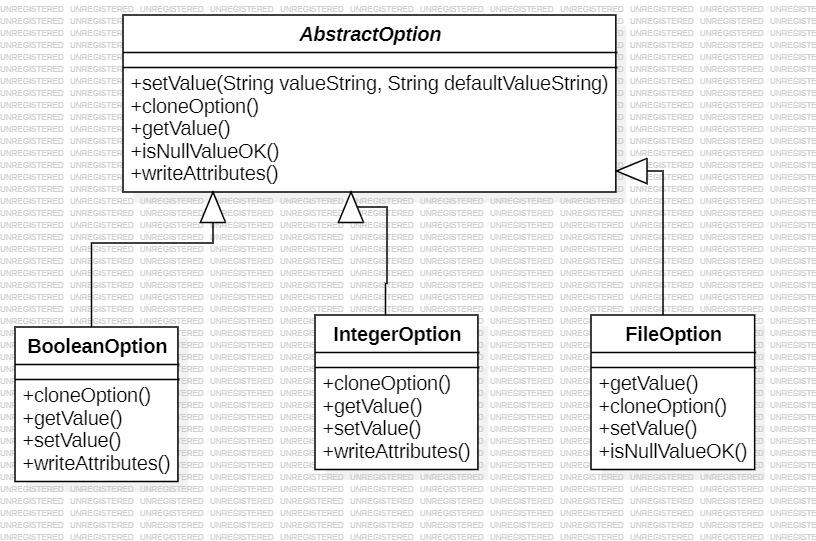
return *singleton*;

}

}

AbstractOption.java -Template method Pattern

Explicação: o AbstractOption define as etapas para executar um algoritmo e fornece implementação padrão que pode ser comum para todas ou algumas das subclasses (BooleanOption, ItegerOption, FileOption,...).



Código:

public abstract class AbstractOption<T> extends FreeColSpecObject

implements Option<T> {

private static final Logger *logger* = Logger.*getLogger*(AbstractOption.class.getName());

private static final String *ENABLED\_BY\_TAG* = "enabledBy";

*/\*\* The option group prefix. \*/*

private String optionGroupId = "";

*/\*\**

*\* Determine if the option has been defined. When defined an*

*\* option won't change when a default value is read from an XML file.*

*\*/*

protected boolean isDefined = false;

private String enabledBy = null;

public boolean isNullValueOK() {

return false;

}

. . .

// Interface Option

*/\*\**

*\* {@inheritDoc}*

*\*/*

@Override

public abstract AbstractOption<T> cloneOption()

throws CloneNotSupportedException;

*/\*\**

*\* {@inheritDoc}*

*\*/*

public String getGroup() {

return this.optionGroupId;

}

*/\*\**

*\* {@inheritDoc}*

*\*/*

public void setGroup(String group) {

this.optionGroupId = (group == null) ? "" : group;

}

*/\*\**

*\* {@inheritDoc}*

*\*/*

@Override

public abstract T getValue();

*/\*\**

*\* {@inheritDoc}*

*\*/*

@Override

public abstract void setValue(T value);

*/\*\**

*\* {@inheritDoc}*

*\*/*

@Override

public String getEnabledBy() {

return enabledBy;

}