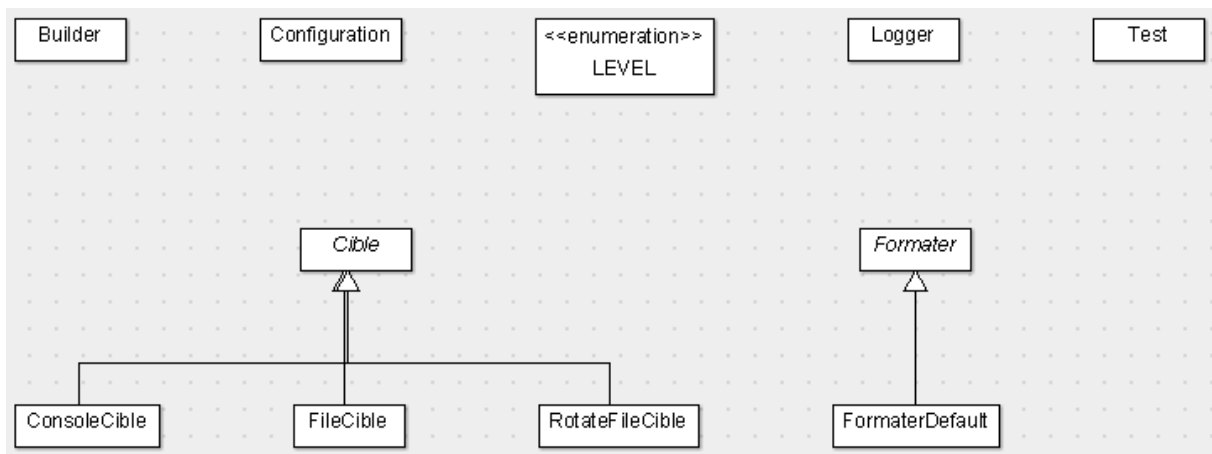
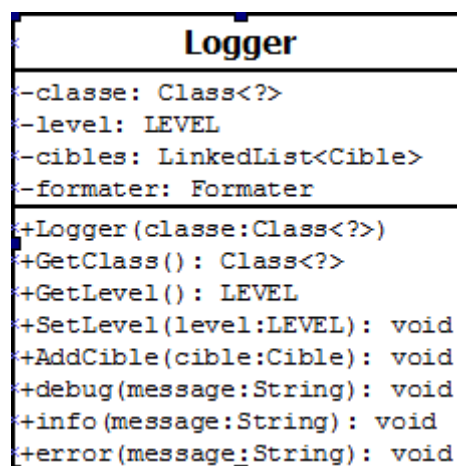


Projet 4A
Design Objet
Framework de logging

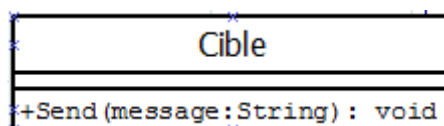
Architecture du Framework



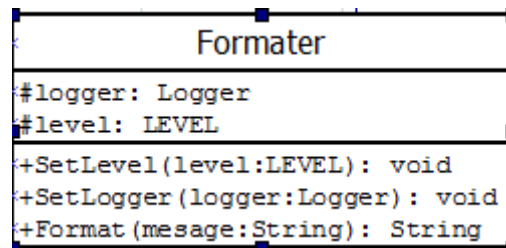
- L'objet **Logger** permet d'envoyer des messages, via les méthodes **debug**, **info** et **error**. Ces messages sont envoyés à chacune des cibles de la liste : **cibles**, en fonction du niveau **level** et de la méthode utilisée.



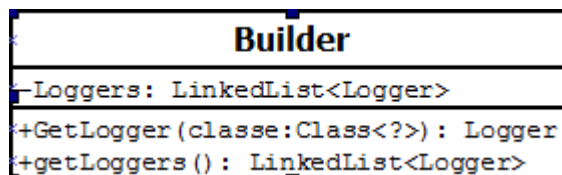
- Un objet issu de la méthode abstraite **Cible** permet d'envoyé des informations à l'utilisateur d'une certaine manière. De nouvelles cibles peuvent ainsi être ajouté par l'utilsateur sans modifier le code.



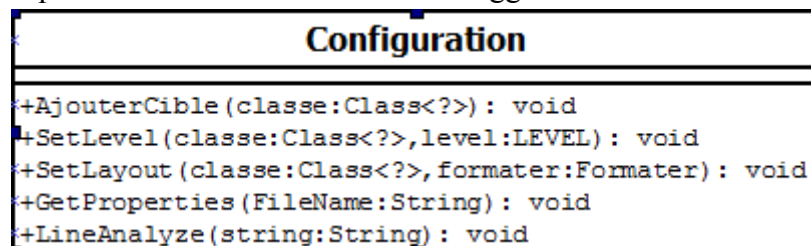
- Un objet issu de la méthode abstraite **Formater** permet de mettre en forme un message via la méthode **Format**. De nouveaux formats peuvent ainsi être ajouté par l'utilsateur sans modifier le code.



- La classe **Builder** contient la liste de tous les **logger** dans une liste : **Loggers**. La méthode **GetLogger** permet à l'utilisateur de trouver ou créer un logger dans la liste avec une classe donnée.



- La classe **Configuration** permet à l'utilisateur de paramétrer ses loggers avec une classe donnée. La méthode **AjouterCible** permet d'ajouter une cible à la liste des cibles du logger, la méthode **SetLevel** et **SetLayout** permettent d'initialiser les attributs level et formater du logger. La méthode **GetProperties** permet d'analyser un fichier properties.txt pour effectuer l'initialisation des loggers.

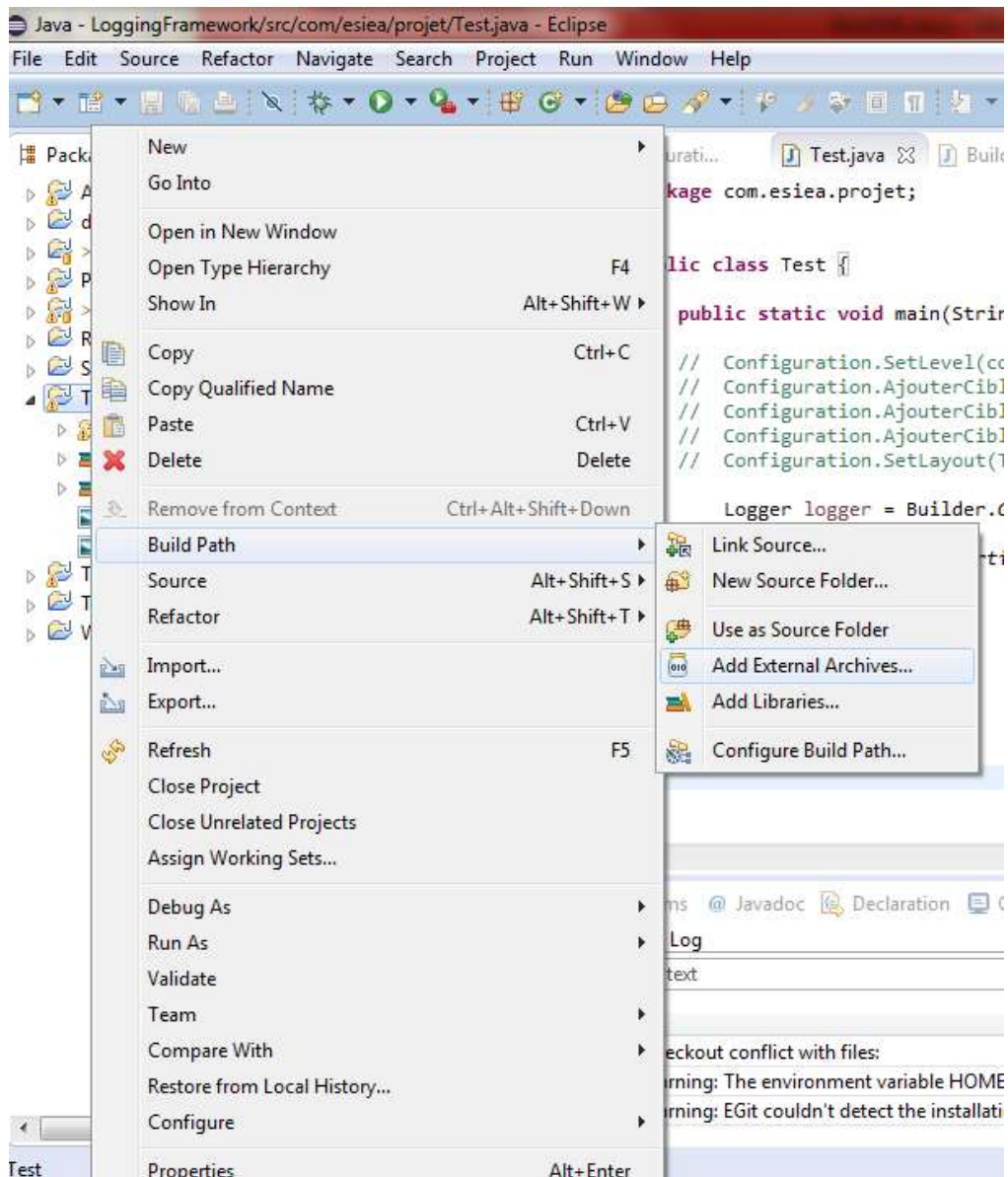


Répartition des tâches :

- Dora : Cible, ConsoleCible, FileCilbe, RotateFileCible, Test
- Marine : Configuration, Formater, FormaterDefault, Test
- Romain : LEVEL, Logger, builder, Test

Installation

Pour installer le framework, il suffit de télécharger l'archive .zip sur GitHub et de la dézipper dans le dossier de votre choix. Sous Eclipse, faire un clic droit sur le projet souhaité, puis Build Path, Add External Archives, puis aller chercher dans le dossier LoggingFramework-Master, le fichier Logging.jar. Pour utiliser le framework, il faut y importer le package via la ligne : `import com.esiea.projet.ClasseConcernée.`



Utilisation

Créer un logger lié à la classe MaClasse :

```
Logger logger = Builder. GetLogger(MaClasse.class)
```

Configurer un logger lié à la classe MaClasse :

- Initialiser le niveau level :
 - o Configuration.SetLevel(MaClasse.class, LEVEL.level);
- Ajouter la Cible cible :
 - o Configuration.AjouterCible(MaClasse.class, cible) ;
- Initialiser le Formater formater
 - o Configuration.SetLayout(MaClasse.class, formater);
- Configuration à partir d'un fichier properties.txt
 - o Configuration.GetPropertiesFile("Properties.txt");

Fichier Properties.txt :

Format d'une ligne de commande dans le fichier Properties.txt. Initialisation de l'**Attribut** de **MaClasse** avec l'**Objet** et les **Argument1** et **Argument2** de son constructeur si besoin.

```
logger.MaClasse.Attribut = Objet Argument1 Argument2
```

Exemple :

```
logger.com.esiea.projet.Test.level = DEBUG
logger.com.esiea.projet.Test.formater = com.esiea.projet.FormaterDefault
logger.com.esiea.projet.Test.cibles = com.esiea.projet.ConsoleCible
logger.com.esiea.projet.Test.cibles = com.esiea.projet.FileCible log.txt
logger.com.esiea.projet.Test.cibles = com.esiea.projet.RotateFileCible rlog.txt 10
```