

Consignes générales

- Travail effectué par groupe de 3. Chaque groupe choisit un thème. Pour chaque thème, les explications complémentaires nécessaires pour la réalisation du travail vous seront fournies.
- Utilisation d'un serveur d'intégration
- Utilisation d'au moins une technique de tests en boîte noire et une technique de tests en boîte blanche (vue au cours ou autre)
- Remise d'un rapport explicitant les objectifs fixés pour les tests, la ou les méthodologie(s) utilisée(s) ainsi que les résultats obtenus.
- Prises d'initiatives requises pour la réalisation du projet en respectant toutefois les quelques contraintes imposées !

Thème 1 : Elaboration d'un Framework pour développement de jeux de plateaux

Une société qui édite des logiciels souhaiterait écrire un framework pour écrire des jeux de plateaux (Les échecs, les dames, othello, ...). Vous êtes en charge de l'élaboration de la solution. Deux contraintes de départ vous ont été formulées.

- Le framework doit permettre de créer facilement les interfaces graphiques pour de nouveaux jeux.
- Certains jeux peuvent être proposés en version distante (au travers du réseau).

La solution proposée fournira un guide comprenant les actions et les tâches à réaliser pour développer un nouveau jeu (par exemple création de classes, modification de fichiers de configuration, ...)

Lien(s) utile(s)

- Utilisation des sockets en Java
<https://docs.oracle.com/javase/tutorial/networking/sockets/index.html>

Thème 2 : Création d'un outil d'analyse de code

Ecrire une application permettant de détecter des classes qualifiées de "God class". Une telle classe a trop de responsabilités et accède généralement directement aux données d'autres classes. La réutilisabilité et la facilité de compréhension de cette classe n'est dès lors pas facile. Une bonne pratique de programmation orientée objet veut que l'on attribue au plus une seule responsabilité à une classe. Les métriques utilisés pour détecter une "God class" sont (cf. ouvrage Object-oriented Metrics in Practice de Stéphane Ducasse et Michèle Lanza)

- Mesure de la complexité de la classe : Weighted Method Count (WMC)
<http://www.aivosto.com/project/help/pm-oo-ck.html>
 - La nombre de méthodes dans une classe ou

- La somme des complexités statiques de toutes les méthodes d'une classe. Dans ce dernier cas, la complexité cyclomatique de McCabe peut être utilisée comme mesure de la complexité.
- Mesure de la cohésion d'une classe : Tight Class Cohesion (TCC) : TCC)

<http://www.aivosto.com/project/help/pm-oo-cohesion.html>

Le métrique sert à évaluer la cohésion d'une classe. Sa valeur est située en 0 et 1. Il correspond au nombre relatif de paires de méthodes d'une classe qui accède au moins à un même attribut de cette classe. En dessous de 1/3, on considérera que la classe a peu de cohésion. Votre solution doit permettre l'utilisation d'un autre métrique de cohésion.

- Indicateur du respect de l'encapsulation des autres classes : Access to Foreign Data (ATFD) : représente le nombre de classes extérieures auxquelles la classe accède les attributs (de manière directe ou par les accesseurs). Pas plus que quelques-uns (à définir) comme limite.

Les classes que l'on cherche à détecter dans un projet pour lequel on se propose d'étudier le code sont donc celles qui sont trop complexes, avec peu de cohésion et qui accèdent directement aux variables membres des autres classes. Remarque : cela ne signifie pas que la classe détectée est mal conçue mais simplement qu'il y a là un indicateur qui nous invite à pousser la réflexion quant à la conception du code.

Lien(s) utile(s)

- un parseur pour le code Java : <https://code.google.com/archive/p/javaparser/wikis/UsingThisParser.wiki>

Thème 3: Elaboration d'un « tiny » framework web

Ecrire un framework web permettant de configurer à l'aide d'un fichier de configuration des actions à réaliser et les pages à afficher si les actions correspondantes ont été réalisées avec succès.

Exemple

```
<?xml version = "1.0" encoding = "UTF-8" standalone = "yes"? >
<workflow>
< stateaction = " EmptyAct ion" name=" empty "viewURI ="/Login.jsp" / >
< stateaction = " Log inAction" name=" login "viewURI ="/ Language. html" / >
< stateaction = " Languag eAction" name=" language " viewURI ="/Login.jsp" / >
</workflow>
```

Une requête à l'adresse <http://monserver/applic/Workflow/Empty.act> signifiera que l'on utilise le contrôleur Workflow et que l'on désire réaliser l'action EmptyAction. Selon le fichier de configuration, si l'action réussit, la vue à utiliser est Login.jsp. Notre exemple illustre un contrôleur fonctionnant selon le principe d'un workflow simple (suite d'actions et absence de branchements - alternatives)

Concrètement, un programmeur utilisant le framework devra effectuer les tâches suivantes :

1. Modification du fichier de mapping action-vue.
2. Création d'une classe xxxAction implémentant l'interface Action.

3. Création éventuelle d'une vue (statique ou dynamique).

Lien(s) utile(s)

- Patterns JEE : <http://corej2eepatterns.com/index.htm>
- Pour passer de l'XML vers des objets en Java (unmarshalling), JAXB : <https://jaxb.java.net/>

Thème 4 : Chiffrement par blocks ECB et CBC

Ecrire une application illustrant le chiffrement par blocks ECB et CBC en Erlang.

Le langage Erlang : Erlang est un langage créé par l'industrie (Ericsson) pour répondre à un besoin particulier, à savoir un langage permettant de construire des systèmes fiables et tolérants aux fautes. Les concepts qui sont implémentés dans le langage en font clairement sa particularité et son originalité. Son utilisation n'a cessé de croître et il est possible aujourd'hui de programmer en Erlang des web services ! Pour des exemples d'applications développées en Erlang :

- <http://erlang.org/faq/introduction.html>. L'exemple récent le plus connu est l'application [Whatsapp](#)

Lien(s) utile(s)

- Modes ECB et CBC https://fr.wikipedia.org/wiki/Mode_d'op%C3%A9ration_%28cryptographie%29
- Cours sur Erlang <http://www.erlang.org/course/course.html>.
- Bit syntax http://www.erlang.org/doc/programming_examples/bit_syntax.html
- Higher order functions <http://www.erlang.org/documentation/doc-5.2/doc/extensions/funs.html>.

Thème 5 : Création d'un utilitaire de sélection de projets basé sur les critères de la NPV et de l'IRR

L'utilitaire permettra les fonctionnalités suivantes

- Ajout de projets ;
- Ajout des cashs flow et de l'échéancier pour un projet ;
- Calcul de la NPV d'un projet ;
- Calcul de l'IRR d'un projet (selon les cashs flow et l'échéancier encodé au préalable) par une méthode itérative de recherche de zéros(méthode au choix)
- Comparaison de deux projets (taux paramétrables)

Lien(s) utile(s) :

- Notion de NPV : https://en.wikipedia.org/wiki/Net_present_value
- IRR <http://www.investopedia.com/terms/i/irr.asp>