

Génie Logiciel et Projet – TD 4 – Conception (revue) du noyau fonctionnel

1. Conception du moteur de projet

Pour les classes de « **traitement** » de votre projet, il s'agit des algorithmes et des déroulements importants du logiciel à développer. L'ensemble des classes de traitement et des classes de données constitue le **noyau fonctionnel (moteur)** de votre projet.

L'objectif de ce TD est ainsi de dresser l'état des lieux de votre conception / programmation. Vous prendrez en considération tous les déroulements / fonctionnements importants du projet. **Ne pas les concevoir sous-forme de diagramme de classe UML**, car dans ce genre de diagramme statique, on ne voit pas comment les méthodes sont implémentées.

Vous pouvez faire **des schémas simples** pour illustrer votre conception. Vous pouvez aussi utiliser le format du pseudo-code pour décrire votre solution de conception du moteur. Les mots clés réservés pour le **pseudo-code** :

ALGORITHME	PROCEDURE	CONSTANTES	VARIABLES	DEBUT	FIN	FONCTION	SI	ALORS
SINON	POUR	TANT_QUE	JUSQU'A	REPETER	SELON	AUTREMENT	BOOLEEN	ENTIER
REEL	CARACTERE	CHAINE	NON	OU	ET	MOD	DIV	PARAMETRES

Les déroulements / fonctionnements algorithmiques de votre logiciel nécessitent toujours des traitements des données : création, lecture, mise à jour et suppression (*CRUD : Create, Read, Update and Delete*). Ainsi, le travail de conception du noyau vous permettra de vérifier si votre modèle de données conçu auparavant convient aux fonctionnements. Adaptez / ajustez votre modèle de données en vous basant sur les besoins des fonctionnements. **Le fonctionnement de votre logiciel est souvent modélisé par une ou plusieurs procédures de déroulement qui est (sont) divisée(s) en différentes étapes (séquentielles ou itératives).**

Le moteur de projet doit fournir un ensemble de fonctions (classes + méthodes) utilisées par la partie IHM. L'IHM doit pouvoir être prévue lorsque le moteur change.

Attention : le projet GLP ne demande pas de mode console formel : donc l'IHM graphique à réaliser obligatoirement.

2. Documentation de votre solution

Le principe reste le même : c'est souvent par écrit que l'on rend les choses plus claires. Documentez votre conception régulièrement. Profitez bien de toutes les formes de présentation possibles : schéma, pseudo-code, tableau, liste, texte, etc. Votre conception doit être concise et cohérente. Son implémentation doit être un travail direct sans ambiguïté.

Note : Quand vous pouvez (ex. après le premier point d'avancement), vous mettrez sur Discord dimanche prochain, **le premier jet (probablement incomplet)** de cette conception qui aura besoin d'être enrichi et complété tout au long de votre projet. Pensez aussi à bien gérer l'équilibre entre le travail de programmation et celui de documentation pour avoir un rythme d'avancement correct.

Les différentes conceptions du projet vous permettent de mieux répartir les tâches. Créez et utilisez au plus vite GitHub pour travailler plus facilement en programmation d'équipe, en suivant les guides comme par exemple :

<https://www.youtube.com/watch?v=LPT7v69guVY&t=98s>

L'utilisation du Git n'est pas obligatoire.