

Programmation Orientée Projet

Méthodes de travail pour le développement du projet

6. Exercices (solution)

- Questions avancées qui pourront être traitées en semaine 2-3: le présent document a surtout souligné que l'interface exportait des fonctions. Cependant il est aussi possible d'exporter des modules de structure (le cours précisera qu'on parle alors d'un type *concret*).
 - Supposons qu'une personne est responsable d'un module C dont l'interface exporte des fonctions et **les modèles des structures** manipulées par ces fonctions (comme paramètre ou résultat).
 - Le module C est-il fiable ? En d'autres termes, la personne responsable de ce module peut-elle garantir que les fonctions exportées dans l'interface fonctionnent toujours correctement ?
 - *NON car tout module qui utilise ce module dispose d'un accès direct aux champs d'une telle structure sans devoir passer par une fonction. Il est donc possible d'écrire des instructions qui affectent des valeurs incorrectes à ces champs, en dehors du contrôle de la personne responsable du module C.*
 - *Cela dit il est n'est pas totalement interdit de mettre en œuvre ce type de module (type concret) mais c'est à réserver à des types de très bas niveau, genre **bibliothèque utilitaire très stable et très robuste** à ce type de liberté laissée en dehors du module C.*
 - Supposons que la personne responsable du module C décide de changer le type un champ d'un modèle de structure. Cela peut-il avoir des conséquences pour les autres modules qui exploitent l'interface du module C ?
 - *Bien sûr ! et même cela peut avoir de lourdes conséquences en termes de devoir ré-écrire toutes les portions de codes qui sont dépendantes des choix antérieurs pour les champs du modèle de structure.*
 - Supposons qu'une personne est responsable d'un module D dont l'interface exporte **SEULEMENT** des fonctions.
 - Le module D peut-il être fiable ? En d'autres termes, la personne responsable de ce module peut-elle garantir que les fonctions exportées dans l'interface fonctionnent toujours correctement ?
 - *OUI il devrait pouvoir être fiable car tout passe par les fonctions mises à disposition dans l'interface et qui sont écrites par la personne responsable du module D. Attention à ne pas offrir de porte dérobée à l'aide de pointeurs car cela détruit les garanties que l'on cherche à mettre en place en cachant le modèle de structure.*
 - Supposons que la personne responsable du module D décide de changer le type un champ d'un modèle de structure dans l'*implémentation* du module D. Cela peut-il avoir des conséquences pour les autres modules qui exploitent l'interface du module D ?
 - *NON si ces détails sont cachés dans l'implémentation. Les autres modules n'ont aucune raison d'être dépendants de ces choix et n'ont aucun besoin d'être recompilés car l'interface du module D ne change pas.*