|  |
| --- |
| 420-SCC  Projet d’apprentissage No.2  Diagrammes de classes UML |

Production du diagramme de classes d’une application existante

Le diagramme UML devrait normalement être réalisé *avant* le début du codage. Quand cette étape n’a pas été réalisée, et que l’application existe déjà, il peut être utile de générer un diagramme pour une application existante. Ceci permet de mieux comprendre sa structure actuelle, et mieux planifier les modifications à apporter à l’application par la suite.

En équipe, vous devez réaliser le diagramme de classes qui correspond au projet *PongMagnetique* dont le code se trouve dans ce même dossier. Il s’agit d’un projet intégrateur réalisé par une équipe d’étudiants, dont on a retiré et altéré la grande majorité du code, tout en laissant sa structure en place (classes et relations entre les classes). Ne vous fiez pas au code souvent absurde qu’il contient, portez plutôt attention à la présence des champs/propriétés (même s’ils semblent peu/pas utilisés par la suite) et à l’utilisation que les classes font les unes des autres.

Vous devez remettre le fichier résultant et son impression au plus tard à l'heure indiquée par le professeur.

Étapes de production

1. Individuellement, lire cet énoncé jusqu’au bout.
2. En équipe, explorez les différentes classes du projet, et dessinez le diagramme de classes correspondant sur votre tableau blanc. N’écrivez pas trop gros, le diagramme final prend un espace considérable. Laissez de l’espace entre les classes pour bien identifier les relations.
3. Prochaine étape : transcription du diagramme final au propre avec ArgoUML . Enregistrer le fichier sous le nom **PA2EquipeXX.zargo**. Collaborez pour le réaliser.
   * Enregistrez souvent pour éviter toute perte de données en cas de problèmes!
4. Réorganisez les éléments du diagramme final pour qu’il soit le plus clair possible, sans excès d’entrecroisements. Sans en abuser, vous pouvez à quelques reprises dupliquer une classe pour simplifier/clarifier (exemple : dupliquer JPanel car plusieurs classes en dérivent). Ceci ne devrait s’appliquer qu’aux classes terminales (celles qui ne pointent pas vers d’autres classes par la suite).
5. Ajoutez votre numéro d’équipe et vos noms directement sur le diagramme (utilisez la zone de texte, le grand A sous le menu de droite dans ArgoUML).
6. Quand le diagramme est terminé :
   * Créer une image avec *Fichier/Enregistrer les graphiques* (choisir png ou jpg)
   * Ouvrir nouveau fichier Word en orientation *Paysage* et choisissez *Marges/Étroites*
   * *Insertion/Image* et choisir votre image de diagramme
   * Sur votre écran, redimensionnez la page jusqu’à ce qu’elle soit la même dimension qu’une vraie feuille de papier…..vous aurez alors une idée de la lisibilité du digramme. Déplacez les éléments, zoomez, etc jusqu’à ce que la lisibilité soit impeccable!
   * Imprimez le fichier Word et remettre au professeur
7. Finalement
   * Copiez votre fichier *.zargo sur* Léa dans la section appropriée.

Critères demandés pour le diagramme UML

* Pour ce travail, on se concentre seulement sur les classes et les relations entre elles : n’ajoutez aucun attribut ni aucune opération. Faites disparaître les deux cases inutiles.
* Toutes les classes du projet doivent apparaître sur le diagramme, avec toutes les relations qu’elles ont avec les autres classes: généralisations, réalisations d’interfaces, dépendances, associations (avec multiplicités si différentes de 1 + toutes les navigabilités).
* Vous pouvez ignorer la notion de paquetage, toutes les classes seront sur le même diagramme.
* Les seules classes Java à inclure sur votre diagramme sont 1) les classes JFrame et JPanel, *uniquement* lorsqu’une classe du projet dérive de celles-ci, ainsi que 2) les interfaces Java (Serializable, Runnable).
* Truc: dans Eclipse, quand vous faites clic-droit sur une classe + *References/Project*, on vous montre les autres classes où celle-ci apparaît dans le code. **Très utile!!**