

# Conception à base de patrons I

## 1- Objectifs

Ce laboratoire permet aux étudiants de se familiariser avec l'implémentation des patrons de conception "**Composite**" et "**Decorator**". Cette implémentation est effectuée à l'aide de visual studio code en C++. Un cadriciel vous est fourni, ainsi que des tests déjà implémentés permettant de vérifier le bon fonctionnement de votre code. Une fois le code complété, vous pourrez ainsi vous assurer que son comportement est correct. On vous suggère d'ailleurs d'aller lire les fichiers de tests afin de comprendre ce qui est demandé.

## 2- Patron Composite (30 points)

Le prototype du logiciel PolyJet permet à l'utilisateur d'organiser et de diviser les parties de son terrain en parcelles et groupes de parcelles. Une parcelle composite peut donc contenir des parcelles ou d'autres parcelles composites, et ce, de façon récursive. Les classes qui participent à ce patron sont:

- `AbsParcelle`
- `ParcelleComposite`
- `AbsParcelleTerrain`

Ici, la classe `AbsParcelle` agit comme le composant abstrait (*abstract component*), `ParcelleComposite` comme le composite, et `AbsParcelleTerrain` comme l'élément feuille.

Remplissez les méthodes dans le fichier `ParcelleComposite.cpp`.

### 3- Patron Décorateur (30 points)

Afin de pouvoir facilement simuler le comportement des parcelles, l'implémentation proposée contient 3 décorateurs qui ajoutent des fonctionnalités spécifiques. Les classes qui participent à ce patron sont:

- AbsParcelleTerrain
- Parcelle
- AbsTerrainDecorateur
- JardinDecorateur
- PelouseDecorateur
- PotagerDecorateur

Remplissez les méthodes dans les fichiers JardinDecorateur.cpp, PelouseDecorateur.cpp, PotagerDecorateur.cpp, et AbsTerrainDecorateur.cpp.

### 4- Questions (40 points)

#### Question 1 (10 points)

Faites 2 diagrammes de classes du logiciel, un pour chaque patron. Indiquez les attributs, les méthodes et les relations entre les classes. Vous pouvez utiliser un autre logiciel qu'Enterprise Architect pour les faire. Vous devez inclure vos 2 diagrammes dans votre fichier tp4.pdf. (voir la section 5- Remise)

#### Question 2 (10 points)

Identifiez les avantages et les désavantages des patrons composite et décorateur.

#### Question 3 (10 points)

Supposons que l'on veuille, dans le cadre du patron composite, obtenir toutes les parcelles feuilles d'un arbre complexe de parcelles. Comment feriez-vous? Sans en faire l'implémentation, que devriez-vous ajouter dans les classes pour y parvenir?

#### Question 4 (10 points)

Au lieu d'utiliser le patron décorateur, un collègue vous propose de simplement faire plusieurs classes pour les différents types de parcelle, qui héritent directement d'une classe abstraite. On aurait alors par exemple ParcelleJardin, ParcellePelouse et ParcellePotager, qui hériteraient directement de AbsParcelleTerrain. Que permet le patron décorateur, qui n'est pas possible avec l'idée de votre collègue?

## 5- Remise

Créez un fichier compressé (.zip) nommé LOG2410\_MatriculeA\_MatriculeB\_TP4.zip. Celui-ci devrait comprendre:

- Un dossier nommé "src" qui contient tous les fichiers .cpp modifiés
- Un fichier nommé "tp4.pdf" qui contient les réponses aux questions. Celui-ci doit inclure les diagrammes de classes de la question 1.