IUT Sénart-Fontainebleau, UPEC
CPOO
IUT Info 2A

Florent Madelaine

2022-23

Semaine 3

Les notions de l'évaluation formative

1. usage : synopsis et DCU.
2. structure : DC, DO.
3. comportement : DSS et DC.
4. Abstraction : DC avec classes abstraites et interfaces, API, couplage faible.

Programme de la semaine

CM : rappels de première année sur les classes abstraites, les interfaces. Illustration par des exemples tirés de java.util. Évocation des API.

TD : Abstraction.

TP : Abstraction et Structure.

Concepts à connaître • classes abstraite• interfaces• couplage faible

TD n° 3

Exercice 1 (Abstraction)**.** Je souhaite proposer une API, par exemple entre une couche logicielle permettant de gérer la persistance des données (par exemple une base de données) et une couche métier utilisant ces données. (NB. en jargon on parle de DAO pour *Data Access Objects*).

Cette API va proposer une classe abstraite A (ou une interface) pour fixer le format d'échange et encapsuler le comportement de données intéressant la couche métier. La couche métier va donc typiquement recevoir un objet d'une classe C concrète héritant de A (ou réalisant A si c'est une interface).

Par quel mécanisme la couche métier peut-elle demander la construction d'un nouvel objet « de type A » ?

Exercice 2 (Abstraction)**.** *Disclaimer : Aucune information préalable sur le rugby n'est nécessaire pour réussir cet exercice.*
Afin de préparer les meilleures tactiques et gagner le prochain tournoi, le séléctionneur de l'équipe de france souhaite modéliser les capacités des joueurs de rugby.

Dans cet exercice, le type des attributs et des méthodes n'est pas très important. Il faut se concentrer sur les classes abstraites et les interfaces, et aussi évidemment quelques classes concrètes qui vont en hériter ou les réaliser.

Question 1 : Les joueurs Les joueurs possèdent un nom, un numéro, une nationalité et un club. Ils possèdent également **toujours** une position. On considérera dans cet exercice 3 types de positions : les *avants*, la *charnière* et les *arrières*. Chaque type de position représente un profil de joueur et les actions d'un joueur dépendent de sa position. De même, le numéro d'un joueur dépend de sa position :

- Un avant a un numéro entre 1 et 8
- Une charnière a un numéro entre 9 et 10
- Un arrière a un numéro entre 11 et 15.

Réalisez un *diagramme de classe* permettant de modéliser ces informations.

Question 2 : Les actions On n'attend pas les mêmes capacités d'un joueur selon sa position.

- Un avant doit pouvoir *déblayer* et *pousser dans la mélée*.
- Un arrière peut *faire un cadrage débordement*, *courir vite* et *jouer au pied*, jouer au pied étant soit *tenter un drop* soit *taper une chandelle*.
- Un charnière sait *réfléchir*, *faire un cadrage débordement* et *jouer au pied*.
- Enfin, tous les joueurs peuvent essayer de *passer* et *boire une bière*.

Enrichissez votre diagramme avec les actions des joueurs. On fera attention à **mettre en avant** le fait que certaines positions ont des capacités en commun.

Exercice 3 (Abstraction)**.** On se penche sur le cas d'un MMOG (*Massive Multiplayer Online Game*).

Le jeu gère des êtres vivants, ce qui signifie qu'ils ont des points de vie. Les animaux et les personnages sont vivants. Les animaux ont une espèce. Il y a deux sorte de personnages. Les PJ (personnages joueurs) qui sont contrôlables par un joueur du jeu et les PNJ (personnages non joueurs) qui sont gérés par le jeu. Les personnages ont des capacités. Les objets ont une valeur. Il y a trois sortes d'objets : les trésors, les artefacts et les véhicules. Le système gère aussi des nuages pour faire joli. Les animaux, les personnages, les véhicules et les nuages peuvent *se déplacer*. Les objets et les animaux sont potentiellement *transportables*. Ils ont un poids. Les véhicules, les animaux et les personnages peuvent potentiellement *transporter* des choses transportables. Ils ne peuvent pas dépasser un certain poids total. Ils peuvent évidemment *déposer* des choses. Les artefacts et certains animaux sont *magiques*, c'est-à-dire qu'ils donnent une capacité. Le moteur de jeu recopie de temps à autres des éléments de manière automatique, par exemple les trésors et les PNJs. Ces éléments sont dit *spawnable*.

Proposez un diagramme de classe en détaillant bien les attributs et méthodes nécessaires (sauf s'il s'agit d'accesseurs classiques pour ne pas surcharger le diagramme).

TP n° 3

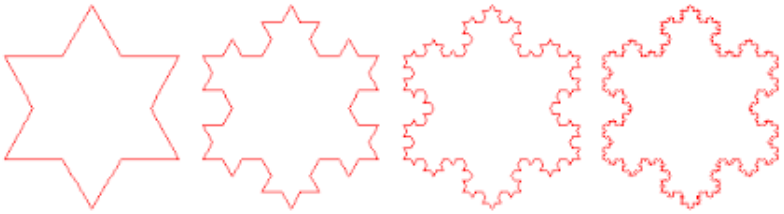
Exercice 4 (Abstraction)**.** On se penche sur le cas d'une partie d'un logiciel permettant de faire de la géométrie en 2 dimension. On se concentre sur les *surfaces avec une aire finie*.

On peut récupérer l'aire d'une surface finie.

Les cercles, rectangles et carrés ont un bord de longueur finie (le périmètre) et on peut récupérer cette longueur. On peut aussi décider si un point appartient à la surface.

Les rectangles et carrés ont leurs bords parallèles aux axes du plan 2D dans lequel on travaille.

Les fractales n'ont pas de bord fini (voir ci-dessous le début de la construction du flocon de Koch.



Vous pouvez trouver plus de détails ici : `http://villemin.gerard.free.fr/Wwwgvmm/Suite/FracCour.htm`

On peut demander un dessin exact pour un rectangle et un carré (un dessin est une chaîne de caractères). On peut demander un dessin approché pour un cercle et une fractale, c'est-à-dire qu'on donne une petite valeur ϵ, et plus on est proche de 0 plus le dessin est précis.

Proposez un diagramme de classe en détaillant bien les attributs et méthodes nécessaires (sauf s'il s'agit d'accesseurs classiques pour ne pas surcharger le diagramme).

Exercice 5 (Structure)**.** Un éditeur désire avoir une structure de représentation de l'ensemble de ses livres édités. Voici la liste des demandes de l'éditeur :

- Un *livre* est représenté par ses métadonnées, c'est à dire un titre, un auteur et un genre. Le texte en lui-même est stocké sous forme de *chapitres*.
- Un chapitre possède un numéro (unique), ainsi que des pages de début et de fin. Un chapitre est également une suite (ordonnée) de *phrases*.
- Une phrase est une suite de *mots*, terminé par une ponctuation. On supposera dans un premier temps qu'il n'y a pas de ponctuation intermédiaire.
- Un mot est une suite de *lettres*, et chaque lettre possède un symbole, mais également des données d'éditions telles que la police de caractères, et des attributs tels que gras, sousligné et/ou italique.

Question 1. Proposez un diagramme de classe pour représenter les livres satisfaisant les demandes de l'éditeur.

Question 2. On souhaite pouvoir modéliser une ponctuation plus complexe, comme les virgules ou les guillemets. Modifiez votre diagramme pour les prendre en compte.
Attention : Une ponctuation n'est pas un mot, et il est important de pouvoir reconstruire la phrase avec la ponctuation au bon endroit.

Question 3. Si je décide de représenter un mot d'un livre comme une composition de lettres, quel soucis potentiel entrevoyez vous avec ce design ?