TD
Programmation orientée objets

Yannick Loiseau yannick.loiseau@uca.fr

Licence 2me année

TD 1

Objets et rappels d'algorithmique

Exercice 1 (Instants) Un instant est défini par un nombre d'heures (entre 0 et 23), de minutes (entre 0 et 59) et de secondes (entre 0 et 59).

Question 1 *:* Est-ce qu'un instant peut changer ? Doit-on définir des méthodes pour lire les valeurs d'un instant ? Doit-on définir des méthodes pour modifier les valeurs d'un instant ?

Question 2 (Mise en œuvre) *:* Écrivez une classe Java permettant de créer des instants.

Question 3 (Différence de deux instants) *:* Écrivez une méthode Java `minus` qui retourne le temps entre deux instants.

Question 4 (Comparaison) *:* Écrivez une méthode `compareTo` qui retourne −1 si l'instant considéré est plus petit qu'un instant `t` passé en paramètre, 0 si ces deux instants sont égaux, et 1 sinon (cette méthode est définie dans l'interface `Comparable<T>` [1]).

Question 5 *:* Ajoutez une méthode `main` à votre programme. Cette méthode `main` crée un tableau d'objets de la classe `Instant` et affiche le résultat de la comparaison de tous les couples possibles.

Question 6 *:* Soit $u = (u_0, \ldots, u_n)$ une liste de $n+1$ instants. Écrivez un algorithme qui retourne le plus petit instant u_i de la liste qui est strictement supérieur à un instant t donné en paramètre.

Question 7 *:* Soit $u = (u_0, \ldots, u_n)$ une liste de $n+1$ instants. Écrivez un algorithme qui trie la liste u, en utilisant l'algorithme précédent. On supposera que la liste ne contient pas deux fois le même élément.

1. http://docs.oracle.com/javase/8/docs/api/java/lang/Comparable.html#

TD 2

Composition et héritage

Exercice 1 (Voitures)

Une voiture de type A est constituée d'un moteur, d'une carrosserie et de quatre roues. La carrosserie est constituée de portes, dont le nombre peut varier entre 2 et 4.

Question 1 *:* Proposez un diagramme de classes pour les voitures A.

Les voitures de type B sont constituées de deux moteurs (l'un à l'avant et l'autre à l'arrière), d'une carrosserie et de six roues. La carrosserie est constituée de portes, dont le nombre peut varier entre 2 et 4.

Question 2 *:* Ajoutez au diagramme de classes précédent les voitures B.

Question 3 *:* Écrivez les fichiers Java correspondant à toutes vos classes.

TD 3

Héritages et surcharges

Exercice 1 (Villes et pays) Un pays est constitué d'un ensemble de villes, mais d'exactement une capitale.

Question 1 (Modèle UML) *:* Proposez un diagramme de classes pour les classes `City`, `Capital` et `Country`.

Question 2 (Implémentation) *:* Écrivez les classes `City` et `Capital` en Java.

Question 3 *:* Comment vous assurez-vous qu'un `Country` ne dispose que d'une seule `Capital` ?

Le paquetage `java.util` contient une interface nommée `Enumeration` [1]. Cette interface déclare deux méthodes : `public boolean hasMoreElements()` et `public Object nextElement()`.

Question 4 *:* En supposant qu'il existe une méthode permettant d'ajouter une `City` à un `Country`, écrivez la fonction qui permet d'ajouter une `Enumeration` de `City` à un Country. La fonction s'appellera : `public boolean addCities(Enumeration<City> cities)`.

Exercice 2 (Héritage multiple) Une personne peut posséder des animaux domestiques. Les animaux domestiques considérés ici sont des chiens, des chats ou des chevaux. Un animal peut courir et crier (on considère qu'ils courent tous de la même manière).

Question 1 *:* Modèle UML Écrivez le diagramme de classes correspondant à cette spécification.

Question 2 *:* Une personne peut posséder un véhicule. Le véhicule est en général une auto ou une moto. Un véhicule a une capacité (deux personne pour la moto, quatre personnes pour la voiture), peut avancer (à condition d'avoir au moins un occupant), et une personne peut y entrer (tant qu'il reste de la place) ou en sortir.

Étendez le diagramme de classes précédant pour le faire correspondre à cette spécification.

1. http://docs.oracle.com/javase/8/docs/api/java/util/Enumeration.html#

Question 3 *:* Écrivez le code de la classe `Vehicule`.

Question 4 *:* Un cheval peut être considéré comme un véhicule.

Modifiez le diagramme de classes précédant pour prendre en compte cette nouvelle hypothèse.

Question 5 *:* Écrivez le code de la classe `Cheval` et de la classe `Moto`.

TD 4

Étude de cas

Exercice 1 (Jeu de tarot (initiation)) Le jeu de tarot se joue avec un jeu de cartes particulier. Au tarot, on différencie les cartes de couleurs (pique, cœur, carreau et trèfle) des atouts. Dans les cartes de couleurs, on trouve les valeurs de 1 à 10, ainsi que le valet, le cavalier, la dame et le roi. Dans les atouts, on trouve les valeurs de 1 à 21, ainsi que l'excuse.

Question 1 (UML) *:* Représentez le diagramme de classes correspondant au jeu de tarot.

Question 2 (Mise en œuvre) *:* Écrivez le code java correspondant aux classes des cartes et du paquet.

À chaque tour, chaque joueur pose une carte à tour de rôle. Le joueur ayant posé la première carte impose la couleur à tous les autres. Celui qui remporte le pli est celui qui possè la carte la plus forte. La valeur des cartes est déterminée de la manière suivante :

- parmi toutes les cartes de la couleur demandée (par le premier joueur), la carte la plus forte est celle ayant la plus forte valeur (le roi remporte donc sur toutes les cartes de la couleur demandée) ;
- toutes les cartes de couleur n'appartenant pas à la couleur demandée ont une valeur nulle et ne peuvent donc pas remporter ;
- le plus fort des atouts joué remporte automatiquement, quelque soit la couleur demandée.

Pour simplifier, on supposera que le premier joueur ne peut pas jouer un atout.

Question 3 (Valeur des cartes) *:* Écrivez une méthode permettant de déterminer la carte la plus forte lorsque quatre cartes `c1`, `c2`, `c3` et `c4` ont été jouées.