

Programmation Java - TP 1

Exercice 1

Définir une classe Primitive qui permet de recevoir par l'intermédiaire de ces 8 constructeurs les différents types de base (types primitifs). La classe permet l'accès et la mise jour de la valeur introduite par le constructeur. On définira les méthodes *get* et *set* avec la fonctionnalité adéquate.

Exercice 2

On demande la réalisation d'une classe Array qui gère comme propriété un tableau d'objets tout en offrant la possibilité de redimensionnement du tableau après chaque appel à une méthode *add* permettant d'ajouter un nouvel objet au tableau. Donner deux versions de la méthode *add* : la première avec 1 seul paramètre (l'objet), l'autre version avec 2 paramètres (l'objet et la position d'insertion). Définir aussi une méthode *remove* pour supprimer un objet désigné par son indice dans le tableau ou par sa référence, et une méthode *get* qui retourne l'objet désigné par son indice.

Exercice 3

1. Créer une classe Liquide contenant seulement une méthode *imprimer()* qui affiche : "je suis un liquide" ;
2. Créer 2 classes dérivées de la classe Liquide, les classes Cafe et Lait, dont les méthodes *imprimer()* affichent respectivement "je suis un Café", "je suis du Lait" ;
3. Créer une classe Tasse ayant un attribut *l* de la classe Liquide et une méthode *AjouterLiquide* (Liquide *li*) et une méthode *imprimer()* ;
4. Tester l'utilisation : créer un tableau de Tasses qui contient différents liquides.

Exercice 4

On demande la réalisation d'une liste chaînée d'objets implémentée récursivement (on ne supposant pas l'existence des classes *LinkedList*, *Vector*, etc.). on ne réalisera alors une classe « List » caractérisé par les propriétés suivantes : *value* (Object) et *next* (List , un pointeur sur la même classe).

La classe « List » offre aussi les méthodes de services suivantes :

- *boolean isEmpty()*
- *void add(Object elemnt)*
- *int size()*
- *Object get(int index)*
- *Object remove(int index)*
- *void clear()*
- *void set(int index, Object element)*
- *String toString()*

Exercice 5

Il s'agit de réaliser une application de gestion d'Hôtel. On considère alors que l'hôtel est caractérisé par une liste de chambres. Une chambre est caractérisée par les propriétés suivantes : numéro, type (simple ou double), état (libre ou occupé), nombre de lits, et le prix. L'application gère aussi les clients chacun par les propriétés suivantes : *id*, *nom*, *ville*, et *pays*.

Lorsqu'un client réserve une chambre, on devait enregistrer sa réservation dans une liste de réservations (une classe). Sachant que chaque réservation est définie par : le client, la chambre réservée, et la période (date d'arrivée, par exemple définir une classe *Date*, et le nombre de jours).

Lorsqu'un client occupe une chambre, on devrait enregistrer l'information dans une liste d'occupations (une classe). Chaque occupation est caractérisée par le client et la chambre.

En fin, la classe « Hotel » devrait fournir, entre autres, les services suivants :

1. Ajouter une nouvelle chambre
2. Ajouter un nouveau client
3. Chercher une chambre par son numéro
4. Réserver une chambre pour un client
5. Attribuer une chambre à un client
6. Déterminer quel client occupe une chambre donnée par son numéro

On demande la réalisation des classes Java permettant l'implémentation de la description précédente.