

Exercice Objets

CONTENU	
Exercice 1 Les poupées Russes	1
Exercice 2	1

Pour chaque exercice vous ferez dans un premier temps le diagramme UML, puis le code en Java

EXERCICE 1 LES POUPEES RUSSES

On veut écrire un programme simulant des poupées russes de différentes tailles. Chaque poupée a une taille donnée, peut s'ouvrir ou se fermer, peut contenir une autre poupée et être contenue dans une autre poupée.

Faire le diagramme de classe `PoupeeRusse`.

Écrire une classe `PoupeeRusse` contenant les méthodes suivantes :

- un constructeur
- Boolean `ouvrir()` : ouvre la poupée si elle n'est pas déjà ouverte et si elle ne se trouve pas à l'intérieur d'une autre poupée
- Boolean `fermer()` : ferme la poupée si elle n'est pas déjà fermée et si elle ne se trouve pas à l'intérieur d'une autre poupée
- Boolean `placerDans(PoupeeRusse p)` : place la poupée courante dans la poupée `p` si c'est possible. Il faut que la poupée courante soit fermée et ne soit pas déjà dans une autre poupée, que la poupée `p` soit ouverte et ne contienne aucune poupée, et que `p` soit d'une taille supérieure à la poupée courante.
- Boolean `sortirDe(PoupeeRusse p)` : sort la poupée courante de la poupée `p` si elle est dans `p` et si `p` est ouverte.

EXERCICE 2 LES MONTRES

On veut réaliser un programme qui représente des montres et les gens qui les portent.

Une montre donne l'heure et les minutes. On peut initialiser une montre soit à partir d'un couple heure/minute donné, soit en utilisant l'heure affichée par une autre montre. Il doit être possible de faire avancer l'heure d'une montre en ajoutant une minute (attention, les minutes sont limitées à 60 et les heures à 24).

- Faire le diagramme de la classe `Montre`
- Écrire une classe qui représente les montres telles que décrites au-dessus et une méthode principale qui crée une montre qui affiche 13h45 et une autre montre qui est un clone de la première.

Une personne a un nom et peut éventuellement porter une montre. On peut faire porter une montre donnée à une personne, si elle n'en a pas déjà une. On peut aussi lui enlever sa montre si elle en a une. Une personne peut demander l'heure à une autre, qui lui donne l'heure sous forme d'une chaîne de caractères, en consultant sa montre si elle en a une (sinon elle peut retourner une chaîne vide).

- Faire le diagramme de la classe `Personne`
- Écrire une classe qui représente les personnes telles que décrites au-dessus.
- On veut faire en sorte que chaque montre ne soit portée que par une seule personne. Proposez des ajouts/modifications des deux classes précédentes pour garantir cela.

--- FIN DU DOCUMENT ---