Java - énoncés de TD - semaine 9
Pierre Parrend

1 Exceptions

Ecrivez un programme Charpentier, qui crée un tableau 'poutres' de 10 entiers.

1.1 Gestion des exceptions

- Tentez d'accéder à l'élement d'indice 10 dans votre tableau, que se passe-t-il ?
- Au lieu de s'arrêter prématurément, vous voulez que votre programme affiche 'Carrramba, encorrre rraté !'. Comment faire ?

1.2 En cas de lecture de fichier

Après avoir lu les plans, le charpentier doit les ranger en ordre pour ses collègues.

De même, un fichier ouvert par un programme doit être libéré, pour permettre à d'autres objets d'y accéder.

- Imaginez que votre programme ouvre des fichiers, avant l'appel erroné au tableau. Affichez un message a l'écran qui indique que vous pouvez les fermer.

1.3 Délegation

Votre programme Charpentier est un programme de supervision, il ne doit pas traiter les données.

- Créez une classe Manoeuvre, contenant une méthode travail(), dans lequel les manipulations de votre tableau ont lieu.
- C'est le Charpentier qui gère la situation. Il commande à Manoeuvre un travail(), mais le traitement d'éventuelles erreurs est fait par lui. Adapter le code en conséquence.

2 Classes abstraites et héritage

2.1 Définition d'une Classe abstraite

Ecrivez une classe abstraite DefinitionVoiture, qui contienne les attributs suivants :

- String marque
- String type

Et les déclarations des méthodes suivantes :

- ouvrirPorte(),
- fermerPorte(),
- demarrer(),
- sArreter().

De plus, une voiture dispose d'une méthode carteGrise(), implémentée, qui affiche à l'écran 'Je suis une Voiture' suivi du type et de la marque.

2.2 Classes d'implémentation

Vous avez dans votre garage deux voitures : une 2CV, Citroen, et une Velsatis, Renault.

Ecrivez, en respectant les consignes, une classe pour chacune de ces voitures.

- Chaque marque de voiture correspond à une classe. L'attribut est donc renseigné par défaut.
- Vos classes héritent de la classe abstraite DefinitionVoiture.
- Dans le constructeur, passez le type en paramètre.
- Implémentez les méthodes qui le nécessitent. Le code de la méthode est composé de l'affichage des manipulations nécessaires.
- A chaque appel de méthode, rappeler la marque et le type de chaque voiture.
- Pour la 2CV, ouvrirPorte et fermerPorte doivent afficher 'je donne un coup de pied dans la porte - je l'ouvre/la ferme'
- Pour la Velsatis, il faut afficher 'approcher la carte magnétique de la portière'.
- Pour la 2CV, demarrer et sArreter affichent : 'allumer le contact'/ 'retirer la clé du contact'
- Pour la Velsatis, demarrer affiche : 'insérer la carte dans le tableau de bord', et sArreter affiche : 'Veuillez attendre un instant, option momentanément non disponible'.

3 Recursivité

On considère un package jeu, qui contient deux classes :

- CreationJoueurRecursif.
- JoueurRecursif.
- La classe CreationJoueurRecursif crée un JoueurRecursif, et appelle la méthode passe() de ce joueur.

La classe JoueurRecursif contient :

- un attribut entier numero,
- un constructeur dont le paramètre entier sert à initialiser l'attribut numero, et qui annonce le numero du joueur,
- une méthode passe, qui crée un JoueurRecursif dont le numero est supérieur de 1 au numero du JoueurRecursif en cours, et appelle la méthode passe de ce nouveau JoueurRecursif.

Vous veillerez à ce que le nombre de joueurs créés soit 11.