

	ING1 : EXAMEN D'ANALYSE ORIENTE OBJET C++ EXAMEN PAPIER (DOCUMENTS NON AUTORISES) DUREE 2 HEURES	
L'équipe pédagogique	Ref : <i>ING1-ANAL ORIENT OBJ PROG</i>	
A l'intention des étudiants d'ING1 GM	06 juin 2018	

Modalités

- Durée : 2 heures.
- Vous devez rédiger votre copie à l'aide d'un stylo à encre exclusivement.
- Toutes vos affaires (sacs, vestes, trousse, etc.) doivent être placées à l'avant de la salle.
- Aucun document n'est autorisé.
- Aucune question ne peut être posée aux enseignants, posez des hypothèses en cas de doute.
- Aucune machine électronique ne doit se trouver sur vous ou à proximité, même éteinte.
- Aucune sortie n'est autorisée avant une durée incompressible d'une heure.
- Aucun déplacement n'est autorisé.
- Aucun échange, de quelque nature que ce soit, n'est possible.

Exercice 1 : Un peu d'algèbre élémentaire (8 points)

Voici une version informatique, orientée objets, d'un cours d'algèbre élémentaire.

Un *groupe* est un *ensemble* muni d'une loi de composition interne qui possède un certain nombre de propriétés (associativité, existence d'un élément neutre et d'un symétrique).

Par ailleurs, un *anneau* est un groupe muni d'une seconde loi de composition interne, associative, distributive par rapport à la loi du groupe et admettant aussi un élément neutre.

Dans une vision orientée objets, cela veut dire que la classe `Anneau` hérite de la classe `Groupe` qui hérite de la classe `Ensemble`.

C'est ce que nous proposons d'implémenter dans cet exercice, en se restreignant ici (pour des raisons de simplicité) aux anneaux finis Z/pZ (Z , ensemble des entiers relatifs).

1. Définir de façon simple la classe `EnsembleFini` représentant l'ensemble des entiers naturels entre 0 et p (il suffit simplement de stocker l'entier (positif) p).
2. Ajouter le constructeur correspondant.
3. Définir ensuite la classe `Groupe` qui hérite de la classe `EnsembleFini` et contient une méthode `add`, prenant deux entiers non signés en argument et retournant l'entier non signé correspondant à leur somme modulo p .
4. Définir ensuite la classe `Anneau` qui hérite de la classe `Groupe` et contient une méthode `mult`, prenant deux entiers non signés en argument et retournant l'entier non signé correspondant à leur produit modulo p .

Exercice 2 : Véhicules (8 points)

1. Après avoir lu en entier le sujet, créer le diagramme de classes pour décrire au mieux le système de gestion des véhicules.
2. Définir une classe `Vehicule` qui a pour attributs des informations valables pour tout type de véhicule : sa marque, sa date d'achat, son prix d'achat et son prix courant.
3. Définir un constructeur prenant en paramètre les trois attributs correspondant à la marque, la date et le prix d'achat (le prix courant sera calculé plus tard).
4. Définir une méthode publique `void affiche(ostream&) const;` qui affiche l'état de l'instance, i.e. la valeur de ses attributs.
5. Définir ensuite deux classes `Voiture` et `Avion`, héritant de la classe `Vehicule` et ayant les attributs supplémentaires suivants :
 - pour la classe `Voiture` :
 - o sa cylindrée ;
 - o son nombre de portes ;
 - o sa puissance ;
 - o son kilométrage ;

- pour la classe `Avion` :
 - o son type (hélices ou réaction) ;
 - o son nombre d'heures de vol.

Pour chacune de ces classes, définir un constructeur qui permette l'initialisation explicite de l'ensemble des attributs, ainsi qu'une méthode affichant la valeur des attributs. Constructeurs et méthode d'affichage devront utiliser les méthodes appropriées de la classe parente !

6. Ajouter ensuite à la classe `Vehicule`, une méthode `void calculePrix()` qui donne le prix courant. On calculera ce prix courant en soustrayant au prix d'achat 1% par année écoulée depuis la date d'achat.

7. Redéfinir cette méthode dans les deux sous-classes `Voiture` et `Avion`, de sorte à calculer le prix courant en fonction de certains critères, et mettre à jour l'attribut correspondant au prix courant :

- Pour une voiture, le prix courant est égal au prix d'achat, moins :
 - o 2% pour chaque année depuis sa fabrication jusqu'à la date actuelle ;
 - o 5% pour chaque tranche de 10'000 km parcourus (on arrondit à la tranche la plus proche) ;
 - o 10% s'il s'agit d'un véhicule de marque « Renault » ou « Fiat » (choix totalement arbitraire qu'on est bien sûr libre de modifier) ;
 - o et plus 20% s'il s'agit d'un véhicule de marque « Ferrari » ou « Porsche » (même remarque que ci-dessus).
- Pour un avion, le prix courant est égal au prix d'achat, moins :
 - o 10% pour chaque tranche de 1000 heures de vol s'il s'agit d'un avion à réaction,
 - o 10% pour chaque tranche de 100 heures de vol s'il s'agit d'un avion à hélices.

Le prix doit rester positif, s'il est négatif votre programme doit le signaler et le gérer en utilisant le mécanisme des exceptions.