



CENTRE SCOLAIRE SAINTE-JULIENNE

Exercices Java – POO Composition

Série 3 - Enoncés

I- Mise en situation

Tu es développeur dans une société et tu dois passer un test en langage Java. A travers une série d'exercices, tu dois comprendre et maîtriser le langage Java pour obtenir la prime salariale.

II- Travail à réaliser

1. Effectuer l'exercice proposé par le professeur.
2. Analyser (diagramme de classe) l'énoncé du point III correspondant au numéro de l'exercice demandé.
3. Réaliser l'exercice.
4. Visualiser le travail.
5. Sauvegarder le document suivant les instructions données.
6. Imprimer le(s) document(s).

III- Enoncés

1. Ex01

Définir une classe **Point** contenant 2 variables d'instance entières (x et y), 1 constructeur à 2 paramètres, les méthodes accesseurs et la méthode toString. Un **Carre** est un Point avec une variable d'instance supplémentaire de type *double(cote)*. On y retrouve 1 constructeur à 3 paramètres, les méthodes accesseurs et la méthode toString. Un **Cercle** est un Point avec une variable d'instance supplémentaire de type *double(rayon)*. On y retrouve 1 constructeur à 3 paramètres, les méthodes accesseurs et la méthode toString. Un **Rectangle** est un Point avec deux variables d'instance supplémentaires de type *double(longueur, largeur)*. On y retrouve 1 constructeur à 4 paramètres, les méthodes accesseurs et la méthode toString. Un **TriangleRectangle** est un Point avec deux variables d'instance supplémentaires de type *double(base, hauteur)*. On y retrouve 1 constructeur à 4 paramètres, les méthodes accesseurs et la méthode toString. Prévoir dans ces classes, une méthode calculerPerimetre retournant le périmètre de la figure et une méthode calculerAire retournant l'aire de la figure. La classe Test est une applet qui crée une instance de chacune des figures et qui affiche, avec la méthode drawString, le périmètre et l'aire de chacune de ces figures. A cet effet, l'utilisation du polymorphisme est impérative.

2. Ex02

Classe: **CompteEnBanque** Variables d'instance: String *titulaire*, int *solde*, String *numero*.

Variable de classe: double *tauxInteret*.

Classe: **CompteCourant** Variables d'instance: String titulaire, int solde, String numero.

Classe: **LivletEpargne** Variables d'instance: String titulaire, int solde, String numero.

Variable de classe: int *soldeMinimum*.

Prévoir pour chacune de ces classes les *méthodes habituelles*, une méthode *retrait* (qui diminue la valeur du solde), une méthode *depot* (qui augmente la valeur du solde) et une méthode *calculerInteret*. Le corps des méthodes retrait et depot est écrit dans la classe CompteEnBanque. Le solde minimum renseigne la valeur en-dessous de laquelle on ne peut pas descendre. Elle adapte éventuellement le solde et affiche un message (retrait accepté ou refusé). La méthode calculInteret adapte le solde. Le taux d'intérêt pour un livret est 3 fois le taux d'intérêt de base.

Classe: **Test** est une *application* qui crée 5 comptes courants et 5 livrets d'épargne. Elle calcule ensuite pour chaque compte en banque et en utilisant le polymorphisme, le nouveau solde via la méthode calculInteret. Elle teste ensuite les 2 cas de retrait pour un livret d'épargne.

3. EX03

Classe: **Adresse** Variables d'instances: String *rue*, String *localite*.

Prévoir les *méthodes habituelles*.

Classe: **Personnel** Variables d'instance: String *nom*, double *salaire*, Adresse *adresse*.

Prévoir les *méthodes habituelles* et une méthode *augmenterSalaire* qui augmente le contenu de la variable d'instance salaire du nombre de % passé en paramètre.

Classe: **Patron** (hérite de Personnel) Variables d'instance: String *nom*, double *salaire*, Adresse *adresse*, String *nomSecretaire*, int *anneesAnciennete*.

Prévoir les *méthodes habituelles* et une méthode *augmenterSalaire* qui augmente le contenu de la variable d'instance salaire du nombre de % passé en paramètre augmenté de 0,5% par année d'ancienneté (on fera appel à augmenterSalaire de Personnel après avoir adapté le pourcentage).

Classe: **Test** est une *application* qui crée des instances de Personnel et de Patron. Elle calcule ensuite pour chaque membre du personnel de l'entreprise et en utilisant le polymorphisme, le nouveau salaire via la méthode augmenterSalaire.