OFPPT

مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle
et de la Promotion du Travail

Examen de Passage

Session Juin 2007

Filière : TSDI

Epreuve : Pratique

Niveau : Technicien Spécialisé

Durée : 4 h 30

Barème : 40 Pts

Variante n° 8

Partie 1 – JAVA (23 Pts)

Vous allez programmer le calcul des salaires hebdomadaires des employés d'une entreprise.

Cette entreprise comporte plusieurs types d'employés :

- Des employés qui sont payés suivant le nombre d'heures qu'ils ont travaillé dans la semaine. Ils sont payés à un certain tarif horaire et leurs heures supplémentaires (au-delà de 40 heures) sont payées 30 % de plus que les heures normales.
- D'autres employés, payés de la même façon, mais leurs heures supplémentaires sont payées 50 % de plus que les heures normales.
- Les commerciaux sont payés avec une somme fixe à laquelle on ajoute 1 % du chiffre d'affaires qu'ils ont fait dans la semaine.
- Vous commencerez par écrire une classe Employe dont hériteront les autres classes.
- Le temps est compté pour faire l'examen. Pour ne pas avoir trop de modificateurs vous simplifierez en ne donnant qu'un seul modificateur setInfosSalaire pour entrer ou modifier les informations brutes nécessaires au calcul des salaires (nombre d'heures de travail, chiffre d'affaire,...). N'essayez pas de faire du polymorphisme avec cette méthode.

- Les classes comporteront au moins 2 constructeurs : un qui ne prend en paramètre que le nom de l'employé et l'autre qui prend en paramètres le nom et toutes les informations pour le calcul du salaire.
- Le calcul des salaires se fera dans la méthode getSalaire() qui sera utilisée pour faire du polymorphisme.

Une classe `Paie` comportera une unique méthode `main()` qui entrera les informations sur des employés des différents types (au moins 3 commerciaux). Les employés seront enregistrés dans un tableau `employes`.

Au moins un des employés sera créé avec le constructeur qui n'a que le nom en paramètre, et vous entrerez ensuite les informations pour son salaire avec la méthode `setInfosSalaire`. Pour au moins un autre employé, vous utiliserez le constructeur pour entrer les informations sur le salaire.

La méthode `main()` affichera le salaire hebdomadaire de chacun des employés dans une boucle "`for`" qui parcourra le tableau des employés. Vous utiliserez le polymorphisme avec un accesseur pour le salaire. L'affichage aura **exactement** la forme : "Ahmed gagne 6500 dh". Vérifiez les calculs des salaires !

1. On va généraliser en ajoutant la possibilité de travailler avec plusieurs entreprises. Ecrivez une classe Entreprise. Une entreprise aura un nom et aura des employés que vous enregistrerez à l'aide d'un Vector. Il devra être possible d'ajouter et d'enlever des employés dans une entreprise.

2. Vous écrirez une méthode toString qui décrit une entreprise en affichant son nom et les noms de tous ses employés (parcourez le `ArrayList` avec les indices).

3. Pour tester, ajoutez quelques employés dans 2 entreprises et faites afficher les noms des entreprises et de leurs employés.

`class Employe`	**2 Pts**
`class Commercial`	**2 Pts**
`class EmployeHoraire`	**3 Pts**
`class Paie`	**6 Pts**
`class Entreprise`	**4 Pts**
`class TestEntreprise`	**2 Pts**
`class EmployeException`	**2 Pts**
`abstract class Employe`	**2 Pts**

Partie 2 – Sql Server (17 Pts)

Soit le schéma de la base de données :

PALETTE (Code, Libellé, Nature, Poids) **MAGASIN** (NumM, Nom, VilleM) **CENTRE** (numC, société, villeC, classe) **LIVRAISON** (numM, numC, Code, quantité) numP étant un code

A) Construire la base de données, ainsi les tables. Vous n'oublierez pas les contraintes d'intégrités avec la bonne syntaxe **(2 Pts)**

B) Exprimer les requêtes suivantes en SQL.

1) Les villes où l'on trouve des magasins (sans duplication d'information!).**(0.75 Pt)**
2) Le numéro des centres (de distribution) effectuant une livraison au magasin "1" (dans l'ordre des numéros de centre). **(0.75 Pt)**
3) Le code des palettes livrées par un centre d'Ecublens à un magasin de Sion. **(0.75 Pt)**
4) Les couples "NumM, NumC" correspondant chacun à un magasin et un centre localisés dans la même ville. **(0.75 Pt)**
5) La nature des palettes livrées par le centre "5". **(0.75 Pt)**
6) Le numéro des centres dont la classe est inférieure à celle du centre "1". **(0.75 Pt)**
7) La quantité totale de palettes "5" en cours de livraison par le centre "5". **(0.75 Pt)**
8) Le nom et la classe des centres (de distribution) localisés à Ecublens. **(0.75 Pt)**
9) Le nom et la ville de tous les magasins. **(0.75 Pt)**
10) Les livraisons concernant plus de 5 palettes.
(en précisant le numéro de magasin, celui du centre et le code des palettes de ces livraisons) **(0.75 Pt)**
11) Le nom des magasins à qui le centre "5" livre au moins une palette. **(0.75 Pt)**
12) Le numéro des magasins à qui un centre, d'une ville différente, livre une ou plusieurs palettes. **(0.75 Pt)**
13) Le numéro des centres livrant à un magasin une quantité de palettes "4" plus grande que la quantité moyenne de ces palettes "4", livrées à ce même magasin. **(0.75 Pt)**
14) Le code (-barre) des palettes livrées par un centre d'Ecublens. **(0.75 Pt)**
15) Les couples de villes tels qu'un centre de la première livre quelque chose à un magasin de la seconde. **(0.75 Pt)**
16) Le nombre total de magasins à qui le centre "1" livre au moins une palette. **(0.75 Pt)**
17) Pour chaque groupe de palettes livrées (au même magasin), le code de la palette, le numéro du magasin et la quantité totale correspondante. **(0.75 Pt)**
18) le code des palettes livrées par un centre à un magasin de la même ville. **(0.75 Pt)**
19) le code des palettes livrées à un magasin de Morges. **(0.75 Pt)**
20) le code des palettes dont la nature commence par "Fr". **(0.75 Pt)**