Concepts de POO en JAVA

Héritage – Classes Abstraites – Interfaces - polymorphisme

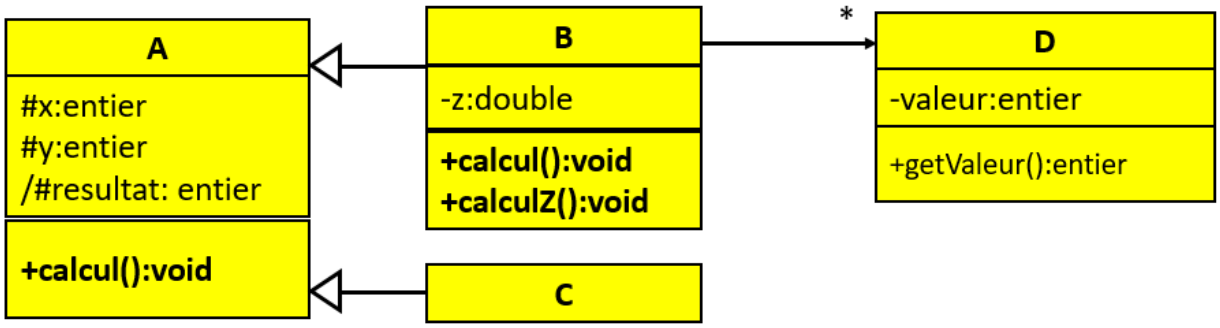
Partie 1 : Héritage

Exercice 1 : généralités

1. Quel est le rôle de l'héritage ? et à quel moment l'utiliser ?
2. En général, quelle est la visibilité des attributs d'une classe abstraite ?
3. Que signifie le polymorphisme ? donner un exemple
4. Quelles sont les limites de l'héritage et quelles sont les alternatives? donner un exemple
5. Préciser le rôle du mot clé super
6. Préciser le rôle de surcasting et de sous casting. Donner des exemples

Exercice 2 : Modélisation & Implémentation

Soit le diagramme de classes suivant :



1. Les principales différences entre les classes dans une hiérarchie d'héritage :
   - Quelles sont les principales différences entre A et B ?
   - Quelles sont les principales différences entre A et C ?
   - Quelles sont les principales différences entre C et B ?
2. A partir du diagramme ci-dessus, parler du polymorphisme
3. Proposer une implémentation de ce diagramme sachant que :
   - La méthode calcul() dans A calcule la somme de x et y et stocke le résultat dans resultat.
   - La méthode calculZ() calcule la somme des valeurs de la collection lesD dans B(valeur d'objet D) et stock le résultat dans z.
   - La méthode calcul() dans B réalise l'opération suivante : resultat=x+y+z.
4. Le constructeur sans paramètres dans A ne doit pas exister. Définir un constructeur dans A qui initialise x et y. si le constructeur sans paramètres n'est pas défini, quelles sont les modifications à apporter à votre code source pour qu'il fonctionne correctement ?
5. Utilisation du modèle :
   - Objets de la classe D
     – Créer 6 objets (d1,d2,d3,d4,d5 et d6) de type D avec différentes valeurs de valeur
   - Objets de la classe mère A
     – Créer un objet a1 de type A, puis calculer le résultat et l'afficher à l'écran
   - Création des objets des classes filles B et C
     – Créer un objet b1 de type B et un objet c1 de type C en référençant ces objets par leurs classes.
     – Créer un objet b2 de type B et un objet c2 de type C en référençant ces objets par leur classe mère.
     – Quelles sont les différences entre b1 et b2, et c1 et c2
   - Soit le code source suivant :

```
public class Program {
public static void main(String[] args) {
D d1, d2, d3, d4, d5, d6;
d1 = new D(1);d2 = new D(2);d3 = new D(3);
d4 = new D(4);d5 = new D(5);d6 = new D(6);
A a1 = new A(1, 1);B b1 = new B(2, 2);A b2 = new B(2, 2);
C c1 = new C(4, 4); C c2 = new C(5, 5);
B b3=new A(3,3); C c3=new A(6,6);
b1.clacul(); b2.clacul(); b3.clacul();
c1.clacul(); c2.clacul(); c3.clacul();
b1.addD(d1); b1.addD(d2);b1.addD(d3);b2.addD(d1);
b2.addD(d2);b3.addD(d3);b1.calculZ(); b2.calculZ();
List<A> liste1 = new ArrayList<>();
liste1.add(a1); liste1.add(b1);liste1.add(b2);
liste1.add(c1);liste1.add(c2); liste1.add(c3);
List<B> liste2 = new ArrayList<>();
liste2.add(a1); liste2.add(b1);liste2.add(b2);
liste2.add(c1);liste2.add(c2);liste2.add(c3);
for (A a : liste1) {
a.calculZ();
System.out.println(a.resultat);
}
for (B a : liste2) {
a.calculZ();
System.out.println(a.resultat);
}
liste2 = liste1;
liste1 = liste2;
}
}
```

- Pour le bout de code source ci-dessus :
  - Signaler les problèmes qui existent s'il y en a et corriger les.
  - En se basant sur le code ci-dessus (probablement corrigé), parler du polymorphisme, surcatsting et sous casting au moment de leurs utilisation.

Partie 2 : Classes abstraites

Exercice 1 : généralités

1. Quel est le rôle d'une classe abstraite ?
2. Quelle relation existe-elle entre l'héritage et les classes abstraites ?
3. A quel moment préférez-vous mettre en place une classe abstraite à la place d'une classe mère concrète ? et vice versa
4. Est-ce que une classe abstraite accepte un constructeur ?
5. Est-ce que une classe abstraite accepte un constructeur ?

Exercice 2 : modélisation et implémentation

On reprend le code source de l'exercice 2 de la partie 1 (version corrigée),

1. A quel moment vous jugez nécessaire de rendre la classe A abstraite ?
2. Rendre la classe A abstraite
   - Quelles sont les erreurs éventuelles suite à ce changement ?
   - Corriger les erreurs
   - Est-ce que la signification du polymorphisme, casting vont changer ?
3. On souhaite afficher des étoiles (*) dans la classe B et des plus (+) dans la classe C à l'aide d'une méthode print
   - Implémenter la méthode print() de B pour afficher les étoiles sachant que le nombre d'étoiles est égal à resultat

- Implémenter la méthode print() de C pour afficher les plus sachant que le nombre de plus est égal à resultat
- En utilisant liste1 et une seule boucle afficher les plus et les étoiles de chaque objet dans liste1.

Partie 3 : Interfaces

Exercice 1 : généralités

1. Quelle est le rôle d'une interface ?
2. Quelle relation existe-elle entre classes concrètes, classes abstraites et interfaces ?
3. Est-ce que l'interface contient des attributs ? quel type de données peut contenir ?
4. A quel moment préférez-vous mettre en place une interface à la place d'une classe abstraite ? et vice versa

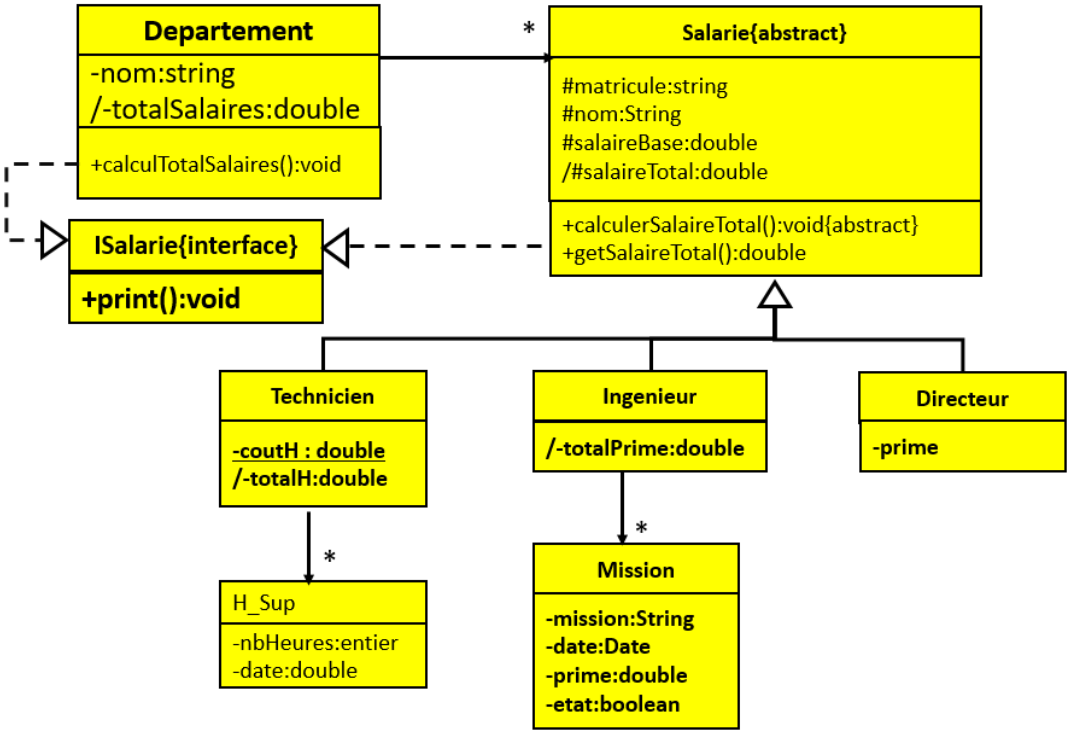
Exercice 2 :

On reprend le code source de l'exercice 2 de la partie 2 (version corrigée),

5. A quel moment vous jugez important d'ajouter une interface Iinterface1 implémentée par la classe abstraite A
6. A quel moment vous jugez important d'ajouter une interface Iinterface2 implémentée par la classe C
7. Ajouter dans Interface1 une méthode f() et dans Interface2 une méthode g()
8. Quels sont les changements à apporter au code pour qu'il fonctionne normalement ?

Application 1 : gestion des salariés

Soit le diagramme de classes suivant :



NB. Ce diagramme est loin de la bonne modélisation : il s'agit d'un exemple pour appliquer l'héritage, classes abstraites et interfaces, polymorphisme, casting

Questions

1. Implémenter le diagramme de classes ci-dessus
2. Définir un constructeur qui permet d'initialiser un salarié (matricule, nom, salaireBase)
3. Constructeurs des classes filles
   - Définir un constructeur pour initialiser un technicien (matricule, nom, salaireBase, grade)
   - Définir un constructeur pour initialiser un ingénieur (matricule, nom, salaireBase, grade)
   - Définir un constructeur pour initialiser un directeur (matricule, nom, salaireBase, prime)
4. Calcul :
   - Calculer le salaire total d'un technicien sachant que son salaireTotal=salaireBase+ totalH
     – totalH est la sommes des montants (nbHeures*CoutH) durant le mois

- Calculer le salaire d'un ingénieur sachant que son salaireTotal=salaireBase+ toatlPrime
  - totalPrime est la somme des primes des différentes missions effectuées durant le mois
- calculer le salaire d'un directeur sachant que son salaire est salaireBase+prime

5. affichage

   affichage du bulletin de technicien

```
----Buletain paie 12/2018----------
matricule:009   Grade A
Nom:ali
prénom:baba
Salaire de base: 6000.00DH
Salaire total: 7000.00DH
Heures suplémentaire du mois 12/2018
3/12/2018    2H   -->400 DH
12/12/2018   1H   -->200 DH
3/12/2018    2H   -->400 DH
```

- implémenter print de technicien pour afficher le résultat ci-dessus

  Affichage du bulletin de l'ingénieur

```
----Buletain paie 12/2018----------
matricule:012   Grade C
Nom:tati
prénom:tata
Salaire de base: 12000.00DH
Salaire total: 16000.00DH
missions du mois 12/2018
Config server       800 DH
Formation angular6  3200 DH
```

- implémenter print de ingénieur pour afficher le résultat ci-dessus

  affichage du bulletin du directeur

```
----Buletain paie 12/2018-------
matricule:021
Nom:ali
prénom:baba
Salaire de base: 21000.00DH
Prime: 7000.00DH
Salaire total: 28000 DH
```

- implémenter print de directeur pour afficher le résultat ci-dessus

  affichage de la liste des salariés

matricule	Salarie	type	Grade	SalaireBase	Prime/Hsup	Salaire total
021	lali lala	directeur		21000	7000	28000
012	tata tati	ingénieur	C	12000	4000	16000
009	ali baba	ingénieur	A	6000	1000	7000

- Implémenter print de département pour afficher le résultat ci-dessus

6. Application :
   - Créer une classe Program ayan le main
   - Créer une collection salaries capable de stocker des salariés
   - Créer et initialiser deux techniciens tec1 et tec2, calculer leurs salaires total
   - Créer et initialiser deux ingénieurs ing1 et ing2, calculer leurs salaires total
   - Créer et initialiser un directeur et calculer son salaire

Application 2 : K-plus proches voisins

- Un algorithme pour l'apprentissage supervisé (Data mining)
- À partir d'un ensemble E d'individus où chaque individu a des caractéristiques (dans ce cas les coordonnées) et une classe (dans ce cas la couleur), (1)
  - pour un nouveau individu X (ici point noir) dont on ne connait que ses caractéristiques et on souhaite connaitre sa classe (2)
    - On identifie les k-plus proches voisins à X
      - Calculer la distance de l'individu X à chaque individu de l'ensemble E (3)
      - Trier ces distances par ordre croissant (3)
      - Extraire les k premiers (4)
  - La classe de X est celle de la majorité de ces k premiers (5)