

TP N°2 en Java

Cycle ingénieur : 2^{ème} Année GI

Exercice I :

Banque		
- NCompte	: int	
- Solde	: float	
- CIN	: String	
+ <<Constructor>> Banque (int NCompte, float Solde, String CIN)		
+	deposer (float Somme)	: void
+	retirer (float Somme)	: void
+	avoirSolde ()	: float
+	avoirInf ()	: String

1. Ecrire le programme "Banque.java" permettant d'implémenter la classe "Banque"
2. Ecrire un programme "Test.java" pour :
 - Créer un compte : 1, 5000.75, "AB 1200"
 - Afficher les informations de ce compte
 - Afficher son solde
 - Déposer 500
 - Afficher les informations de ce compte
 - Afficher son solde
 - Retirer 200
 - Afficher les informations de ce compte
 - Afficher son solde

Exercice 2 :

Soit la classe prédéfinie dans "java.net"

```

class InetAddress{
    public static InetAddress getLocalHost();
    //retourne l'adresse Internet de la machine locale
    public static InetAddress getByName(String);
    //retourne l'adresse d'une machine à partir de son nom
    public String getHostName();
    //retourne le nom d'une machine à partir d'une adresse Internet
}
    
```

- Ecrire un programme "test.java" pour afficher l'adresse Internet et le nom d'une machine (locale ou distante). Evoquer le cas où la machine est donnée en argument

Exercice 3 :

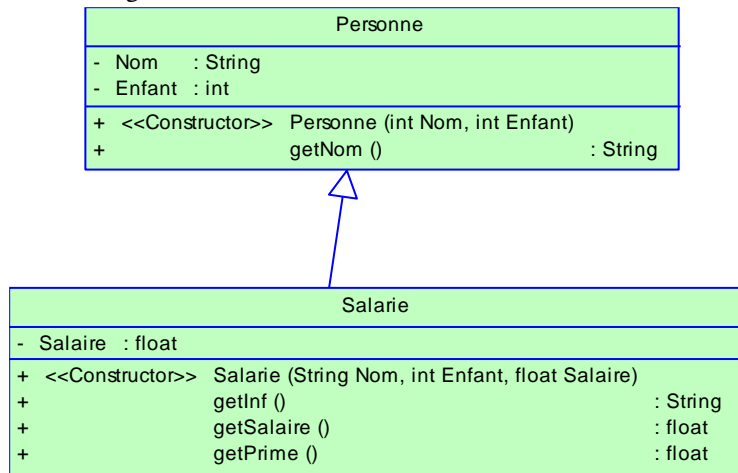
Le but est d'implémenter la classe "Robot" dont la spécification est donnée ci-dessous :

Robot		
- Abs	: int	
- Ord	: int	
- Orientation	: String	
+ <<Constructor>> Robot (int Abs, int Ord, String Orientation)		
+ <<Constructor>> Robot ()		
+	avancer ()	: void
+	tournerDroite ()	: void
+	tournerGauche ()	: void

1. Ecrire le programme "Robot.java" et "Test.java" (test.java contenant la méthode main())
2. Modifier le programme "Robot.java" de telle sorte que la méthode "avancer()" permettra de déplacer le robot d'une position donnée en argument.

Exercice 4 :

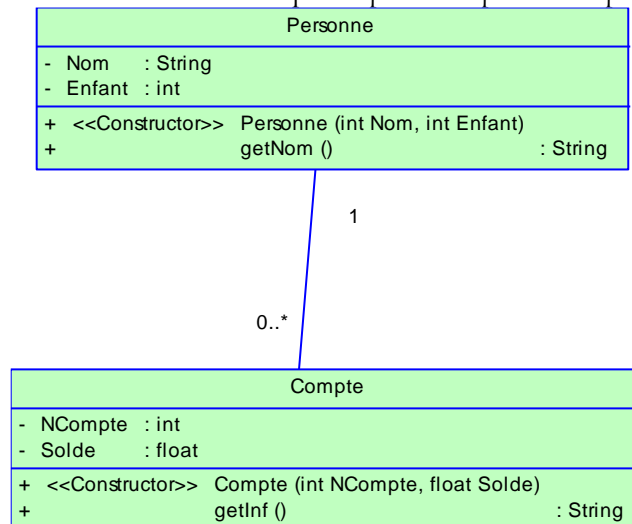
Soit le diagramme de classes d'UML suivant :



1. Créer les classes "Personne.java" et "Salarie.java", sachant que la prime est calculée à la base de la formule suivante : $\text{prime} = (5 * \text{Salaire} * \text{Nombre d'enfant}) / 100$
2. Ecrire un programme "Test.java" pour manipuler plusieurs personnes et plusieurs Salariés.

Exercice 5 :

Soit le diagramme de classe traduisant le fait qu'une personne peut avoir plusieurs comptes :



Implémenter les deux classes en proposant deux solutions :

- Ajout d'une référence de personne dans la classe compte
- Ajout d'un tableau de références de compte dans la classe personne

Exercice 6 :

On voudrait réaliser un programme de gestion d'un stock de matériels informatique et de logiciels.

- 1- Créer une interface produit avec comme méthodes :
String getDesignation()
float getPrixUnitaire ()
float getQuantité()
char getNature () qui devra retourner la nature Matériel ('M') ou Logiciel ('L') du produit.
- 2- Réaliser une classe Matériel et une classe Logiciel qui implémente l'interface Produit. La classe Matériel dispose d'un constructeur de 3 paramètres pour communiquer les attributs Désignation, PrixUnitaire et Quantité. La classe Logiciel dispose d'un constructeur à 5 paramètres : Désignation, PrixUnitaire, Quantité, éditeur et AnnéeEdition. La classe Logiciel dispose aussi de deux méthodes de plus : getEditeur () et getAnneeEdition () .
- 3- Réaliser une classe **gestionDeStock** qui dispose des membres suivants :
 - Une LinkedList permettant de stocker la liste de Matériels et Logiciels du stock
 - Une méthode ajouter () qui reçoit en paramètre un Produit.
 - Une méthode lister () qui affiche la liste des informations adéquates de chaque produit du stock.
- 4- Imaginer et implémenter d'autres opérations sur le stock dans la classe **GestionDeStock**.