

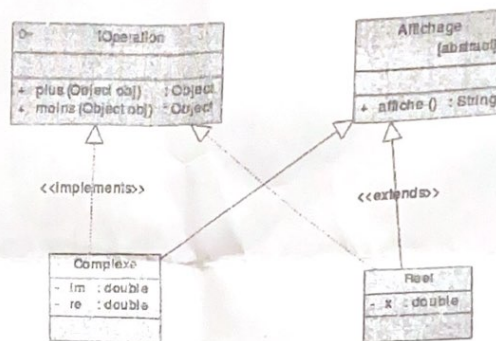
Feuille de TP numéro 4

Objectifs

Classes abstraites ; Interfaces ; Gestion des Exceptions.

Exercice 1

Considérons le diagramme de classe suivant :



1. Écrire l'interface **IOperation** qui contient deux méthodes `plus()` et `moins()` qui ont un seul argument de type `Object` et qui retournent un `Object`.
2. Écrire la classe abstraite **Affichage**.
3. Écrire la classe **Complexe** qui hérite de **Affichage** et qui implémente **IOperation** :

```
public class Complexe extends Affichage implements IOperation { ... }
```

La classe **Complexe** contient :

 - deux attributs `re` pour désigner la partie réelle du nombre complexe et `im` pour sa partie imaginaire ;
 - un constructeur qui a deux arguments : `re` et `im` ;
 - une implémentation des deux méthodes : `plus()` qui retourne la somme de l'objet complexe appelant la méthode et l'objet complexe argument de la méthode et `moins()` qui retourne leur soustraction ;

Remarque : N'oubliez pas de caster `Object` à `Complexe` ;

 - une redéfinition de `affiche()` qui retourne une chaîne de caractère de la forme `a + ib`.
4. Écrire la classe **Reel** qui hérite de **Affichage** et qui implémente **IOperation**. La classe **Reel** contient :
 - un attribut `x` et un constructeur qui a `x` comme argument ;
 - une implémentation des deux méthodes : `plus()` et `moins()` ;

- une redéfinition de `afficher()` qui retourne une chaîne de caractère de la forme x .
- 5. Écrire une classe **Test** qui contient la méthode `main()`.
 - déclarer un tableau de type **IOperation** de taille 4 ;
 - instancier les deux premiers éléments du tableau en utilisant la classe **Complexe**, et les deux derniers en utilisant la classe **Reel** ;
 - afficher le tableau.
 - calculer la somme des deux nombres complexes et la somme des deux nombres réels et afficher les résultats.
- 6. **Packages.** Modifier toutes les classes pour que :
 - l'interface **IOperation** soit dans le package **TP6.Interfaces**
 - la classe abstraite **Affichage** soit dans le package **TP6.Abstr**
 - les classes **Complexe**, **Reel** et **Test** soient dans le package **TP6.Bean**.

Exercice 2

On désire gérer la sécurité dans un système d'exploitation. Pour cela, on définit les informations nécessaires des utilisateurs autorisés à accéder au système comme suit :

- **Nom utilisateur** : Nom de l'utilisateur qui sera utilisé pour la connexion ;
- **Mot de passe** : Mot de passe associé à l'utilisateur ;
- **Numéro utilisateur** : Numéro d'utilisateur ;
- **Privilège** : Le privilège dont bénéficie l'utilisateur et qui peut être 0 pour administrateur, et 1 pour utilisateur. Toute valeur affectée au privilège qui est différente de 0 et 1 déclenche une exception.

1. Créer la classe **Utilisateur** qui contient un nom, un mot de passe, un numéro et un privilège.
2. Ajouter un constructeur qui a comme arguments les quatre attributs cités ci-dessus ;
3. Définir la fonction `afficher ()` qui affiche toutes les informations d'un utilisateur ;
4. Créer un programme de test appelé **Form1** qui permet de créer deux instances d'utilisateurs ayant les informations suivantes et de les afficher.

Propriété	Utilisateur1	Utilisateur2
Nom d'utilisateur	Admin	User
Mot de passe	0000	1111
Numéro d'utilisateur	1	2
Privilège	0	1

5. Ajouter une exception nommée **Err** qui sera déclenchée si le privilège associé à l'utilisateur est différent de 0 ou 1.
6. Modifier le code **Form1** pour qu'on puisse créer les utilisateurs suivants et de les afficher.

Propriété	Utilisateur1	Utilisateur2
Nom d'utilisateur	Admin	User
Mot de passe	0000	1111
Numéro d'utilisateur	1	2
Privilège	3	1