

9 avril 2008

Contrôle d'informatique no 3

Durée : 1 heure 45'

Nom :

Groupe : ☐

Prénom :

Barème sur 110 points

No	1	2
Nombre points	50 points	60 points

Remarque générale : toutes les questions qui suivent se réfèrent au langage de programmation Java (à partir du JDK 5.0) et les réponses doivent être rédigées à l'encre et d'une manière propre sur ces feuilles agrafées.

Sujet no 1.

On considère un projet Java contenant les trois classes ci-dessous groupées dans un même package, à savoir *cms_ctr3*. On demande d'écrire les résultats affichés à l'écran suite à l'exécution de ce projet.

```

package cms_ctr3;

public class A
{
    double a = 1;

    A()
    {
        this(10);
        System.out.println("A sans arg !");
    }

    A(double arg)
    {
        this.a = arg;
        System.out.println("A avec 1 arg !");
    }

    static A calculer(A arg1, double arg2)
    {
        System.out.println("calculer de classe A.");
        return new A(arg2 * arg1.a);
    }

    void analyser(Object arg)
    {
        System.out.print("L'analyse de A : ");
        if(arg instanceof A)
        {
            System.out.println("C'est un A !");
        } else
        {
            System.out.println("C'est un Object !");
        }
    }

    void afficher( )
    {
        System.out.println("Un objet A avec a : " + a + ".");
    }
} //fin de la classe A

```

```

package cms_ctr3;

public class B extends A
{
    char b = 'A';

    B()
    {
        this('Z', 5);
        System.out.println("B sans arg !");
    }

    B(char arg)
    {
        b = arg;
        System.out.println("B propre !");
    }

    B(double arg)
    {
        super(arg);
        System.out.println("B heritage !");
    }

    B(char arg1, double arg2)
    {
        super(arg2);
        b = arg1;
        System.out.println("B complet !");
    }

    static B calculer(B arg1, int arg2)
    {
        arg1.b = (char)(arg1.b + arg2);
        System.out.println("calculer de classe B - var_1.");
        return arg1;
    }

    static B calculer(int argPrim, B argSecond)
    {
        A.calculer(argSecond, argPrim);
        calculer(argSecond, argPrim);
        System.out.println("calculer de classe B - var_2.");
        return argSecond;
    }

    void analyser(Object arg)
    {
        System.out.print("L'analyse de B : ");
        if(arg instanceof B)
            System.out.println("C'est un B !");
        else
            super.analyser(arg);
    }

    void afficher()
    {
        super.afficher();
        System.out.println("Un objet B avec b = " + b + ".");
    }
} //fin de la classe B

```

```

package cms_ctr3;

public class CP_Ctr3Exo1
{
    public static void main(String[ ] args)
    {
        Object [ ] tab = new Object[7];
        System.out.println("Première partie !");
        tab[0] = new B(-10);
        System.out.println("-----");
        tab[1] = new A();
        System.out.println("-----");
        tab[2] = new A(22);
        System.out.println("-----");
        tab[3] = new B();
        System.out.println("-----");
        tab[4] = new B('D', 33);
        System.out.println("-----");
        tab[5] = new B('K');
        System.out.println("-----");
        System.out.println("-----");

        System.out.println("Deuxième partie !");
        ((A)tab[2]).analyser(tab[1]);
        ((A)tab[2]).analyser((A)tab[1]);
        ((A)tab[2]).analyser(tab[4]);
        ((A)tab[2]).analyser((B)tab[4]);
        System.out.println("-----");
        ((B)tab[5]).analyser(tab[1]);
        ((B)tab[5]).analyser((A)tab[1]);
        ((B)tab[5]).analyser(tab[4]);
        ((B)tab[5]).analyser((B)tab[4]);
        System.out.println("-----");
        ((A)tab[5]).analyser(tab[4]);
        ((A)tab[5]).analyser((A)tab[4]);
        System.out.println("-----");
        System.out.println("-----");

        System.out.println("Troisième partie !");
        tab[6] = A.calculer((A)tab[1], 2);
        ((A)tab[1]).afficher();
        ((A)tab[6]).afficher();
        System.out.println("-----");
        tab[6] = B.calculer(2, (B)tab[4]);
        ((A)tab[4]).afficher();
        ((A)tab[6]).afficher();
        System.out.println("-----");
        tab[6] = B.calculer((B)tab[5], 2);
        ((B)tab[5]).afficher();
        ((B)tab[6]).afficher();
        System.out.println("-----");
    } //fin de la méthode main
} //fin de la classe principale

```

[illegible]

[illegible]

Sujet no 2.

Le but de cet exercice est d'écrire une classe appelée *NC* qui pourra être utilisée pour l'instanciation d'objets correspondant à des nombres complexes définis sous forme algébrique par leur partie réelle et imaginaire.

2.1 Il faut d'abord écrire le code d'une interface publique appelée *ICalculable*, qui fait partie d'un package appelé *cms_ctr3* et qui précise les en-têtes de trois méthodes utiles pour calculer avec des nombres complexes, à savoir :

- la méthode *conjuguer* sans argument qui, appelée pour un objet de type *NC* correspondant à un nombre complexe $(x + iy)$, va transformer l'objet appelant de sorte qu'il va correspondre au nombre complexe conjugué $(x - iy)$;
- la méthode *creerConjugué* sans argument qui, appelée pour un objet de type *NC* correspondant à un nombre complexe $(x + iy)$, va créer et retourner l'adresse d'un nouvel objet de type *NC* correspondant au nombre complexe conjugué $(x - iy)$;
- la méthode *calculerProduit* avec un argument de type *NC* qui, appelée pour un objet de type *NC*, va créer et retourner l'adresse d'un nouvel objet de type *NC* correspondant au produit entre les nombres complexes associés à l'argument et à l'appelant de la méthode.

Rappel : $(x + iy) \cdot (u + iv) = (xu - yv) + i(xv + yu)$

On présente ci-dessous le squelette de l'interface *ICalculable* et il faut compléter ce canevas en fonction des indications données en commentaire.

```
//déclaration de package
```

```
.....  
//en-tête de l'interface ICalculable  
.....
```

```
.....  
{ //en-tête de la méthode conjuguer  
.....
```

```
.....  
//en-tête de la méthode creerConjugué  
.....
```

```
.....  
//en-tête de la méthode calculerProduit  
.....
```

```
.....  
} //fin de l'interface ICalculable
```

2.2 Il faut maintenant écrire le code de la classe *NC* qui fait partie du même package *cms_ctr3*, implémente l'interface *ICalculable* et qui doit prévoir :

- deux champs publics destinés à stocker la partie réelle et, respectivement, imaginaire du nombre complexe qui est représenté par l'objet *NC* ;
- un constructeur avec deux arguments qui stocke, dans l'objet qui sera créé, la valeur du premier argument dans le champ prévu pour la partie réelle et la valeur du deuxième argument dans le champ prévu pour la partie imaginaire ;
- un constructeur sans argument qui appelle le constructeur avec deux arguments afin de créer un objet *NC* correspondant "par défaut" au nombre complexe $(1 - i)$;
- une méthode d'instance *calculerModule* sans argument, qui retourne le module du nombre complexe correspondant à l'objet appelant ;

Rappel : Le module d'un nombre complexe $(x + iy)$ est une valeur réelle positive donnée par la relation $\sqrt{x^2 + y^2}$.

- une méthode statique (surchargée) *calculerModule* qui retourne le module du nombre complexe correspondant à son seul argument de type *NC* ;
- les implémentations des méthodes (redéfinies) discutées au point **2.1** ;
- des versions statiques surchargées pour les méthodes discutées au point **2.1**.

On présente ci-dessous le squelette de la classe *NC* et il faut compléter ce canevas en fonction des indications données en commentaire.

```
//déclaration de package
.....

.....

//en-tête de la classe NC
.....

.....

{
    //déclaration du champ correspondant à la partie réelle
    .....

    .....

    //déclaration d'un champ correspondant à la partie imaginaire
    .....

    .....
```



```
//définition du constructeur avec deux arguments
```

```
//définition du constructeur sans argument
```

```
//définition de la méthode d'instance calculerModule
```

```
//définition de la méthode statique (surchargée) calculerModule
```

```
//implémentation de la méthode d'instance (redéfinie) conjuguer
```

```
//définition de la méthode statique (surchargée) conjuguer
```

```
//implémentation de la méthode d'instance (redéfinie)  
//creerConjugue
```


2.3 Afin de créer et de manipuler des objets de type *NC*, il faut finalement compléter, en suivant les indications données sous formes de commentaires, la classe principale *CP_Ctr3Exo2* qui fait partie du même package *cms_ctr3*.

Rappel : La numérotation des éléments d'un tableau commence avec l'indice 0.

Remarque : Afin de pouvoir utiliser les méthodes d'instance prévues dans une classe à l'aide d'un objet appelant ayant comme type effectif cette classe mais dont la référence est stockée dans une variable de type classe ascendante, il faut faire des casts appropriés.

```
package cms_ctr3;
```

```
public class CP_Ctr3Exo2
```

```
{    public static void main(String[] args)
    {
```

```
        //créez une variable de type tableau d'objets de type
        //Object (où Object est la classe prédéfinie "racine")
```

```
        //créez un tableau de 4 éléments de type Object et
        //stockez son adresse dans la variable créée ci-dessus
```

```
        //créez un objet de type NC correspondant au nombre
        //complexe (-3 + 4i) et stockez son adresse dans le
        //premier élément du tableau
```

```
        //créez un objet de type NC correspondant au nombre
        //complexe (2 - i) et stockez son adresse dans le
        //deuxième élément du tableau
```

```

//créez un objet de type NC à l'aide du constructeur
//sans argument et stockez son adresse dans le
//troisième élément du tableau
.....

.....

.....

//stockez dans le dernier élément du tableau l'adresse
//d'un nouvel objet de type NC correspondant au
//conjugué du nombre complexe associé au premier
//élément du tableau
.....

.....

.....

//écrivez une instruction par laquelle l'objet de type NC
//vers lequel pointe le deuxième élément du tableau
//va correspondre au conjugué du nombre complexe auquel
//il était associé précédemment
.....

.....

.....

//créez une nouvelle variable de type NC et y stockez
//l'adresse d'un nouvel objet de type NC correspondant
//au produit entre les nombres complexes associés
//au premier et au deuxième élément du tableau
.....

.....

.....

//créez une nouvelle variable d'un type primitif
//approprié et y stockez le module du nombre complexe
//associé au premier élément du tableau
.....

.....

.....

} //fin de la méthode main
} //fin de la classe principale

```