

# IF121 - TP2 Bis CORRECTION

20 Octobre 2004

## Exercice 1 : Nombres complexes \*

La difficulté de cet exercice est la représentation d'un nombre complexe, car l'ordinateur ne sait pas les gérer. Pour cela on doit utiliser 2 variables : par exemple, le nombre  $a + ib$  sera représenté par les 2 variables de type *double*  $a$  et  $b$ . Pour l'addition et la multiplication, on doit donc faire, à chaque fois, 2 calculs séparés (partie réelle et partie imaginaire).

```
public static void exercice1 (){
    double n1, n1i, n2, n2i;
    n1 = Deug.readDouble();
    n1i = Deug.readDouble();
    n2 = Deug.readDouble();
    n2i = Deug.readDouble();
    Deug.println ("n1 + n2 = "+(n1+n2)+" + "+(n1i + n2i)+"i");
    Deug.println ("n1 * n2 = "+((n1*n2)-(n1i*n2i))+" + "+((n1*n2i)+(n2*n1i)+"i");
}
```

## Exercice 2 : Résolution d'une équation du second degré \*\*

Il faut juste utiliser des conditionnelles simples pour tester la valeur du déterminant.

```
public static void exercice2 (){
    double a, b, c, discriminant;
    a = Deug.readDouble();
    b = Deug.readDouble();
    c = Deug.readDouble();
    if (a == 0) {
        Deug.println ("1 solution: "+ (-c/b));
    }
    else {
        discriminant = Math.pow(b, 2) - (4*a*c);
        Deug.println("Discriminant = "+discriminant);
        if (discriminant < 0) Deug.println ("Pas de solution");
        else if (discriminant == 0) Deug.println ("1 solution: " + (-b/(2*a)));
        else {
            double sol1 = (-b-Math.sqrt(discriminant))/(2*a);
            double sol2 = (-b+Math.sqrt(discriminant))/(2*a);
            Deug.println ("2 solutions: "+ sol1 + " et " + sol2);
        }
    }
}
```

**Exercice 3 :** *Ouvrir une fenêtre \**

```
public static void exercice3 (){
    int width = Deug.readInt ();
    int height = Deug.readInt ();
    if ((width <= 1024) && (height <= 768) && (width >= 0) && (height >= 0)){
        Deug.startDrawings(width, height);
        int a = Deug.readChar (); // attente
        Deug.stopDrawings();
    }
    else {
        Deug.println ("La taille doit etre inferieure a 1024x768");
    }
}
```

**Exercice 4 :** *Un point \**

```
public static void exercice4 (){
    int x = Deug.readInt ();
    int y = Deug.readInt ();
    if ((x >= 0) && (y >= 0) &&
        (x < 320) && (y < 240)) {
        Deug.startDrawings(320, 240);
        Deug.drawPoint(x, y);
        int a = Deug.readChar (); //attente
        Deug.stopDrawings();
    }
    else {
        Deug.println ("Le point n'est pas dans la fenetre");
    }
}
```

**Exercice 5 :** *Une ligne \*\**

```
public static void exercice5 (){
    int x1 = Deug.readInt ();
    int y1 = Deug.readInt ();
    int x2 = Deug.readInt ();
    int y2 = Deug.readInt ();
    if ((x1 >= 0) && (y1 >= 0) && (x2 >= 0) && (y2 >= 0) &&
        (x1 < 640) && (y1 < 480) && (x2 < 640) && (y2 < 480) &&
        (x1 != x2) && (y1 != y2)) {
        Deug.startDrawings(640, 480);
        Deug.drawLine(x1, y1, x2, y2);
        int a = Deug.readChar (); //attente
        Deug.stopDrawings();
    }
}
```

**Exercice 6 : Afficheur digital \*\*\***

Pour cet exercice, le mieux est d'utiliser un tableau et d'écrire une méthode qui simulera l'afficheur. Les paramètres de cette méthode sont respectivement l'abscisse, l'ordonnée et les états des diodes de l'afficheur.

```
public static void afficheur (int x, int y, boolean[] etats) {
    int a = x + 50;
    int b = y + 50;
    int c = y + 100;
    if (etats[0]) Deug.drawLine (x, y, a, y);
    if (etats[1]) Deug.drawLine (x, y, x, b);
    if (etats[2]) Deug.drawLine (a, y, a, b);
    if (etats[3]) Deug.drawLine (x, b, a, b);
    if (etats[4]) Deug.drawLine (x, b, x, c);
    if (etats[5]) Deug.drawLine (a, b, a, c);
    if (etats[6]) Deug.drawLine (x, c, a, c);
}
```

Pour utiliser cette méthode, on doit initialiser le tableau en fonction du chiffre qui est donné par l'utilisateur. Il suffit ensuite d'appeler la méthode précédente.

```
public static void exercice6 (){
    boolean etats[] = {false, false, false, false, false, false, false} ;
    switch(Deug.readInt()) {
    case 0: etats[0] = etats[1] = etats[2] = etats[4] = etats[5] = etats[6] = true;
        break;
    case 1: etats[2] = etats[5] = true;
        break;
    case 2: etats[0] = etats[2] = etats[3] = etats[4] = etats[6] = true;
        break;
    case 3: etats[0] = etats[2] = etats[3] = etats[5] = etats[6] = true;
        break;
    case 4: etats[1] = etats[2] = etats[3] = etats[5] = true;
        break;
    case 5: etats[0] = etats[1] = etats[3] = etats[5] = etats[6] = true;
        break;
    case 6: etats[0] = etats[1] = etats[3] = etats[4] = etats[5] = etats[6] = true;
        break;
    case 7: etats[0] = etats[2] = etats[5] = true;
        break;
    case 8: etats[0] = etats[1] = etats[2] = etats[3] = etats[4] = etats[5] = etats[6] = true;
        break;
    case 9: etats[0] = etats[1] = etats[2] = etats[3] = etats[5] = etats[6] = true;
        break;
    }
    Deug.startDrawings(60, 110);
    afficheur (5, 5, etats);
    char d = Deug.readChar (); //attente
    Deug.stopDrawings();
}
```