IF121 - TP3 - CORRECTION

27 Octobre 2004

RAPPEL : les solutions des ces exercices ne sont pas uniques, seule une version possible est donnée dans cette correction.

Exercice 1 : *Révisions*
Les expressions suivantes sont-elles correctes ? (justifiez)

1. Deug.println ("Hello") ; -> OUI
2. Deug.println (Hello) ; -> NON, il manque les guillemets.
3. int x = Deug.readInt() ; -> OUI
4. 25 = Deug.readInt() ; -> NON, 25 est une constante entière, pas une variable.
5. int x = Deug.readDouble() ; -> NON, on ne peux pas mettre un double dans un int (on peut faire le contraire).
6. x = 5 ; -> OUI, le '=' désigne une assignation.
7. x == 5 ; -> NON, le '==' désigne un test.
8. char c = 'abcd' ; -> NON, un seul caractère est admis, sinon on utiliserait une chaîne de caractères.
9. if (x == 5) ; {...} -> NON, le ';' termine une expression, or le 'if' n'est pas terminé.
10. if (x = 5) { ... } -> NON, il faut une expression booléenne, pas une assignation.
11. if (x) { ... } ; -> OUI, ssi x est de type *boolean*
12. if (x == 25) { ... } else { ... } -> OUI

Exercice 2 : *Calculs sur les booléens **

```
static void exercice2 () {
    boolean x, y;
    int a, b, c;
    /* Initialisation des variables [...] */
    Deug.println (!x || y);
    Deug.println ((b%a==0) && (c%b==0) || y);
    x = (b > 5) && (a%b == 5);
    y = a%c == 0;
    Deug.println (!x || y);
}
```

Exercice 3 : *Calculs sur les booléens (bis) **

```
static void exercice3 () {
    boolean x, y;
    /* Initialisation des variables [...] */
    boolean result = false;
    if (!x) { result = true; }
    else { if (y) result = true; }
    Deug.println (result);
    result = false;
```

```
        if (b%a==0) { if (c%b==0) result = true; }
        else { if (y) result = true; }
        Deug.println (result);
        result = x = y = false;
        if (b > 5) { if (a%b == 5) x = true;}
        if (a%c == 0) y = true;
        if (!x) { result = true; }
        else { if (y) result = true; }
        Deug.println (result);
    }
```

Exercice 4 : *Référendum **
Si on sait être malin, cet exercice devient très facile :-) Il faut utiliser une variable supplémentaire pour compter le nombre de fois qu'un électeur a répondu *oui*.
Une solution possible est :

```
    static void flush () { char toto = Deug.readChar (); }
    static void exercice4 () {
        int nb_oui = 0;
        for (int i = 0 ; i < 5 ; i++) {
            if (Deug.readChar () == 'o') nb_oui++;
            flush();
        }
        if (nb_oui >= 3) Deug.println ("OUI");
        else Deug.println("NON");
    }
```

Exercice 5 : *Calcul d'une durée **

```
    private static void exercice5 () {
        int N = Deug.readInt();
        Deug.println ("Jours = " + (N/86400));
        N %= 86400;
        Deug.println ("Heures = " + (N/3600));
        N %= 3600;
        Deug.println ("Minutes = " + (N/60));
        Deug.println ("Secondes = " + (N%60));
    }
```

Exercice 6 : *Résolution d'une équation du second degré - acte II ***
On reprend le principe de l'exercice du TP2-bis, mais en séparant, dans le cas où le discriminant est négatif, la partie réelle et la partie imaginaire comme dans l'exercice 1 du TP2-bis.

Exercice 7 : *Additionneur 4 bits ****
Pour cet exercice, la seule difficulté est de trouver les expressions boolénnes qui permettent d'additionner 2 nombres sur 4 bits. Pour cela, il faut écrire la table de vérité !!! On trouve alors pour un bit i :
- $Somme = a_i \oplus b_i \oplus r_{i-1}$
- $Retenue = a_i \cdot b_i + ((a_i \oplus b_i) \cdot r_{i-1})$

Remarque : pour les conversions entre bases, tout est dans le cours.

```
    static void exercice7 () {
        boolean a0, a1, a2, a3;
        boolean b0, b1, b2, b3;
        boolean s0, s1, s2, s3;
        boolean r0, r1, r2, r3;
        int x = Deug.readInt();
```

```
        int y = Deug.readInt();
        /* Conversion x en base 2 */
        a0 = x%2 != 0;
        x /= 2;
        a1 = x%2 != 0;
        x /= 2;
        a2 = x%2 != 0;
        x /= 2;
        a3 = x%2 != 0;
        /* Conversion de y en base 2 */
        b0 = y%2 != 0;
        y /= 2;
        b1 = y%2 != 0;
        y /= 2;
        b2 = y%2 != 0;
        y /= 2;
        b3 = y%2 != 0;
        /* Addition */
        s0 = a0 ^ b0;
        r0 = a0 && b0;
        s1 = a1 ^ b1 ^ r0;
        r1 = ((a1 ^ b1) && r0) || (a1 && b1);
        s2 = a2 ^ b2 ^ r1;
        r2 = ((a2 ^ b2) && r1) || (a2 && b2);
        s3 = a3 ^ b3 ^ r2;
        r3 = ((a3 ^ b3) && r2) || (a3 && b3);
        /* Conversion du resultat en base 10 */
        int result = 0;
        if (s0) result = 1;
        if (s1) result += 2;
        if (s2) result += 4;
        if (s3) result += 8;
        Deug.println ("Resultat = " + result);
    }
```

Exercice 8 : *QCM ****

Pour faire court, et ne pas gener la compréhension, on ne va traiter que deux questions, avec 3 propositions.

```
    static void exercice8 () {
        int r1=0, r2=0;
        Deug.println ("<Question1>: 1) <reponse1> 2) <reponse2> 3) <reponse3>");
        switch (Deug.readInt()) {
        case 1 : r1 = -1; break;
        case 2 : r1 = 1; break;
        case 3 : r1 = -1; break;
        case 4 : Deug.println("Reponse inconnue"); r1 = 0; break;
        }
        Deug.println ("<Question2>: 1) <reponse1> 2) <reponse2> 3) <reponse3>");
        switch (Deug.readInt()) {
        case 1 : r2 = -1; break;
        case 2 : r2 = -1; break;
        case 3 : r2 = 1; break;
        case 4 : Deug.println("Reponse inconnue"); r2 = 0; break;
        }
        if (r1+r2 < 0) Deug.println("Note = 0/2");
        else Deug.println("Note = " + (r1 + r2) + "/2");
        Deug.println("Detail: Question1=" + r1 + "pts, Question2=" + r2 + "pts");
    }
```

Exercice 9 : *Rectangle **

Cet exercice est conçu pour écrire une méthode qui sera (très) utile dans l'exercice suivant.(bien sur, il ne faudra pas oublier d'ouvrir une fenêtre graphique avant d'utiliser cette méthode)

```
static void myDrawRect(int x, int y, int l, int h) {
    int y1 = y+h;
    int x1 = x+l;
    Deug.drawLine(x, y, x1, y);
    Deug.drawLine(x, y, x, y1);
    Deug.drawLine(x, y1, x1, y1);
    Deug.drawLine(x1, y, x1, y1);
}
```

Exercice 10 : *Histogramme ****

Il suffit d'utiliser la méthode de l'exercice précédent pour afficher l'histogramme. Les notes, la moyenne et la hauteur de la fenêtre sont passées en paramètre à la méthode, pour plus de lisibilité.

```
static void exercice10 (int n1, int n2, int n3, int moy, int height) {
    int l = 40, x = 0, y, h;
    height--;
    y = height - n1;
    h = height - y;
    myDrawRect(x, y, l, h);
    x += l;
    y = height - n2;
    h = height - y;
    myDrawRect(x, y, l, h);
    x += l;
    y = height - n3;
    h = height - y;
    myDrawRect(x, y, l, h);
    x += l;
    y = height - moy;
    h = height - y;
    myDrawRect(x, y, l, h);
}
```

Exercice 11 : *Trétraèdre ****

La formule de projection en perspective d'un objet sur le plan déquation $z = 0$ est la suivante (simple application du théorème de Thalès) : soit le point de coordonnées (X, Y, Z), son projeté est le point de coordonnées (x, y) avec $x = \frac{d*X}{Z}$ et $y = \frac{d*Y}{Z}$ où d est la distance entre l'oeil est l'écran (on suppose que lóeil est sur l'axe (0, Z)). Pour la translation, il suffit d'ajouter une certaine valeur aux coordonnées. L'affichage se fait en utilisant la fonction *drawLine*.

```
class Point2D {
    public int X;
    public int Y;
    public Point2D (int X, int Y) {
        this.X = X;
        this.Y = Y;
    }
}
```

```
class Point3D extends Point2D {
    public int Z;
    public Point3D (int X, int Y, int Z) {
        super (X, Y);
        this.Z = Z;
    }
}

class Tetraedre {
    private static int transX = 100;
    private static int transY = 100;
    private static int transZ = 50;
    private static int d = 32;

    private static Point2D transfo (Point3D p3) {
        return new Point2D (256*p3.X/(p3.Z+transZ)+transX,256*p3.Y/(p3.Z+transZ)+transY);
    }

    private static void drawLine (Point2D p1, Point2D p2){
        Deug.drawLine (p1.X, p1.Y, p2.X, p2.Y);
    }

    private static void drawTetraedre () {
        Point2D p1 = new Point3D (5,5,20);
        Point2D p2 = new Point3D (5,0,15);
        Point2D p3 = new Point3D (0,5,25);
        Point2D p4 = new Point3D (0,0,30);
        p1 = transfo ((Point3D)p1);
        p2 = transfo ((Point3D)p2);
        p3 = transfo ((Point3D)p3);
        p4 = transfo ((Point3D)p4);
        drawLine (p1, p2);
        drawLine (p1, p3);
        drawLine (p1, p4);
        drawLine (p2, p3);
        drawLine (p2, p4);
        drawLine (p3, p4);
    }
```