Correction du partiel IF 122 de vendredi 22 avril 2005

1 Le jeu de dame

Exercice 1

```
public mod_1() {
    tab = new int[32];

    for(int i=0; i<12; i++) tab[i] = 2;
    for(int j=12; j<20; j++) tab[j] = 0;   // cette ligne n'est pas nécessaire
    for(int k=20; k<32; k++) tab[k] = 1;
}
```

Exercice 2

```
public int coordonnees(char c, int n) {
    // on suppose que les coordonnées rentrées sont celles d'une case valide
    int base = 0;
    switch(c) {
        case 'a' : case 'b' : base = 0; break;
        case 'c' : case 'd' : base = 1; break;
        case 'e' : case 'f' : base = 2; break;
        case 'g' : case 'h' : base = 3; break;
    }
    return (base + 4*(8-n));
}
```

Exercice 3

```
public mod_2() {
    tab = new int[8][4];

    for(int i=0; i<8; i++) {
        for(int j=0; j<4; j++) {
            if (i<3) tab[i][j] = 2;
            else if (i>=5) tab[i][j] = 1;
            else tab[i][j] = 0;             // cette ligne n'est pas nécessaire
        }
    }
}
```

Exercice 4

```
public int [][] coordonnees(char c, int n) {
    // on suppose que les coordonnées rentrées sont celles d'une case valide

    int [] t = new int[2];   /* la case 0 représente l'abscisse
                                et la case 1 l'ordonnée
                             */

    switch(c) {
        case 'a' : case 'b' : t[0] = 0; break;
        case 'c' : case 'd' : t[0] = 1; break;
        case 'e' : case 'f' : t[0] = 2; break;
        case 'g' : case 'h' : t[0] = 3; break;
    }

    t[1] = 8-n;
    return t;
}
```

Exercice 5

```
public void affiche() {
    boolean decalage = true;

    Deug.println("-----------------");

    for(int i=0; i<32; i++) {
        if (tab[i] == 0) Deug.print("| | ");

        else if (tab[i] == 1) {
            if (decalage) Deug.print("| |N");
            else Deug.print("|N| ");
        }

        else {
            if (decalage) Deug.print("| |B");
            else Deug.print("|B| ");
        }

        if (i%4 == 3) {
            Deug.println("|");
            Deug.println("-----------------");
            decalage = ! decalage;
        }
    }
}
```

Exercice 6

```
public void affiche() {
    Deug.println("-----------------");

    for(int i=0; i<8; i++) {
        for(int j=0; j<4; j++) {

            if (tab[i][j] == 0) Deug.print("| | ");

            else if (tab[i][j] == 1) {
                if (i%2 == 0) Deug.print("| |N");
                else Deug.print("|N| ");
            }

            else {
                if (i%2 == 0) Deug.print("| |B");
                else Deug.print("|B| ");
            }
        }

        Deug.println("|");
        Deug.println("-----------------");
    }
}
```

2 Opérations sur les ensembles

Exercice 7

```
public static boolean estVide (Ensemble l) {
    return (l == null);
}
```

Exercice 8

```
public static void affiche (Ensemble l) {
    if (estVide(l)) Deug.println();
    else {
        Deug.print(l.valeur + " ");
        affiche(l.suivant);
    }
}
```

Exercice 9

```
public static int nombreElement (Ensemble l) {
    if (estVide(l)) return 0;
    else return (1 + nombreElement(l.suivant));
}
```

Exercice 10

```
public static boolean estElement (Ensemble l, int a) {
    if (estVide(l)) return false;
    else {
        if(l.valeur == a) return true;
        else return estElement(l.suivant, a);
    }
}
```

Exercice 11

On suppose que le constructeur suivant existe :

```
public Ensemble(int v, Ensemble s) {
    valeur = v;
    suivant = s;
}
```

Ensuite on définit la méthode `ajoute` :

```
public static Ensemble ajoute (Ensemble l, int v) {
    if (estVide(l)) return new Ensemble(v,null);
    else return new Ensemble(l.valeur, ajoute(l.suivant, v));
}
```

Exercice 12

```
public static Ensemble intersection (Ensemble l, Ensemble s) {
    if (estVide(l)) return null;
    else {
        if (! estElement(s,l.valeur)) return intersection(l.suivant,s);
        else return new Ensemble(l.valeur, intersection(l.suivant,s));
    }
}
```

Exercice 13

```
public static Ensemble union (Ensemble l, Ensemble s) {
    if (estVide(l)) return s;
    else {
        if (estElement(s,l.valeur)) return union(l.suivant,s);
        else return new Ensemble(l.valeur, union(l.suivant,s));
    }
}
```

Exercice 14

```
public static Ensemble soustraction (Ensemble l, Ensemble s) {
    if (estVide(l)) return null;
    else {
        if (estElement(s,l.valeur)) return soustraction(l.suivant,s);
        else return new Ensemble(l.valeur, soustraction(l.suivant,s));
    }
}
```

Exercice 15

```
public static void main (String [] args) {
    Ensemble l1 = new Ensemble(1, new Ensemble(4, new Ensemble(2,null)));
    Ensemble l2 = new Ensemble(2, new Ensemble(7, new Ensemble(6,null)));

    afficher(l1);
    afficher(l2);

    Ensemble u = union(l1,l2);
    Ensemble i = intersection(l1,l2);
    Ensemble s = soustraction(l1,l2);

    afficher(u);
    afficher(i);
    afficher(s);
}
```