

# Programmation

Mickaël Foursov – L3 Info Rennes 1

2015 – 2016, S1

## TP1

Sujet : tri de boules Vertes, Blanches et Rouges (tableau de `char` en Java).

```
1 public class Lecture {
2     public static InputStream ouvrir(String path);
3     public static boolean finFichier(InputStream f);
4     public static void fermer(InputStream f);
5     public static char lireChar(InputStream f);
6     public static String lireString(InputStream f);
7     public static int lireInt(InputStream f) throws NumberFormatException;
8     public static int lireIntln(InputStream f) throws NumberFormatException;
9     public static double lireDouble(InputStream f) throws NumberFormatException;
10    public static double lireDoubleln(InputStream f) throws NumberFormatException;
11    public static String lireUnite(InputStream f, boolean lire);
12 }
13 public class Ecriture {
14     public static OutputStream ouvrir(String path);
15     public static void fermer(OutputStream f);
16     public static void ecrireChar(OutputStream f, char c);
17     public static void ecrireString(OutputStream f, String s);
18     public static void ecrireStringln(OutputStream f, String s);
19     public static void ecrireStringln(OutputStream f);
20     public static void ecrireInt(OutputStream f, int n);
21     public static void ecrireInt(OutputStream f, int n, int times);
22     public static void ecrireDouble(OutputStream f, double d);
23 }
```

Toutes les méthodes sont surchargées par un équivalent pour l'entrée / sortie standard. L'algo de tri est donné, il faut juste corriger quelques erreurs pour respecter les invariants donnés aussi.

## TP2

Ce TP consiste à trouver un entier inférieur à 100 tel que son carré et son cube contiennent une seule fois chaque chiffre. Pas d'API fournie, faut juste faire un petit algo. Je crois qu'il y avait une autre partie mais elle était chiant, osef.

## TP3

Premier exo : suite des Fourmis (1, 11, 21, 1211, 111221, ...), un peu de parsing et de comptage. Deuxième exo : classement d'entiers entrés en ligne de commande, par insertion. Troisième exo : classement de paires d'entiers, un bon gros copier-coller.

## TP4

```
1 public class List<T extends SuperT> {
2     public List();
3     public ListIterator iterator();
4     public boolean isEmpty();
5     public void clear();
6     public void setFlag(T);
7     public void addHead(T);
8     public void addTail(T);
9     public List<T> clone();
10    public String toString();
11 }
12 public interface Iterator<T> {
13     public abstract void goForward();
14     public abstract void goBackward();
15     public abstract void restart();
16     public abstract boolean isOnFlag();
17     public abstract void remove();
18     public abstract T getValue();
19     public abstract T nextValue();
20     public abstract void addLeft(T);
21     public abstract void addRight(T);
22     public abstract void setValue(T);
23     public abstract String toString();
24 }
```

## TP6

```
1 public class BinaryTree<T> {
2     public BinaryTree();
3     public TreeIterator iterator();
4     public boolean isEmpty();
5     public int height();
6     public String toString();
7 }
8 public interface Iterator<T> {
9     public abstract void goLeft();
10    public abstract void goRight();
11    public abstract void goUp();
12    public abstract void goRoot();
13    public abstract boolean isEmpty();
14    public abstract NodeType nodeType();
15    public abstract void remove();
16    public abstract void clear();
17    public abstract T getValue();
18    public abstract void addValue(T);
19    public abstract void setValue(T);
20    public abstract void switchValue(int); // wtf is this I don't know
21 }
```