

Les bases du langage

Semestre d'hiver 2017/2018 – filière ITI

Objectifs

Ce laboratoire répond aux objectifs suivants:

- 1) Compilation, exécution avec ligne de commande en *shell*
- 2) Structures de contrôle : *for*, *while*, *if*, *break*, etc
- 3) Maîtrise de la syntaxe de base
- 4) Compréhension intuitive de la notion d'objet
- 5) Utilisation de l'API Java

Reddition

Le TP est à terminer pour le début du prochain laboratoire

I. Descriptif

Java est un environnement très riche et un langage tout usage.

Ce laboratoire vous permet d'apprendre les bases du langage Java en évitant le plus possible une approche objet pour l'instant !

II. Obligation

Vous devez toujours travailler en ayant accès à tout moment à l'API Java. Celle-ci est accessible en :

<http://download.oracle.com/javase/6/docs/api/>

III. Implémentations

Etape A : Hello World

On commence par un Hello World dans l'environnement de développement *Eclipse* :

Créer *Nouveau projet Tp1a* puis *Nouvelle classe Tp1a* ;

Copiez le programme suivant :

Programme 1 :

```
//Tp1a.java1
public class Tp1a {
    public static void main(String[] args) {
        System.out.println("Hello World !!!");
    }
}
```

Compilation et exécution avec le menu '**run**' ou la flèche > (6^{ième} icône)

¹ D'un programme au suivant, les nouveautés essentielles sont graisées et les questions s'y rapportent souvent

Dans un programme Java, les commentaires se font comme en C/C++ avec `//` pour une ligne et en entourant avec `/*` et `*/` un bloc qu'on souhaite commenter.

Etape B : écho d'arguments

Modifier le programme Tp1b comme suit :

Programme 2 Hello World avec echo

```
//TP1b.java2
public class TP1b {
    public static void main(String[] args) {
        System.out.println("Hello World !!!");
        System.out.println("Mon nom est: " + args[0]);
    }
}
```

Compiler puis entrer les arguments (par exemple vos nom prénom ...) de votre programme à l'aide du menu d'eclipse
=> *run* > *run configurations* > *onglet argument*.
Exécuter celui-ci.

On reprend ensuite le programme précédent pour afficher tous les arguments de la ligne de commande.

Programme 3 Hello World avec echo de tous les arguments

```
//TP1b.java
public class TP1a {
    public static void main(String[] args) {
        System.out.println("Hello World !!!");
        System.out.print("Mon nom est: ");
        for (int i=0;i<args.length;i++) {
            System.out.print(args[i]+" ");
        }
        System.out.println();
    }
}
```

Fig 3 : Commandes de base pour programme avec affichage de tous les arguments

Compiler et exécuter celui-ci.

QUESTIONS :

1. Que se passe t-il si vous entrez les arguments suivant : « il fait beau » Pourquoi ?
2. Citez quelques mots réservés du langage Java ?
3. Que signifie `String[]` ?

Etape C : structures de contrôle

Taper, compiler et exécuter le programme suivant :

```
public class TP1c {
    public static void main(String[] args) {
        int x = (int) (Math.random()*10); // x est un entier entre 0 et 10
        int y = (int) (Math.random()*10);
        System.out.println(" au debut x et y valent ....");
    }
}
```

```

        System.out.println(" x=" + x + " y=" + y);
        while ( x < 5) {
            if ( y < 5 ) {
                x++ ; // x = x +1
            }
            else {
                x--;
                y++;
            }
            System.out.println(" x=" + x + " y=" + y);
        }
    }
}

```

Fig 4 : Exemple d'utilisation des structures de contrôles *if*, *else* et *while*

Etape D : et si nous lisions au clavier ?

Voici le code suivant :

```

//clavier.java

import java.util.Scanner;

public class Clavier {
    public static Scanner in = new Scanner(System.in);
}

```

Créer ce fichier depuis eclipse, dans votre projet. Nommer le Clavier.java. Enregistrer celui-ci dans votre projet en cours. Ecrire et exécuter alors le programme suivant :

Programme Tp1d : lecture au clavier

```

//TP1d.java

public class TP1d {
    public static void main(String[] args) {
        System.out.println("Alors qu'avez-vous à me dire:");
        String phrase = Clavier.in.next(); // pour lire un String
        System.out.println(phrase);
    }
}

```

Fig 5 : exemple de lecture d'un String avec la classe Clavier

Introduisons maintenant une répétition de lecture et aussi une détermination d'une condition d'arrêt alternative.

Programme TP1d2 Détection d'une fin de fichier ou d'un mot de rupture

```

//TP1d2.java
public class TP1d2 {
    public static void main(String[] args) {
        System.out.println("Alors qu'avez-vous à me dire:");
        while(true) {
            String phrase = Clavier.in.next();
            System.out.println(phrase);
            if (phrase==null || phrase.equals("quit")) break;
            System.out.println("Alors qu'avez-vous à me redire:");
        }
    }
}

```

QUESTIONS :

4. Citer et décrire tous les opérateurs du programme précédent ?
5. Citer et commenter les instructions de programmation ?

Etape E : Comment « mal » programmer en java

1. Ecrire un petit programme qui calcule l'aire d'un disque et en affiche la valeur à l'écran. Dans un premier temps, initialiser la valeur du rayon directement dans le programme. Ensuite, implémenter une fonction d'entête : **public static** double aire(double r);

Finalement, incorporer dans votre programme une saisie au clavier de la valeur du rayon à l'aide de l'appel

```
Clavier.in.nextDouble() ;
```

2. Compléter le programme suivant en implémentant une fonction qui effectue la multiplication de deux matrices.

```
//Matrices.java
import java.util.*;

public class Matrices {
    public static void main(String[] args) {
        int[][] A = new int[4][2];
        int[][] B = new int[4][2];
        randInit(A,0,10);
        randInit(B,0,10);
        println(A);
        println(B);
        println(mult(A,B));
    }

    public static void randInit(int[][] M,int inf,int sup) {
        Random alea = new Random();
        for (int i=0;i<M.length;i++)
            for (int j=0;j<M[0].length;j++)
                M[i][j] = inf + alea.nextInt(sup-inf);
    }

    public static void println(int[][] M) {
        // à compléter
    }

    public static int[][] mult(int[][] A, int[][] B) {
        // à compléter
    }
}
```

Etape F : un HelloWorld graphique...

```
//HelloWin.java
import java.awt.*;
public class HelloWin {
    public static void main(String[] args) {
        Frame f = new Frame("HelloWin");
        f.setSize(200,200);
        f.setLocation(100,100);
        f.setBackground(Color.pink);
        f.setVisible(true);
    }
}
```

Ensuite en utilisant l'appel `Thread.sleep(1000)` pour une attente passive de 1 seconde, décaler pendant 10 secondes la fenêtre de 10 pixels par seconde suivant les axes horizontaux et verticaux. Aidez-vous de l'API Java !