

De Pascal à Java

Il est possible de programmer en Java d'une manière proche de la programmation impérative qu'est celle du Pascal, même si dans la suite du cours, le style de programmation **diffère** complètement.

Les « programmes » pascal et java peuvent être écrits avec des éditeurs de textes du plus basique au plus sophistiqué. Afin de tester les programmes vus dans ce premier TD, Voici les différences notables :

Pascal	Java
program machin ;	public class Machin{ }
Cette ligne n'est pas obligatoire. Pas de différence Majuscule/minuscule	Convention : Majuscule pour les noms de classe Tous les identifiants sont sensible à la casse
Enregistrer sous	
machin.pas (ou Machin.pas)	Machin.java
Non compilable donc non exécutable	compilable mais non exécutable
“Programme” minimum (en toute rigueur, une classe n'est pas un programme (voir la suite du cours) mais on se contentera de cette approximation)	
program kedal ; Begin End .	public class Kedal { public static void main (String[] args){ } }
Compilation	
fpc kedal (fpcompile kedal ou kedal.pas)	javac Kedal.java (.java obligatoire)
Résultat de la compilation	
kedal (kedal.exe sous windows)	Kedal.class
Execution	
./kedal (kedal sous windows)	java Kedal Remarque : javac Machin.java → pas d'erreur java Machin → Exception in thread "main" java.lang.NoSuchMethodError: main
binaire exécutable par le SE/plateforme matérielle pour lequel il a été compilé	bytecode java exécutable sur tout SE/plateforme (il doit exister une JVM)
Utilisation de bibliothèques (ou packages)	
program machin ; uses sysutils,math ; ...	import java.util.*; import java.io.*; public class Machin { ... }
Déclaration de blocs	
begin ... end	{ ... }
Déclaration de constantes, types et de variables	
Par des sections commençant par const , type , var	Les seuls types fabriqués sont des classes (class) Pas de mot particulier pour les variables.
Syntaxe : <nom> : <type> ;	<type> <nom> ;
Avant le corps du programme, procédure ou fonction (avant bloc principal)	À l'intérieur de chaque bloc
Passage de paramètres	
Aux procédures et fonctions	Aux méthodes
4 modes : - in constant : (const <nom> : <type>) - in variable : (<nom> : <type>) - out : (out <nom> : <type>) - in/out : (var <nom> : <type>)	1 seul mode : in variable : (<type> <nom>) modifiable à l'intérieur, pas de répercussion à l'extérieur

Paramètres multiples	
De types différents : ... (a : integer ; b : string) (String a , int b) ...
Du même type : ... (a,b : integer) (int a , int b) ...
Fonction	Méthode avec valeur de retour
<pre> program testFonction ; function f (x : real) : real; begin f := 2*x+3; end ; var a : real; Begin A := f(5); End . </pre>	<pre> public class TestFonction { public float f (float x){ return 2*x+3; } public static void main (String[] args) { TestFonction test = new TestFonction(); float a = test.f(5); } } </pre> <p>attention : contrairement à pascal, java est sensible à la casse (A et a ci-contre)</p>
Procédure	Méthode sans valeur de retour (« void »)
<pre> program testProc ; procedure p (x : real) ; begin // ce qu'on veut end ; Begin // le programme End . </pre>	<pre> public class TestProc { public void p (float x){ // ce qu'on veut } public static void main (String[] args) { // le programme } } </pre>

Le « writeln » exemple :

```

public class Essai2 {
  public static void main (String[] args){
    int a=7;
    {
      int x = 2;
      boolean fini = x < 0; // fin de boucle
      int nb = 0; // nb tours de boucle
      while ( ! fini ) {
        x = x-1;
        nb = nb + 1;
        fini = x < 0;
      }
      System.out.println(nb);
      System.out.println(a);
    }
  }
}

```

```

    double x=3.75;
    // System.out.println(nb);//erreur
    System.out.println(x);
    System.out.println(a);
  }
  System.out.println(a);
}

/*
a est visible partout
nb n'est visible que dans le premier bloc
il y a un x différent dans chaque bloc
Il faut impérativement 2 blocs, ici sinon la
re-déclaration de x donnerait une erreur
*/

```

Le « readln » (dans la pratique, on utilise une bibliothèque pour simplifier ce qui suit) :

```

import java.io.*;
public class LireClav {
  public static void main (String[] args) throws java.io.IOException{
    BufferedReader source = new BufferedReader(new InputStreamReader(System.in));
    System.out.println("tapez une chaine");
    String donnee = source.readLine();
    System.out.println(donnee);
    System.out.println("tapez un entier");
    int a = Integer.parseInt( source.readLine());
    System.out.println(a);
  }
}

```

Exercices : refaire ceux du S1 et début du S2 (passage de paramètre par valeur uniquement)

Attention : affectation = (et non :=) , comparaison == (et non =) chaînes **String** (et non **string**) entre "" (et non **entre** ' ') commentaires entre /* ... */ (et non {...} ou(* ... *)) entier **int** (et non **integer**) etc. .