

Cours de Programmation Orientée Objet JAVA

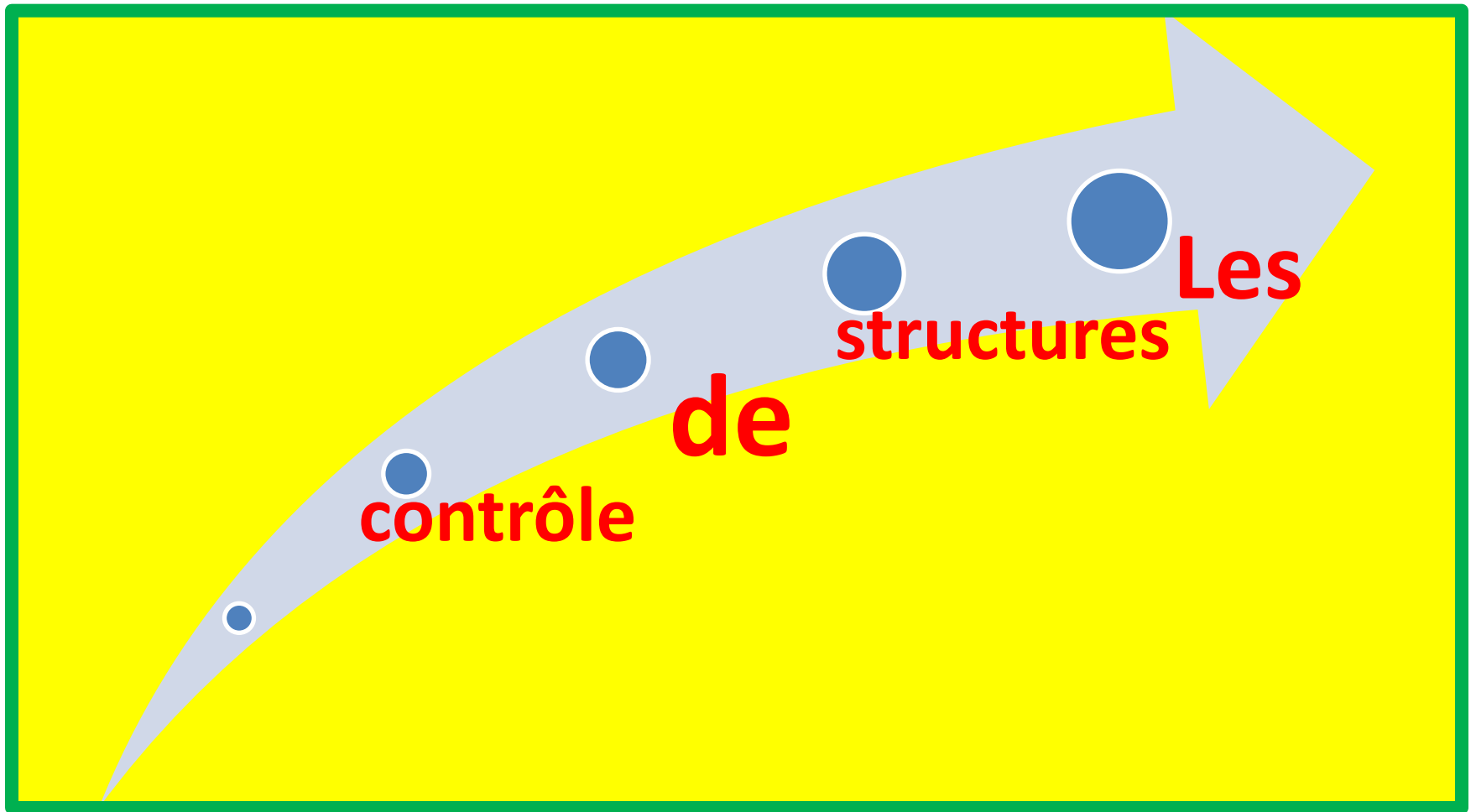
Demba SOW

*Docteur en
Mathématiques et en
Cryptologie*

L.A.C.G.A.A.

F.S.T. / U.C.A.D.

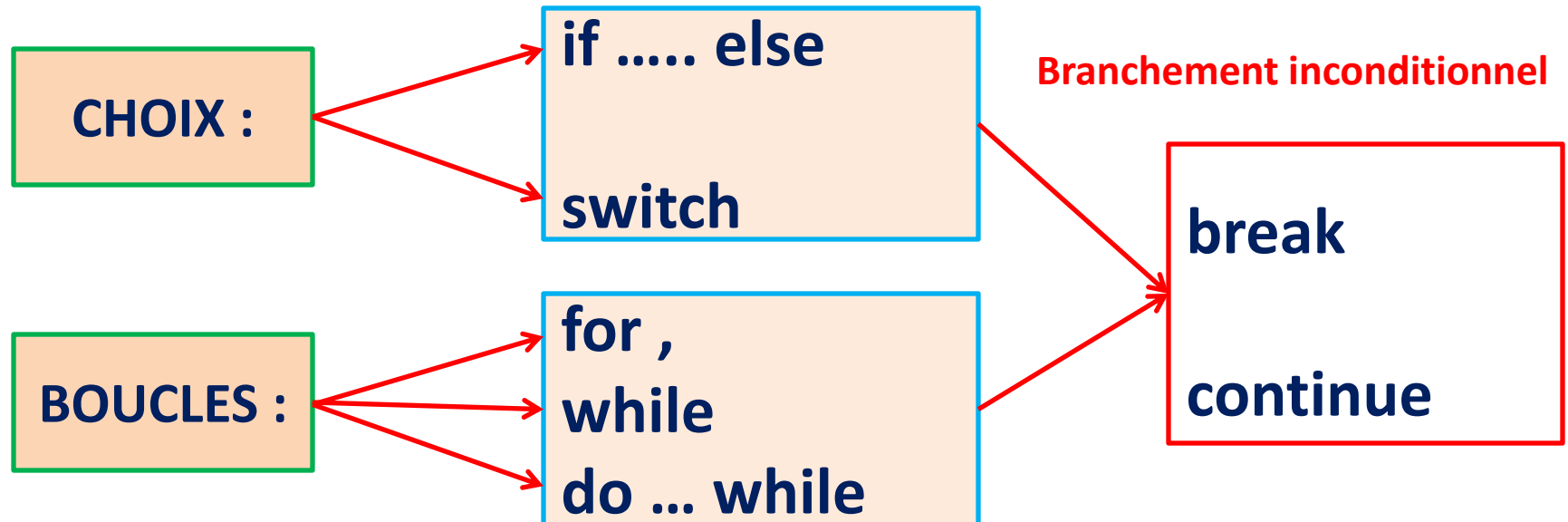


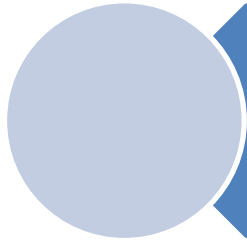


Les structures de contrôle

Les instructions d'un programme (**main**) sont à priori exécutées de façon séquentielle.

Les instructions de contrôle permettent de s'affranchir de cet ordre pour effectuer des **choix** et des **boucles**.





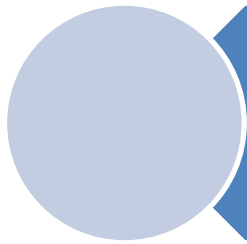
Choix :

if ...else

switch

```
package essai;
public class Exemple_If_Else{
int final MIN = 100;
int final Max = 1000;
int solde ;
public static void main(String args [ ])
{ if ( solde < MIN)
    System.out.println("solde insuffisant" ) ;
  else
    if (solde == MAX)
      System.out.println("solde suffisant" ) ;
}
```

```
package essai ;
public class Exemple_Switch{
int final MIN = 100;
int final Max = 1000;
int choix , solde;
public static void main(String args [ ])
{ switch(choix)
  { case 1: solde = MIN;
    System.out.println("solde insuffisant" ) ;
    break;
    case 2: solde = MAX ;
    System.out.println("solde suffisant" ) ;
    break;
    default : break
  } }}
```



Syntaxe:

if ...else

switch

if (condition)

instruction_1

[else

instruction_2]

Condition booléenne (true / false)

Expressions quelconques

Les **crochets** renferment des instructions facultatives.

switch (expression)

{ case constante_1 : [suite_d'instruction_1]

case constante_2 : [suite_d'instruction_2]

.....

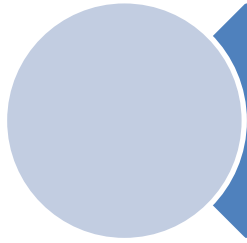
case constante_n : [suite_d'instruction_n]

default : [suite_d'instructions]

}

Expression de type **byte**, **char**, **short** ou **int** .

Expression **constante** d' un type compatible par affectation avec le type de **expression**



L'instruction do while

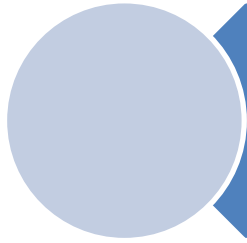
```
package essai ;
public class Exemple_Do_While{
public static void main(String args [ ])
{ do
  { System.out.println("saisir un entier strictement positif " ) ;
    n = Clavier.lireInt( ) ; // saisir à partir du clavier
    if ( n < 0) System.out.println("la saisie est invalidée: recommencez" ) ;
  } while ( (n < 0) || (n == 0) );
}
}
```

do instruction

Expression quelconque

Condition booléenne

while (condition) ;



L'instruction while

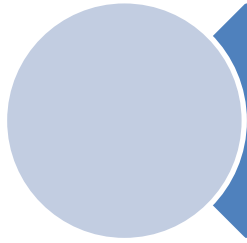
```
package essai ;  
public class Exemple_While{  
    public static void main(String args [ ] )  
    {while ( n <=0)  
        { System.out.println("saisir un entier strictement positif " ) ;  
          n = Clavier.lireInt( ) ; // saisir à partir du clavier  
          if ( n < 0) System.out.println("la saisie est invalidée: recommencez" ) ;  
        }  
    }  
}
```

while (condition)

instruction

Condition booléenne

Expression quelconque



L'instruction for

```
package essai ;  
public class Exemple_For{  
    public static void main(String args [ ])  
    {   int tab [ ] = new int [ 100] ; // tableau d'entiers de taille 100  
        for( int i = 0 ; i < 100 ; i ++ )  
        {  
            tab [ i ] = i + 1;  
        }  
    }  
}
```

for ([initialisation] ;[condition] ; [incrémentation])

instruction

Branchement inconditionnel

break / continue

Ces instructions s'emploient principalement au sein des boucles.

break

Elle sert à interrompre le déroulement de la boucle, en passant à l'instruction suivant la boucle.

```
package essai ;
public class Exemple_Break {
public static void main(String args [ ])
{ int tab [ ] = new int [ 10] ; // tableau d'entiers de taille 10
  for( int i = 0 ; i < 10 ; i ++ )
    { if ( i == 5 ) break ; // initialiser seulement les 5 premiers elts du tableau
      tab [ i ] = i + 1 ;
    } // ← le break nous branche à la sortie du for pour continuer
  for( int i = 0 ; i < 10 ; i ++ )
    System.out.println(" éléments du tableau:" + " "+tab [ i] );
}
}
```

éléments du tableau: 1 2 3 4 5 0 0 0 0 0

break avec imbrication de boucles

```
package essai;
public class Essai_Break_Imbr {
public static void main(String args [ ])
{ int tab [ ] = new int [ 10] ; // tableau d'entiers de taille 100
  for( int i = 0 ; i < 10 ; i ++ )
  {
    for ( int j =i;j <10;j++)
      { if ( j == 3 || j == 4 || j == 5 ) break ;
        tab [ j ] = j+1;
      } // ← le break branche ici
  }
  for( int i = 0 ; i < 10 ; i ++ )
    System.out.println(" éléments du tableau:" + " "+tab [ i ] );
}
```

éléments du tableau: 1 2 3 0 0 0 7 8 9 10

En cas de boucles imbriquées, l'instruction break fait sortir uniquement de la boucle la plus interne.

break avec étiquette

```
package essai;
public class Essai_Break_Etiq {
public static void main(String args [ ])
{ int tab [ ] = new int [ 10]; // tableau d'entiers de taille 100
repeat: for( int i = 0 ; i < 10 ; i ++ )
{
    for ( int j =i;j <10;j++)
    { if ( j == 3 || j == 4 || j == 5 ) break repeat;
      tab [ j ] = j+1;
    }
} // ← cette fois le break branche ici
for( int i = 0 ; i < 10 ; i ++ )
    System.out.println(" éléments du tableau:" + " "+tab [ i] );
}
}
```

Étiquette : pour sortir
de deux boucles
imbriquées


éléments du tableau: 1 2 3 0 0 0 0 0 0 0

Continue ordinaire

continue

L'instruction continue permet de passer *prématurément* au tour de boucle suivant.

```
package essai ;  
public class Exemple_Continue_Ord{  
    public static void main(String args [ ])  
    { int tab [ ] = new int [ 10] ; // tableau d'entiers de taille 10  
      for( int i = 0 ; i < 10 ; i ++ ) // ← ici  
      { if ( i == 5 ) continue ; // on poursuit la boucle for  
        tab [ i ] = i + 1 ;  
      }  
      for( int i = 0 ; i < 10 ; i ++ )  
        System.out.println(" éléments du tableau:" + " "+tab [ i ] );  
    }  
}
```



éléments du tableau: 1 2 3 4 5 0 7 8 9 10

Continue avec étiquette

```
package essai;
public class Essai_Continue_Etiq {
public static void main(String args [ ])
{ int tab [ ] = new int [ 10] ; // tableau d'entiers de taille 100
  again: for( int i = 0 ; i < 10 ; i ++ ) // ← cette fois le continue branche ici
  {
    for ( int j =i;j <10;j++)
      { if ( j == 3 || j == 4 || j == 5 ) continue;
        tab [ j ] = j+1;
      }
  }
  for( int i = 0 ; i < 10 ; i ++ )
    System.out.println(" éléments du tableau:" + " "+tab [ i ] );
}
```

ATTENTION: on ne mentionne pas le nom de l' étiquette avec continue.

éléments du tableau: 1 2 3 0 0 0 7 8 9 10

