

UEF 1 : Informatique & Programmation

Faculté des Sciences de Nice

DEUG 2001-2002

Jérôme DURAND-LOSE

Jean-Paul ROY

COURS 7

Itérations

Tant que ... répéter ...

Répéter ... tant que ...

7-2

Tâches répétitives

Tant que l'on y arrive pas
Réessayer

Tant que ce n'est pas la bonne solution
Chercher une autre solution

Tant que ce n'est pas fini
Continuer

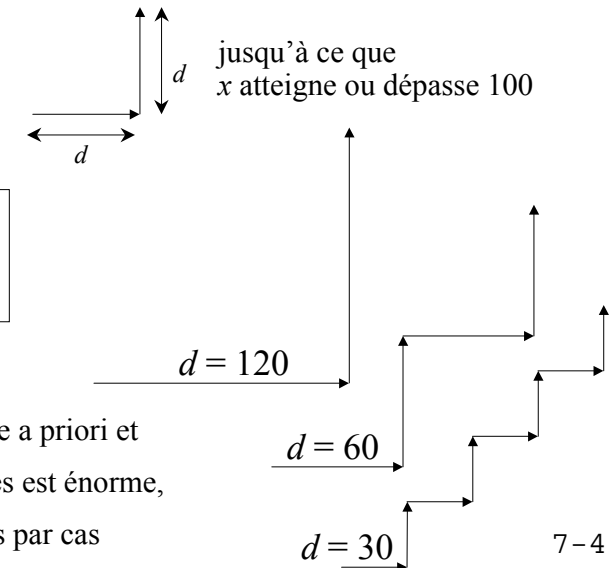
Demander l'entrée
Recommencer tant qu'elle est incorrecte

7-3

Dessins répétitifs

Problème
Faire des mouvements

```
t.left( 90 );  
t.forward( d );  
t.right( 90 );  
t.forward( d );
```



La variable d est inconnue a priori et
le nombre de cas possibles est énorme,
impossible de faire du cas par cas

7-4

Dessins répétitifs

Ce que l'on veut dire

```
Turtle t = new Turtle();
tant que t.xcoor() < 100 faire
    t.left( 90 );
    t.forward( d );
    t.right( 90 );
    t.forward( d );
fin tant que
```

Comment l'écrire
en Java

```
Turtle t = new Turtle();
while ( t.xcoor() < 100 ) {
    t.left( 90 );
    t.forward( d );
    t.right( 90 );
    t.forward( d );
}
```

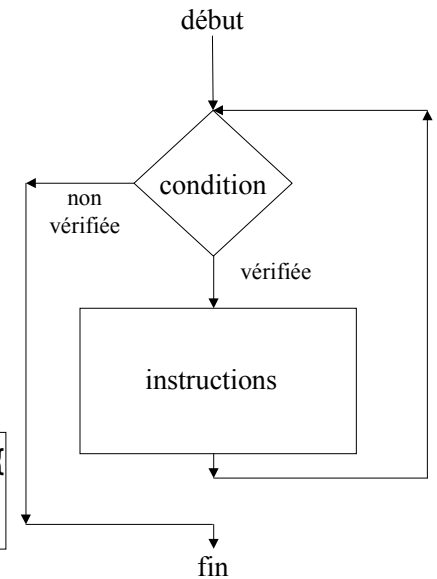
7-5

Schéma et syntaxe

Tant que condition faire
instructions
fin tant que

```
while ( <condition> ) {
    <instructions>
}
```

```
while ( /* précision insuffisante */ ) {
    /* affiner la précision */
}
```



7-6

Problème ▷ algorithme ▷ programme

« Avec un compte rémunéré à 2,5 %,
combien d'années faut-il pour doubler sa mise ? »

« Heu, 2,5 % par an,
cela fait 25 % en 10 ans,
il faut donc 40 ans ! »
(un « bon client »)

« Facile $\frac{\ln 2}{\ln 1,025} = 28,071$ »
(un actuaire)

7-7

Problème ▷ algorithme ▷ programme

Calcul de la nouvelle valeur

$$x' \leftarrow x + x * 2,5 \% = x * 1,025$$

Si j'ai x euros, à l'année 0

À l'année 1, j'ai $x * 1,025$ euros

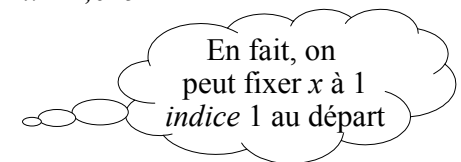
À l'année 2, j'ai $x * 1,025 * 1,025$ euros

À l'année 3, j'ai $x * 1,025 * 1,025 * 1,025$ euros

À l'année 4, j'ai $x * 1,025 * 1,025 * 1,025 * 1,025$ euros

....

calculons ces valeurs



7-8

Problème ▷ algorithme ▷ programme

Deux variables et leurs types :
 année => entier
 indice => nombre approché

Année	Indice
0	1,000000000000
1	1,025000000000
2	1,050625000000
3	1,076890625000
4	1,103812890625
...

Fin de l'affichage?

7-9

Problème ▷ algorithme ▷ programme

Constante

intérêt : nombre approché = 0,025 i.e. 2,5 %

coefficient : nombre approché = 1 + intérêt

Variables

nombre d'années : entier = 0

indice : nombre approché = 1

Algorithme pour trouver en combien d'années le capital est doublé

Tant que l'indice est plus petit que 2 faire

 indice ← indice multiplié par coefficient

 nombre d'années ← nombre d'années + 1

Afficher le résultat

7-10

*/** Affiche le nombre d'années qu'il faut pour doubler son capital avec 2,5 % d'intérêts **/*

class PourDoubler {

/ CONSTANTES */*

*/** Taux d'intérêt de base **/*

static final double INTERET = .025;

*/** Coefficient annuel correspondant à cet intérêt **/*

static final double COEFFICIENT_ANNUEL = 1 + INTERET;

/ LANCEMENT de la CLASSE */*

public static void main (String [] arg) {

int nombreAnnees = 0;

double indice = 1;

while (indice < 2) {

 indice = indice * COEFFICIENT_ANNUEL;

 nombreAnnees = nombreAnnees + 1;

 }

 System.out.println ("Il faut " + nombreAnnees

 + " années pour doubler son capital avec un intérêt de "

 + INTERET + ".");

 }

}

Constantes
en majuscule

7-11

import unsa.Console;

*/** Affiche le nombre d'années qu'il faut pour doubler son capital avec un taux demandé
* à l'utilisateur. Une trace de la boucle est faite, l'indice est affiché pour chaque année **/*

class PourDoublerAvecTrace {

public static void main (String [] arg) {

int nombreAnnees = 0; */* nombre d'années écoulées */*

double indice = 1; */* indice de référence 1 à l'année 0 */*

/ taux d'intérêt de base */*

double interet = Console.readDouble("Quel pourcentage d'intérêt ?");

/ coefficient annuel correspondant à cet intérêt */*

double coefficientAnnuel = 1 + interet / 100;

while (indice < 2) { *// tant que pas doublé*

/ mise à jour */*

 indice = indice * coefficientAnnuel;

 nombreAnnees = nombreAnnees + 1;

/ trace de la boucle */*

 System.out.println ("année : " + nombreAnnees+ " indice : " + indice);

 }

 System.out.println ("Il faut " + nombreAnnees

 + " années pour doubler son capital avec un intérêt de " + interet + ".");

 }

}

7-12

Voir le déroulement de la boucle

faire une « trace »,
pour avoir plus d'informations,
pour mieux comprendre,
pour mettre au point,
...

```
année : 1  indice : 1.025
année : 2  indice : 1.050625
année : 3  indice : 1.0768906249999999
année : 4  indice : 1.103812890624999
.....
année : 27 indice : 1.9478000182997102
année : 28 indice : 1.9964950187572028
année : 29 indice : 2.0464073942261325
```

Il faut 29 années pour doubler son capital avec un intérêt de 2.5%

7-13

Programme : du texte au sens

Que fait ce programme *totalelement illisible* ?

```
/* le programme compile et marche bien */
import unsa.Console;
class C{
public static void main(String[]aaaa){
int a=Console.readInt ( "" );
int aa=Console.readInt ( "" );
String aaa="";
do{ aaa=a%aa+aaa;//mise à jour
a=a/aa; }while(0<a);// tant que c'est positif
Console.println(aaa);//affichage
System.exit(0); } }
```

7-14

Programme plus lisible

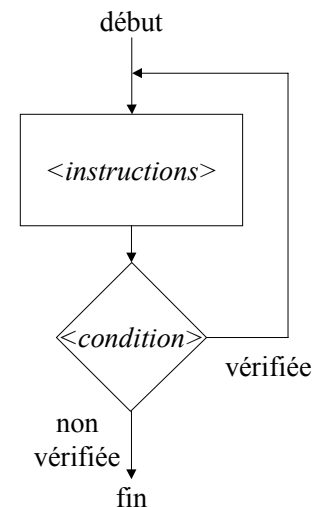
```
import unsa.Console;
class ClasseMystere {
    public static void main ( String[] args ) {
        int i = Console.readInt ( "Entrer un entier" );
        int j = Console.readInt ( "Entrer un entier" );
        String s = "";
        do {
            s = ( i % j ) + s;
            i = i / j;
        } while ( 0 < i );
        Console.println ( s );
        System.exit ( 0 );
    }
}
```

7-15

do ... while ... ?

Syntaxe :

```
do {
    <instructions>
} while ( <condition> );
```



Les instructions sont exécutées au moins une fois

7-16

Jeux de tests

i	j	s
	5	

i	j	s
	2	

On reconnaît un algorithme déjà connu !

7-17

Programme correctement présenté

```

/** affiche l'écriture d'un nombre dans une base entre 2 et 10 */
import unsa.Console;
class EcritureBase {
    public static void main ( String[] inutilisé ) {
        int nombre = Console.readInt ( "Nombre = ?" );
        int base = Console.readInt ( "Base = ?" );
        if ( ( 2 <= base ) && ( base <= 10 ) ) {
            String ecriture = "";
            do {                // extraction d'un « digit »
                ecriture = (nombre % base) + ecriture;
                nombre = nombre / base;
            } while ( 0 < nombre );
            Console.println ( ecriture );
        }
        System.exit(0);
    }
}

```

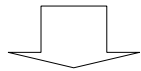
7-18

while ... et do ... while ... sont « équivalents »

```

while ( <condition> ) {
    <instructions>
}

```



```

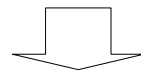
if ( <condition> ) {
    do {
        <instructions>
    } while ( <condition> );
}

```

```

do {
    <instructions>
} while ( <condition> );

```



```

<instructions>
while ( <condition> ) {
    <instructions>
}

```

7-19