

FICHE 5

LES STRUCTURES DE CONTROLE

(Fiches Java)

1 Les boucles

Les boucles en Java peuvent se faire de plusieurs façons. Elles sont calquées sur la syntaxe du C. Les compteurs éventuellement mis en place peuvent être déclarés **localement au bloc** qui définit la boucle. La portée de cette variable est alors restreinte au bloc de code interne à la boucle.

1.1 Les types de boucles

On dispose des types suivants :

- Boucle for
- Boucle While
- Boucle do ... while

1.2 Exemple

```
import java.io.*;

public class Pideux
{
    public static void main(String[] args)
        throws java.io.IOException
    {
        long      n = 1;
        double    S, S1, x=Math.PI;
        char      c='y';
        InputStreamReader sr = new InputStreamReader(System.in);
        BufferedReader in = new BufferedReader(sr);

        System.out.print("valeur de pi2 sur 6 : ");
```

```

System.out.println(x*x/6);

while (c=='y' || c=='Y')
{
    do
    {
        S=0.0;
        S1=0.0;
        System.out.print("Entrer le nombre de termes : ");
        try
        {
            n=Long.valueOf(in.readLine()).longValue();
        }
        catch (Exception e)
        {
            n=0;
        }
    }
    while (n<0);
    for (long i=1; i <=n; i++)
    {
        S=S+1.0/(i+0.0)/(i+0.0);
    }
    for (long i=n; i > 0; i--)
    {
        S1=S1+1.0/(i+0.0)/(i+0.0);
    }
    System.out.print("n = ");
    System.out.print(n);
    System.out.print("      S= ");
    System.out.println(S);
    System.out.print("      S1= ");
    System.out.println(S1);
    System.out.println();
    System.out.print("Un autre calcul?(Y/N) ");
    try
    {
        c=in.readLine().charAt(0);
    }
    catch (Exception e)
    {

```

```

        c='n';
    }
}
}
}

```

2 Les conditions

2.1 Les types de conditions

- Condition if
- Condition if ... else
- Condition switch

2.2 Exemple

```

package rr.modulog.syntaxe;

public class ArbreDeFonction{
    public byte          opertype;          // type de l'operateur (0..6)
    public char          signe;             // Si type=0,4
    public short         fonctcode;        // Si type=1
    public short         varcode;          // Si type=2
    public double        constante;        // Si type=3
    public char          separateur;       // Si type=5
    public short         constform;        // Si type=6
    public ArbreDeFonction oper1;          // Si type=0,1,4,5
    public ArbreDeFonction oper2;          // Si type=0,5

    public ArbreDeFonction(){
    }

    public ArbreDeFonction(byte b, char c, short fc, short vc, double ct, char sep,
        short ctf, ArbreDeFonction P1, ArbreDeFonction P2) {
        this();
        opertype=b;
        signe=c;
    }
}

```

```
fonctcode=fc;
varcode=vc;
constante=ct;
separateur=sep;
constform=ctf;
oper1=P1;
oper2=P2;
}
```

```
/******
```

```
public void imprime(int niv){
    switch(this.opertype){
        case 0:
            for (short i=1; i <= niv; i++)
                System.out.print('-');
            System.out.println(signe);
            if (oper1!=null){
                oper1.imprime(niv+1);
            }
            else{
                for (short i=1; i <= niv+1; i++)
                    System.out.print('-');
                System.out.println("nil");
            }
            if (oper2!=null){
                oper2.imprime(niv+1);
            }
            else{
                for (short i=1; i <= niv+1; i++)
                    System.out.print('-');
                System.out.println("nil");
            }
            break;
        case 1:
            for (short i=1; i <= niv; i++)
                System.out.print('-');
            System.out.println(fonctcode);
            if (oper1!=null){
                oper1.imprime(niv+1);
            }
    }
}
```

```

    }
    else{
        for (short i=1; i <= niv+1; i++)
            System.out.print('-');
        System.out.println("nil");
    }
    break;
case 2:
    for (short i=1; i <= niv; i++)
        System.out.print('-');
    System.out.print("Var ");
    System.out.println(varcode);
    break;
case 3:
    for (short i=1; i <= niv; i++)
        System.out.print('-');
    System.out.println(constante);
    break;
case 4:
    for (short i=1; i <= niv; i++)
        System.out.print('-');
    System.out.println(signe);
    if (oper1!=null){
        oper1.imprime(niv+1);
    }
    else{
        for (short i=1; i <= niv+1; i++)
            System.out.print('-');
        System.out.println("nil");
    }
    break;
case 5:
    for (short i=1; i <= niv; i++)
        System.out.print('-');
    System.out.println(separateur);
    if (oper1!=null){
        oper1.imprime(niv+1);
    }
    else{
        for (short i=1; i <= niv+1; i++)
            System.out.print('-');

```

```

        System.out.println("nil");
    }
    if (oper2!=null){
        oper2.imprime(niv+1);
    }
    else{
        for (short i=1; i <= niv+1; i++)
            System.out.print('-');
        System.out.println("nil");
    }
    break;
case 6:
    for (short i=1; i <= niv; i++)
        System.out.print('-');
    System.out.print("Const ");
    System.out.println(constform);
    break;
}
}
}

```