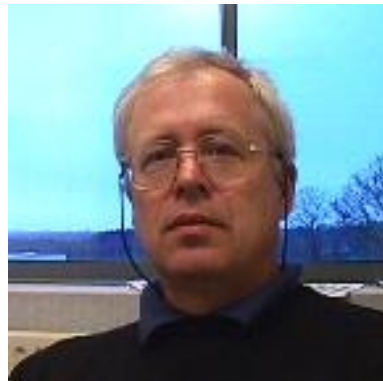


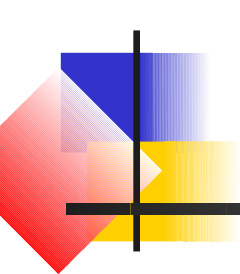
Programming with Java

conditions, loops and arrays

P. Bakowski



bako@ieee.org



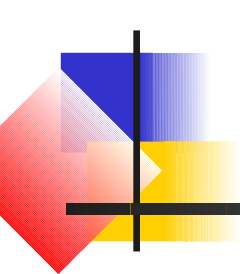
Conditions

A conditional statement (simple) has the following form:

```
if (condition) { instructions1 }  
else { instructions2 }
```

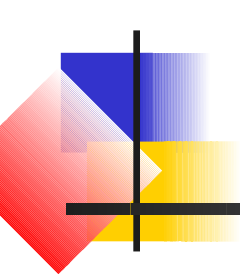
A conditional statement (multiple) has the following form:

```
if (condition0){ inst0 }  
else if (condition1){ inst1 }  
...  
else if (conditionn){ instn }  
else { instfalse }
```



Conditions : class Taxes

```
public class Taxes{
public static void main(String[] args) {
double pHT, pTTC;
int t;
System.out.println("Give the price without VAT: ");
pHT = Keyboard.readDouble();
System.out.println("Give the rate (normal->0 reduced->1)");
t = Keyboard.readInt();
if(t==0){ pTTC = pHT + (pHT*0.196); }
else { pTTC = pHT + (pHT*0.05); }
System.out.println("The price with VAT : "+ pTTC);
}
}
```



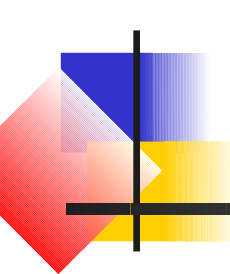
The loop : for

```
public class Rectangle {  
    public static void main(String[] args){  
        System.out.println("*****");  
        System.out.println("*****");  
        System.out.println("*****");  
        System.out.println("*****");  
        System.out.println("*****");  
    }  
}
```

```
public class Rectangle {  
    public static void main(String[] args){  
        for(int i=0;i<5;i=i+1) {  
            System.out.println("*****"); }  
    }  
}
```



Repeats *i* times the instructions in the braces, for *i* ranging from 0 to 4

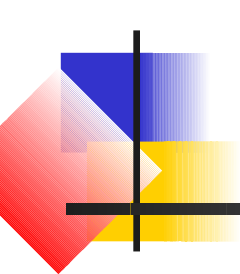


The loop : for (cont. Rectangle)

```
public class Rectangle {  
    public static void main(String[] args){  
        for(int i=0;i<5;i=i+2) {  
            System.out.println("****"); }  
    }  
}
```

```
public class Rectangle {  
    public static void main(String[] args){  
        for(int i=0;i<=5;i=i+2) {  
            System.out.println("****"); }  
    }  
}
```

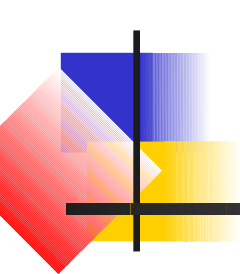
```
public class Rectangle {  
    public static void main(String[] args){  
        for(int i=0;i==5;i=i+2) {  
            System.out.println("****"); }  
    }  
}
```



The loop : for Rectangle2

```
public class Rectangle2 {  
    public static void main(String[] args){  
        int l;  
        System.out.println("how many lines of stars?");  
        l=Keyboard.readInt();  
        for(int i=0;i<l;i=i+1) {  
            System.out.println("****"); }  
    }  
}
```

```
Java/Essais>java Rectangle2  
how many lines of stars?  
6  
****  
****  
****  
****  
****  
****
```



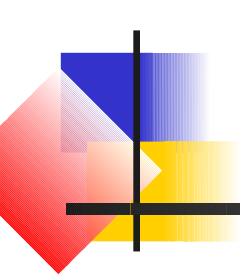
The loop : while

Sometimes, the **for** loop is not enough. This happens when we can not in advance determine the number of steps.

For this, we have the **while** loop; it has the following form:

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

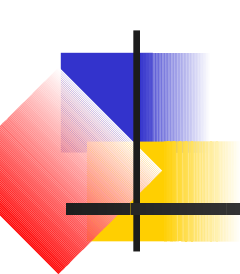
```
while(testResponse==false)  
{  
System.out.println("Give the rate (normal->0 reduced->1)");  
t = Keyboard.readInt();  
if (t==0||t==1){ testResponse = true ; }  
else { testResponse = false ;}  
}
```



The loop : while

```
public class Taxes2 {
public static void main(String[] args){
double pHT,pTTC ;
int t =0;
boolean testResponse = false;
System.out.println("Give the price without VAT: ");
pHT = Keyboard.readDouble();
while(testResponse==false){
    System.out.println("Give the rate (normal->0
                                reduced->1)");

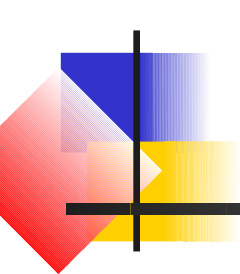
    t = Keyboard.readInt();
    if(t==0 || t==1){testResponse=true;}
    else { testResponse=false;}
}
if (t==0){ pTTC=pHT+(pHT*0.196); }
else { pTTC=pHT+(pHT*0.05); }
System.out.println("The price with VAT :"+pTTC);
}
}
```

Termination of loop : while

```
public class b1 {  
    public static void main(String[] args){  
        int i =1;  
        while(i!=10){  
            System.out.println(i +",");  
            i = i+1 ;  
        }  
    }  
}
```

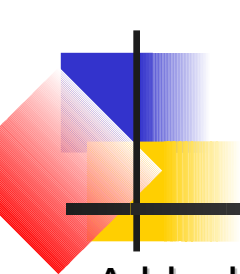
```
public class b2 {  
    public static void main(String[] args){  
        int i =1;  
        while(i!=10){  
            System.out.println(i +",");  
            i = i+2 ;  
        }  
    }  
}
```



The loop : do-while

The reason of a **do-while** loop is to run at least once the instructions in the body of loop before testing the **exit** condition.

```
total=0;
do
{
System.out.println("Give an integer(end with 0):");
n = Keyboard.readInt();
total = total + n;
}
while(n!=0);
```



The notion of block

A block is a sequence of instructions contained in braces. Thus, the body of a loop is a block, the statements between the braces of an **if**, an **else**, **else if** is a block. The statements between the braces of the **hand** also form a block.

```
public static void main(String[] args)
{ /*begin of block 1*/
    int a=2;
    System.out.println("value of a:"+a);
    if(a==0){ /* begin of block 2 */
        int b=3+a;
        System.out.println("value of b:"+b);
    } /* end of bloc 2 */
    else { /* begin of block 3 */
        int c=3+a;
        System.out.println("value of c:"+c);
    } /* end of block 3 */
    System.out.println("value of a:"+a);
} /* end of block 1 */
}
```

Nested loops

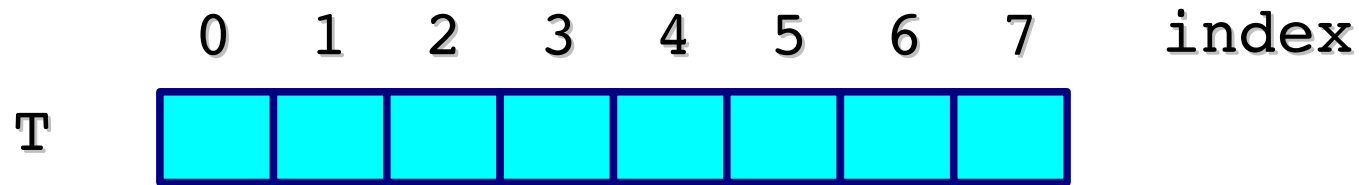
When there is a loop within a loop it is called **nested loops**.
Here is an example.

```
public class ExImb{  
    public static void main(String[] args {  
        for(int i=0;i<5;i=i+1)  
        {  
            for(int j=0;j<8;j=j+1)  
            {  
                System.out.println(j);  
            }  
            System.out.println(«\n»);  
        }  
    }  
}
```

```
Java/Essais>java ExImb  
01234567  
01234567  
01234567  
01234567  
01234567
```

Arrays

A table is a structure involving several values of the same type, called components of the table.



`T[2]` – component

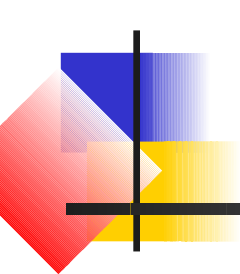
`T[0],T[1],T[2],T[3],T[4],T[5],T[6],T[7]`

Declaration :

```
T[] tab;
```

Creation :

```
tab = new T[size] ; // for example 8
```



Arrays: declaration and creation

Declaration :

```
int [] tab;  
double [] t;  
String [] m;  
tab[0] = 5;    // error
```

Allocation :

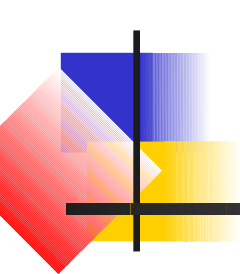
```
tab = new int[4];    // 0 – default value
```

Initialisation :

```
tab[0]=1; tab[1]=9; tab[2]=2; tab[3]=4;
```

Declaration, allocation and initialisation :

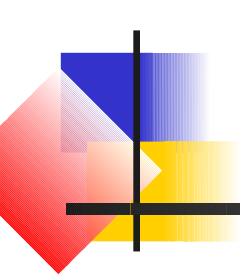
```
int tab[] = {1,9,2,4} ;
```



Arrays: initialisation

Initialisation by default :

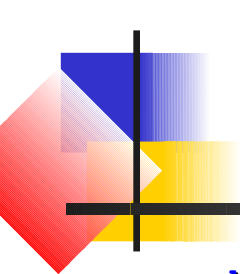
```
int [] tb = new int[3];  
           // 0,0,0  
char [] ch = new char[3];  
           // '\0','\0','\0'  
boolean [] bt = new boolean[3];  
            // false,false,false
```



Class DisplayTab

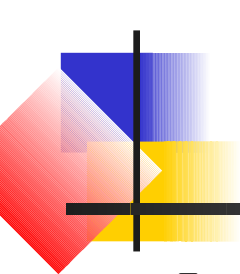
```
public class DisplayTab {  
    public static void main(String args[]){  
        int[] tab={10,20,30,40};  
        for(int i=0;i<=tab.length-1;i++)  
        {  
            System.out.println("tab["+i+"]="+tab[i]);  
        }  
    }  
}
```

```
Java/Essais> java DisplayTab  
tab[0] = 10  
tab[1] = 20  
tab[2] = 30  
tab[3] = 40
```

Class Notes

```
public class Notes {
public static void main(String args[]){
int NotesNb;
System.out.println("Number of notes?");
NotesNb=Keyboard.readInt();
double[] theNotes = new double[NotesNb];
for(int i=0;i<=theNotes.length-1;i++){
    System.out.println("Note nb."+(i+1)+"?");
    theNotes[i]=Keyboard.readDouble();    }
System.out.println();
System.out.println("Notes in the array:");
System.out.println("*****");
for(int i=0;i<=theNotes.length-1;i++){
    System.out.print("Note nb."+(i+1)+"=");
    System.out.println(theNotes[i]);  }
}
}
```



Class Notes

```
Java/Essais> java Notes
```

```
Number of notes? 4
```

```
Note nb. 1? 7.6
```

```
Note nb. 2? 11
```

```
Note nb. 3? 14
```

```
Note nb. 4? 5
```

```
Notes in the array:
```

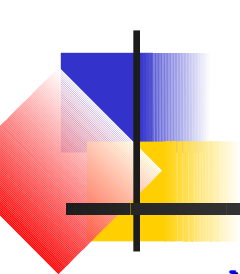
```
*****
```

```
Note nb. 1 = 7.6
```

```
Note nb. 2 = 11.0
```

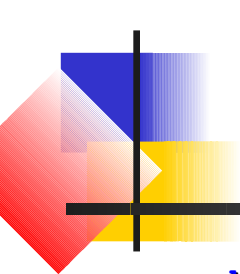
```
Note nb. 3 = 14.0
```

```
Note nb. 4 = 5.0
```



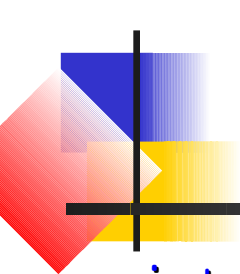
Equality between arrays

```
public class Tab5 {  
    public static void main(String[] argv){  
        int[] tab1,tab2,tab3;  
        tab1= new int[3];  
        tab1[0]=10;tab1[1]=20;tab1[2]=30;  
        tab2 = new int[3];  
        tab2[0]=10;tab2[1]=20;tab2[2]=30;  
        tab3 = tab2 ;  
        if(tab1==tab2){  
            System.out.println("tab1 and tab2 are equal");  
        } else {  
            System.out.println("tab1 and tab2 are different");  
        }  
        if(tab2==tab3){  
            System.out.println("tab2 and tab3 are equal");  
        } else {  
            System.out.println("tab2 and tab3 are different"); }  
    }  
}
```



Sorting the array by selection

```
public class SortSelection{
public static void main(String args[]){
int n; int [] tab;
System.out.println("Number of integers to sort?");
n = Keyboard.readInt(); tab = new int[n];
for(int i=0;i<=tab.length-1;i++){
System.out.println("An integer?");
tab[i] = Keyboard.readInt();}
for(int i=0;i<=tab.length-1;i++){ int Im=i;
for(int j=i+1;j<=tab.length-1;j++) {
if(tab[j]< tab[Im]){ Im=j ; }
}
int buffer=tab[i]; tab[i]=tab[Im]; tab[Im]=buffer;
}
System.out.println("Sorted array:");
for(int i=0;i<=tab.length-1;i++){
System.out.println(tab[i]);}
System.out.println();}
}
```



Multidimensional array

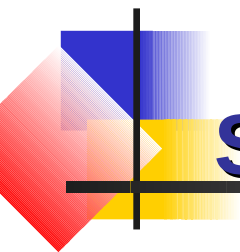
```
int [] t;           // one dimension
int [][] m;         // two dimensions
char [][][] p;      // three dimensions

int [][] t = new int [3][5]; //3 lines, 5 columns
T[1][2] = 7;

int [][] t = new int [3][5];
System.out.println(t.length);    // displays 3
System.out.println(t[1].length); // displays 5

int [][] tab = { {1,2,3,4}, {5,6,7,8}, {9,10,11,12}};
int [] t = tab[1];
for (int j = 0; j<= t.length-1; j++) {
System.out.println(t[j]);
}

// what are the displayed values ?
```



Summary

Conditions : `if .. else`

Loops: `for, while, do-while`

Arrays:

- simple
- multi-dimensional