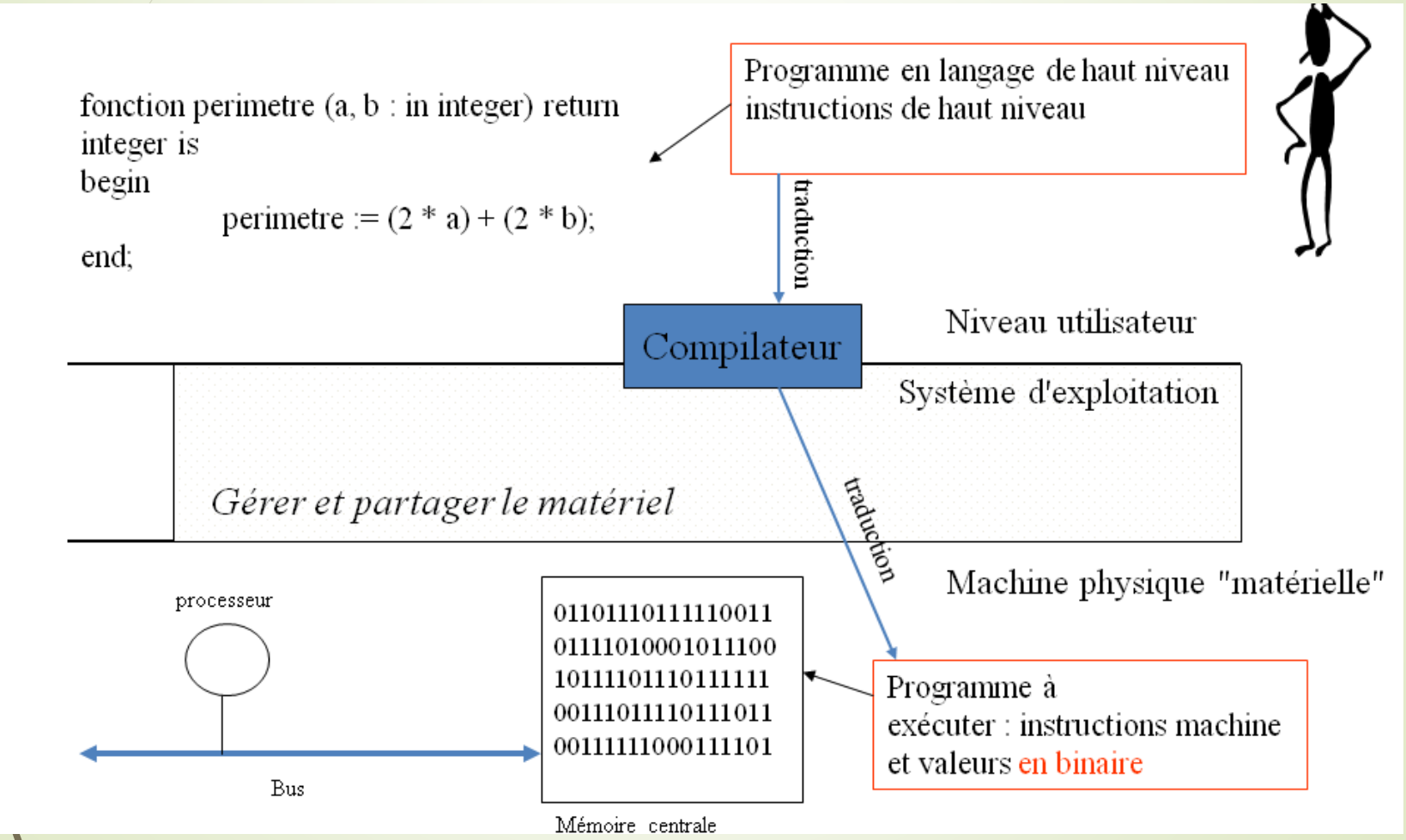
Introduction à la programmation JAVA

Définitions : programme

- Un programme est constitué d'instructions (actions) qui agissent sur des données (constantes, variables)
  - Les constantes et variables peuvent avoir un type (caractères, entier, flottant)
  - Les constantes et variables doivent être créées au début du programme
  - Il existe différents types d'instructions
    - Entrées – sorties lires les données depuis le clavier; écrire sur l'écran
    - Calcul (+ - / *); affectation
    - test / logique < > <= >=
    - Les instructions de contrôle : itération et conditionnelle

JAVA

- Java est un langage de haut niveau



Exemple d'un programme Java simple

Le programme `Bonjour`

Ce texte correspond à un programme Java syntaxiquement et sémantiquement correct.

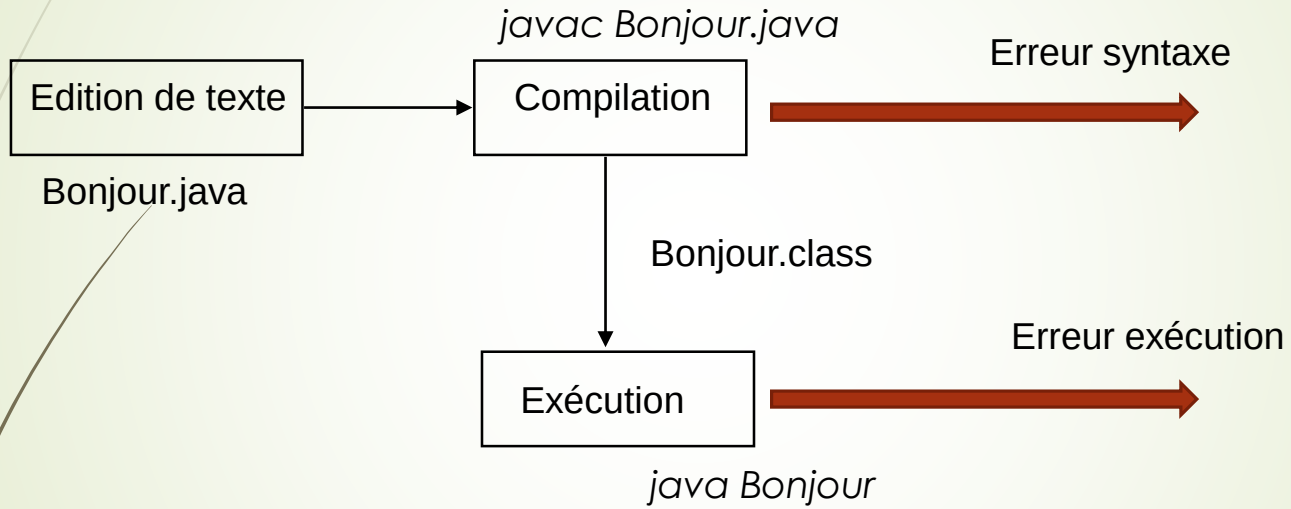
```
/* Premier programme
   -a enregistrer dans le fichier Bonjour.java
   -a compilar par javac Bonjour.java
   -a executer par java Bonjour
*/

public class Bonjour {
   public static void main (String[] args) {
      Terminal.ecrireStringln("Bonjour tout le monde !");
   }
}
```

- Le texte entre `/*` et `*/` est un commentaire.
- Le nom après `public class` est le nom du programme. Dans cet exemple, c'est `Bonjour`.
- Ce programme ne manipule qu'une donnée : le message

  `Bonjour tout le monde !`

Exemple d'un programme Java simple



Format d'un programme java

```
public class  Nom_Programme{
public static void main (String[] args) {
....
}
}
```

```
public class essai1 {
public static void main (String[] args) {
int x;
int y;
y=2;
x=y+5;
x=x*2;
}
}
```

Déclarations variables x et y avec un type

Instructions simples
Affectation =
Calcul : + - * /

Instructions entrées-sorties

- Ces instructions sont définies dans le fichier Terminal.java qui est placé dans le même répertoire que les programmes que nous construisons.

- lireInt() // Lire un entier

```
x = Terminal.lireInt();
```

- ecrireString(String s) // écrire une chaine de caractères

```
Terminal.ecrireString( "bonjour ") ;
```

- ecrireInt(int i) // Afficher un entier

```
Terminal.ecrireInt( x) ;
```

Exercice 1 : programme Java équivalent ?

```
FONCTIONS_UTILISEES
VARIABLES
  x EST_DU_TYPE NOMBRE
  y EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
  AFFICHER "Quelle est la valeur de y ?"
  LIRE y
  x PREND_LA_VALEUR y + 3
  AFFICHER "Voici la valeur de x"
  AFFICHER x
FIN_ALGORITHME
```

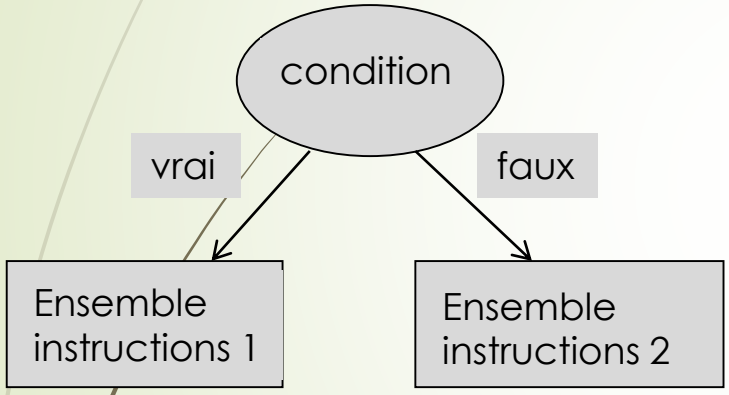
```
public class  Exo1{
public static void main (String[] args) {
int x;
int y;
Terminal.ecrireString("Quelle est la valeur de y ?");
y = Terminal.lireInt();
x = y + 3;
Terminal.ecrireString("Voici la valeur de x : ");
Terminal.ecrireIntln(x);
}
}
```

Instructions de controle

- Les structures de contrôles sont les éléments du langage qui déterminent l'ordre dans lequel les instructions sont exécutées.
  - La séquence : exécution en séquence des instructions
  - La conditionnelle : introduit un aiguillage, un choix entre des ensembles de instructions
  - La boucle : introduit une itération, un ensemble de instructions est exécuté plusieurs fois.

Instructions conditionnelle

- La conditionnelle est une structure de contrôle qui permet de lier l'exécution d'une séquence d'instructions au résultat d'un test. Le test est appelé condition ou prédicat.



Si (prixfinal < 200)

Alors

prix = prixfinal + 20

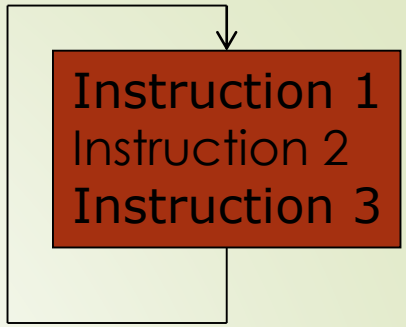
Sinon

prix = prixfinal + 5

Fsi

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

Instructions itération



- Une **boucle** est une structure de contrôle itérative qui permet de répéter un traitement un certain nombre de fois.

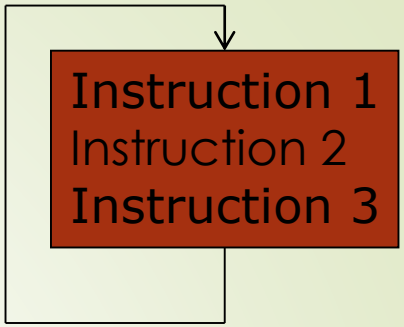
```
FONCTIONS_UTILISEES
VARIABLES
  x EST_DU_TYPE NOMBRE
  y EST_DU_TYPE NOMBRE
  I EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
POUR I ALLANT_DE 1 A 3
  DEBUT_POUR
  AFFICHER "Quelle est la valeur de y ?"
  LIRE y
  x PREND_LA_VALEUR y + 3
  AFFICHER "Voici la valeur de x"
  AFFICHER x
  FIN_POUR
FIN_ALGORITHME
```

```
for (i=1; i <= N; i = i + 1)
{
}
```

```
public class  Exo1{
public static void main (String[] args) {
int x;
int y;
int i;
for (i=1; i <= 3; i = i + 1) {
Terminal.ecrireString("Quelle est la valeur de y ?");
y = Terminal.lireInt();
x = y + 3;
Terminal.ecrireString("Voici la valeur de x : ");
Terminal.ecrireIntln(x);
}
}
}
```

Instructions itération

- Une **boucle** est une structure de contrôle itérative qui permet de répéter un traitement un certain nombre de fois.




```
FONCTIONS_UTILISEES
VARIABLES
  x EST_DU_TYPE NOMBRE
  y EST_DU_TYPE NOMBRE
  I EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
  I PREND_LA_VALEUR 1
  TANT_QUE (I <= 3) FAIRE
    DEBUT_TANT_QUE
    AFFICHER "Quelle est la valeur de y ?"
    LIRE y
    x PREND_LA_VALEUR y + 3
    AFFICHER "Voici la valeur de x"
    AFFICHER* x
    I PREND_LA_VALEUR I+1
    FIN_TANT_QUE
FIN_ALGORITHME
```

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

```
public class  Exo2{
public static void main (String[] args) {
int x;
int y;
int i;
i = 1;
while( i <= 3) {
Terminal.ecrireString("Quelle est la valeur de y ?");
y = Terminal.lireInt();
x = y + 3;
Terminal.ecrireString("Voici la valeur de x : ");
Terminal.ecrireIntln(x);
i = i + 1;
}
}
}
```

Exercice 2 : programme Java équivalent

```
x = 10;
y = 0;
Tant que (x > 0)
Faire
      y = y + x;
      x = x – 1;
Fait
```

```
public class  Exo2{
public static void main (String[] args) {
int x;
int y;
x = 10;
y = 0;
while (x > 0)
{
  y = y + x;
  x = x - 1;
}
Terminal.ecrireString("Voici la valeur de y : ");
Terminal.ecrireIntln(y);
}
}
```

Exerçons nous….

1) Un utilisateur donne un prix en franc. Le programme affiche sa conversion en euros.

- Programme ?

Exerçons nous…

2. Un utilisateur donne deux nombres x et y. Puis il choisit de faire soit l'opération x + y ou l'opération x * y. Le programme exécute l'opération demandée et affiche le résultat. Le programme recommence ces opérations tant que l'utilisateur le demande.

- Algorithme ?

- Programme ?

Exerçons nous….

3. Ecrivez un programme pour calculer x puissance y où x et y sont des valeurs saisies par l'utilisateur

- Programme ?

Devoir à rendre

Ce sont les soldes. Le programme calcule la remise et le prix final d'un article et affichent ces deux valeurs.

Si le prix de l'article soldé est supérieur à 100 euros, la réduction est de 40 %.

Si le prix de l'article soldé est inférieur à 50 euros, la réduction est de 10 %.

Si le prix de l'article est compris entre 50 et 100 euros, la réduction est de 20 %.

Le programme demande le prix d'un article et affiche en final le montant de la réduction et le prix remisé.

Question 1 : donnez le programme algobox qui permet de coder ce problème pour un article.

Question 2 : donnez le programme Java qui permet de coder ce problème pour n articles, n est donné par l'utilisateur.

Question 3 : BONUS : en utilisant l'exemple qui suit, essayer d'utiliser un tableau pour stocker les données des n articles.

Devoir à rendre

Jeu d'essais

	Article 1	Article 2	Article 3	Total
Prix non soldé	120	52	12	184
réduction	48	10,4	1,20	59,6
Montant soldé	72	41,6	10,8	124,4

Notion de tableau

4. Un tableau est une structure qui contient n éléments de même type.

int [] tabNum; // déclaration du tableau

tabNum = new int[taille]; // création, taille est le nombre d'éléments

```
public class Hellonombretab {
public static void main (String[] args) {
int [] tabentier;
int i, x;
i = 0;
tabentier = new int[3];
while (i <3) {
  Terminal.ecrireStringln("donnez un chiffre  ");
  x=Terminal.lireInt();
  tabentier[i] = x;
  i = i + 1;
}
for (i=0; i < tabentier.length; i = i + 1) {
 Terminal.ecrireString("élément du tableau  ");
 Terminal.ecrireInt(i);
 Terminal.sautDeLigne();
 Terminal.ecrireInt(tabentier[i]);
 Terminal.sautDeLigne();}
}
}
```