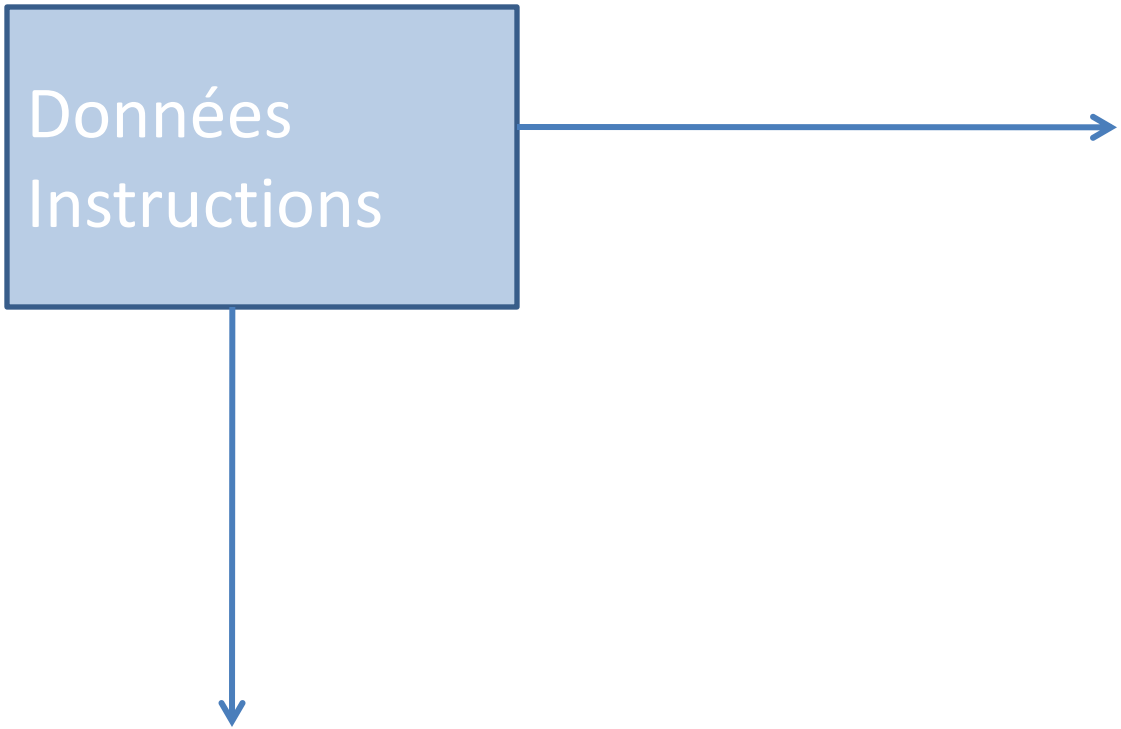
Cours 3 JAVA

Mlle AMEUR .K
E-Mail: ameur.khadidja@univ-ouargla.dz

Programme

- Partie 1 : Les bases du langage Java
  - Introduction
  - Généralités
  - Types Primitifs
  - Operateurs Et Expressions
  - Instructions De Contrôle
  - Tableaux
  - Chaines de caractères

Introduction



Les données de Java

Java utilise les types de données suivants:

- les nombres entiers
- les nombres réels
- les caractères et chaînes de caractères
- les booléens
- les objets

Les types de données prédéfinis

Type Codage Domaine
char 2 octets caractère
int 4 octets
long 8 octets
byte 1 octet
short 2 octets
float 4 octets
double 8 octets [1.7 10-308 , 1.7 10+308] en valeur absolue
boolean 1 bit true, false
String référence d'objet chaîne de caractères
Date référence d'objet date
Character référence d'objet char
Integer référence d'objet int
Long référence d'objet long
Byte référence d'objet byte
Float référence d'objet float
Double référence d'objet double
Boolean référence d'objet boolean

Déclaration des données

Rôle des déclarations?

->

Déclaration des constantes

- Pourquoi déclarer des constantes ?

  La lecture et la modification du programme sera plus facile ;

- Comment :

  final type nom=valeur;

- Exemple:

  final float PI=3.141592F

Déclaration des données

Les conversions entre nombres et chaînes de caractères

nombre -> chaîne *"" + nombre*

chaine -> int *Integer.parseInt(chaine)*

chaîne -> long *Long.parseLong(chaine)*

chaîne -> double *Double.valueOf(chaine).doubleValue*

chaîne -> float *Float.valueOf(chaine).floatValue()*

Les changements de type

- Il est possible, dans une expression, de changer momentanément le codage d'une valeur.
- type casting.
- La syntaxe du changement du type d'une valeur dans une expression est la suivante ***(type)*** **valeur** prend alors le type indiqué. Cela entraîne un changement de codage de la valeur.
- exemple :
  - int i, j;
  - float isurj;
  - isurj= (float)i/j; // priorité de () sur /

Expressions Arithmétiques

- Opérateurs
- + addition
- - soustraction
- * multiplication
- / division

% **modulo:** le résultat est le reste quelque soit la nature des opérandes, le quotient étant lui entier.

- Expressions -> règles de préséance des opérateurs

Fonctions mathématiques :

- *double sqrt(double x) racine carrée*
- *double cos(double x) Cosinus*
- *double sin(double x) Sinus*
- *double tan(double x) Tangente*
- *double pow(double x,double y) x à la puissance y (x>0)*
- *double exp(double x) Exponentielle*
- *double log(double x) Logarithme népérien*
- *double abs(double x) valeur absolue*

Expressions relationnelles

Opérateurs

- Les opérateurs sont les suivants : <, <=, ==, !=, >, >=

Oordre de priorité

- >, >=, <, <=
- ==, !=

Expressions booléennes

Opérateurs

Négation ! ET && OU ||

Ordre de priorité

- !
- &&
- ||

Priorité générale des opérateurs

- () [] fonction gd
- ! ~ ++ -- dg
- new (type) opérateurs cast dg
- * / % gd
- + - gd
- << >> gd
- < <= > >= instanceof gd
- == != gd
- & gd
- ^ gd
- | gd
- && gd
- || gd
- ? : dg
- = += -= etc. . dg

Structure de choix simple

la condition est entourée de parenthèses.

- chaque action est terminée par point-virgule.
- les accolades ne sont pas terminées par point-virgule.
- les accolades ne sont nécessaires que s'il y a plus d'une action.
- la clause else peut être absente.
- Il n'y a pas de then.

Structure de cas

S

La valeur de l'expression de contrôle, ne peut être qu'un entier ou un caractère.

- l'expression de contrôle est entourée de parenthèses.
- la clause *default peut être absente.*
- les valeurs *vi sont des valeurs possibles de l'expression. Si l'expression a pour valeur vi , les actions derrière la clause* ***case vi sont*** exécutées.
- l'instruction *break fait sortir de la structure de cas. Si elle est absente à la fin du bloc d'instructions de la valeur vi, l'exécution se* poursuit alors avec les instructions de la valeur *vi+1.*

}

En Java


```
int choix, erreur;
switch(choix)
{
case 0: System.exit(0);
case 1: M1();break;
case 2: M2();break;
default: erreur=1;
}
```

Structure de répétition
Nombre de répétitions connu

Syntaxe

Notes

- les 3 arguments du *for sont à l'intérieur d'une parenthèse.*
- les 3 arguments du *for sont séparés par des points-virgules.*
- chaque action du *for est terminée par un point-virgule.*
- l'accolade n'est nécessaire que s'il y a plus d'une action.
- l'accolade n'est pas suivie de point-virgule.

Structure de répétition
Nombre de répétitions inconnu

Il existe de nombreuses structures en Java pour ce cas.

- **Structure tantque (while)**

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

On boucle tant que la condition est vérifiée. La boucle peut ne jamais être exécutée.

Structure de répétition
Nombre de répétitions inconnu

Structure répéter jusqu'à (do while)

La syntaxe est la suivante :

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

- On boucle jusqu'à ce que la condition devienne fausse ou tant que la condition est vraie. Ici la boucle est faite au moins une fois.

Structure de répétition
Nombre de répétitions inconnu

- **Structure pour générale (for)**

La syntaxe est la suivante :

```
for(instructions_départ;condition;instructions_fin_boucle)
{
instructions;
}
```

Exemples

Les programmes suivants calculent tous la somme des n premiers nombres entiers.

```
for(i=1, somme=0;i<=n;i=i+1)
            somme=somme+a[i];

for (i=1, somme=0;i<=n;somme=somme+a[i], i=i+1);

i=1;somme=0;
while(i<=n)
          { somme+=i; i++; }

i=1; somme=0;
do somme+=i++;
     while (i<=n);
```

Exercice 1

Développez une application qui demande à l'utilisateur d'enter deux nombres, reçoit ces deux nombres et afficher la somme, le produit, la différence selon le choix d'utilisateur.

(Boite de dialogue)

Références

- H.M.Deitel et P.J.Deitel, Comment Programmer en JAVA 4ème édition.
- X. BLANC & J. DANIEL , Le langage Java.