

Lire les entrées

Pour saisir des informations et les stocker dans des variables nous allons utiliser la méthode suivante :

Pour lire ce que vous tapez au clavier, vous allez devoir utiliser un objet de type **Scanner**.

Or, la classe **Scanner** se trouve dans le package **java.util**.

Pour utiliser cette fonction, il faut, au préalable, importer dans le programme ce package de la manière suivante :

```
import java.util.Scanner;
```

Si , vous souhaitez charger l'ensemble des classes du package **java.util**, écrire

```
import java.util.Scanner;
```

Lire les entrées

Donc lorsque vous créez une classe java qui permette de saisir des données provenant d'un utilisateur, on va donc utiliser

- pour faire **afficher (a)** , la fonction **System.out.println("a")**
- Pour faire **saisir (a)**, on doit :
 - Instancier l'objet scanner :
Scanner sc = new Scanner(System.in);
 - Puis affecter la saisie à une variable :
Type variable = sc.nextLine();

Lire les entrées

Pour illustrer, voici l'exemple suivant :

```
//Appel Package java.util
import java.util.*;
public class Exemple1 {
    public static void main(String[] args){
        //Un nouvel objet scanner est instancier
        Scanner sc = new Scanner(System.in);
        // Affichage du premier message
        System.out.println("Saisissez un entier : ");
        // Affectation de l'entrée à la variable entier i
        int i = sc.nextInt();
        // Affichage du deuxième message
        System.out.println("Saisissez une chaîne : ");
        // Affectation de l'entrée à la variable chaîne str
        String str = sc.nextLine();
        System.out.println("FIN ! ");
    }
}
```

Le Cnam Rhône-Alpes - 4 rue Bayard - 69007 Lyon - 04 78 58 30 99

Exercices

1 : Ecrire l'exemple précédent sous eclipse et puis compiler pour générer le fichier .class.

2 : A partir du 1^{er} algorithme créé, veuillez écrire le programme qui permet de saisir le prix HT d'un article et de calculer son prix total TT, TVA = 20% . Pour cela on doit utiliser l'instruction nextfloat()

Rappel

Pour Exercice 2 :

```
Algorithme    Calcul_PTTC
Variables      PHT, PTTC : Réel
Constante     TVA = 0.2
Début
    Afficher("Entrez le prix hors taxe : ")
    Saisir(PHT)
    PTTC ← PHT + ( PHT * TVA)
    Afficher("Le prix TTC est ", PTTC)
Fin
```

L'instruction conditionnel

Rappel : Pour l'utilisation des instructions conditionnelles on doit connaître les opérateurs de comparaison suivants:

- « == » : permet de tester l'égalité.
- « != » : permet de tester l'inégalité.
- « < » : strictement inférieur.
- « <= » : inférieur ou égal.
- « > » : strictement supérieur.
- « >= » : supérieur ou égal.
- « && » : l'opérateur ET.
- « || » : le OU.

L'instruction conditionnel

La structure Si ... Alors ... Sinon ... FinSi

En JAVA, la syntaxe est la suivante :

```
if(condition)
{
    Instruction(s)1 //Exécution si la condition est remplie
}
else
{
    Instruction(s)2 //Exécution si la condition n'est pas remplie
}
```

Exercice 1 : Écrire le programme JAVA qui affiche si un nombre entier saisi au clavier est pair ou impair.

Rappel Exercice

On dit qu'un nombre entier n est pair si le reste r de la division entière de n par 2 est égale à 0. Sinon il est impair.

```
Algorithme Parité
//déclaration des variables
Variables n, r : entiers
Début
    Afficher ("Entrez la valeur de n : ")
    Saisir (n)
    //r est le reste de la division entière de n par 2
    r ← n Mod 2
    Si r = 1 Alors
        Afficher (n, " est impair")
    Sinon
        Afficher(n, " est pair")
    FinSi
Fin
```

L'instruction conditionnel

La structure Si ... Alors ... SinonSi ... Sinon FinSi

En JAVA, la syntaxe est la suivante :

```
if(condition1)
{
    Instruction(s)1 //Exécution si la condition1 est remplie
}
elseif(condition2)
{
    Instruction(s)2 //Exécution si la condition2 est remplie
}
else
{
    Instruction(s)3 //Exécution si la condition n'est pas remplie
}
```

Exercice : Écrire le programme qui demande deux nombres m et n à l'utilisateur et l'informe ensuite si le produit est négatif ou positif.

Rappel Exercice

```
Algorithme Signe_Produit
//déclaration des variables
Variables m, n : entiers
Début
    Afficher("Entrez deux nombres m et n :")
    Saisir (m,n)
    //Verifie que m ou n =0
    Si m = 0 Ou n = 0 Alors
        Afficher("Le produit est nul")
    //Verifie que m et n sont positifs ou négatifs
    SinonSi (m < 0 Et n < 0) Ou (m > 0 Et n >0) Alors
        Afficher("Le produit est positif")
    Sinon
        Afficher("Le produit est négatif")
    FinSi
Fin
```

L'instruction conditionnel

Structure de choix multiples

En JAVA, la syntaxe est :

```
switch (selecteur)
{
    case Valeurs1:
        Instruction(s)1;
        break;
    case Valeurs2:
        Instruction(s)2;
        break;
    .....
    default:
        Instruction(s)3;
}
```

Exercice : Écrire algorithme qui affiche le mois en toute lettres selon son numéro saisi.

Rappel Exercice

Algorithme Mois
Variables N : Entier
Début
 Afficher("Donner le numéro du mois :")
 Saisir(N)
 Suivant N faire
 1 : Afficher("Janvier")
 2 : Afficher("Février")
 3 : Afficher("Mars")
 4 : Afficher("Avril")
 5 : Afficher("Mai")
 6 : Afficher("Juin")
 7 : Afficher("Juillet")
 8 : Afficher("Août")
 9 : Afficher("Septembre")
 10 : Afficher("Octobre")
 11 : Afficher("Novembre")
 12 : Afficher("Décembre")
 Sinon
 Afficher("Le numéro saisi est incorrecte :")
 FinSuivant
Fin

Les boucles itératives

La boucle TantQue ... Faire

En JAVA, la syntaxe est :

```
while (Condition)
{
    Instruction(s) //Instructions à répéter
}
```

Exercice : Ecrire un algorithme qui calcule $S = 1 + 2 + 3 + 4 + 5 + \dots + N$.

Rappel Exercice

```
Algorithme Somme
//déclaration des variables
Variables S, I, N : Entier
Début
    Afficher("Entrer la valeur de N :")
    Saisir(N)
    //Affectation valeur aux variables
    S ← 0
    I ← 1
    //Lancement de la boucle itérative
    TantQue I <= N Faire
        S ← S + I
        I ← I + 1
    FinTantQue
    //Affichage du résultat
    Afficher("La somme ", S)
Fin
```

Les boucles itératives

La boucle Pour ... Faire

En JAVA, la syntaxe est

```
for(int i = ValeurInit ; i <= ValeurFin; Itération) // Itération i++ => i+1
{
    Instruction(s);
}
```

Exercice : Écrire un algorithme qui saisi un nombre entier et qui calcule la somme des entiers pairs jusqu'à ce nombre.

Rappel Exercice

Algorithme Somme

//déclaration des variables

Variables S, I, N :Entier

Début

Afficher("Entrer la valeur de N :")

Saisir(N)

//Initialisation de la variable résultat

S ← 0

//Lancement de la boucle itérative avec un pas de 2

Pour I allant de 0 à N Pas 2 Faire

S ← S + I

FinPour

//Affichage du résultat

Afficher("La somme des nombres pairs est :", S)

Fin

Les boucles itératives

La boucle Répétez ... Jusqu'à

En JAVA, la syntaxe est

```
do{  
    Instruction(s)  
}while(condition);
```

Exercice : Ecrire un algorithme qui calcule la somme des N premiers nombres entiers. On suppose que N est strictement positif.

Rappel Exercice

Algorithme Somme

//déclaration des variables

Variables S, I, N : Entier

Début

Afficher("Entrer une valeur strictement positif :")

Saisir(N)

//Initialisation des variables

S ← 0

I ← 1

//Lancement de la boucle itérative

Répéter

S ← S + i

I ← I + 1

Jusqu'à I >= N

//Affichage du résultat

Afficher("La somme des", N, " premiers entiers est :", S)

Fin

Exercices

Exercice 1 : Simulation d'une calculatrice

Écrire un algorithme qui permet de saisir deux variables réelles a et b et un opérateur simple : +, -, *, / et afficher le résultat.

Exercice 2 : Mention

Écrire un algorithme qui lit la moyenne générale (MG) d'un étudiant et affiche la mention.

Exercice 3 : Maximum de dix nombres

Écrire un algorithme qui permet d'afficher le maximum parmi dix nombres saisis au clavier.

Rappel Exercice 1

Algorithme Calculatrice
Variables A, B, R : Réel
OP : caractère

Début

//Saisir des données
Afficher("Entrez la valeur de A et B :")
Saisir(A, B)
//Saisie de l'opérateur
Afficher("Entrez l'opérateur de votre choix :")
Saisir(OP)

Suivant OP Faire
'+' : R ← A+B
'-' : R ← A-B
'*' : R ← A * B
'/' : Si B = 0 alors
 Afficher("Division par zéro")
 //Sortir du programme
 Quitter
 Sinon
 R ← A / B
 FinSi
 Sinon
 Afficher("Erreur de saisie")
 //Quitter la procédure
 Quitter
 FinSuivant
 Afficher(A, OP, B, " = ", R)
Fin

Rappel Exercice 2

Algorithme Mention
//MG : Moyenne Générale
Variable MG : réel
Mention : chaîne
Début
 Afficher("Donner la moyenne générale de l'étudiant : ")
 Saisie(MG)
 Si MG >= 16 Alors
 Mention ← "Très bien"
 Sinon Si MG >= 14 alors
 Mention ← "Bien"
 Sinon Si MG >= 12 alors
 Mention ← "Assez bien"
 Sinon Si MG >= 10 alors
 Mention ← "Passage"
 Sinon
 Mention ← "Mauvais résultat"
FinSi
 Afficher("La mention de l'étudiant est : ",Mention)
Fin

Rappel Exercice 3

Algorithme Max
Variables N, I, Max : Entier
Début
 Max ← 0
 Pour I Allant de 1 à 10 Faire
 Afficher ("Entrer un nombre :")
 Saisir(N)
 Si I = 1 ou N > Max Alors
 Max ← N
 FinSi
 FinPour
 Afficher("Le nombre le plus grand est : ",Max)
Fin