

DUT SRC – IUT de Marne-la-Vallée  
02/12/2011  
INF120 - Algorithmique

# ***Cours 5***

## ***Entrées-sorties et fonctions***

# Sources

---

- *Le livre de Java premier langage*, d'A. Tasso
- Cours INF120 de J.-G. Luque
- Cours de J. Henriet : <http://julienhenriet.olymp-network.com/Algo.html>

# Plan du cours 5 – Entrées-sorties et fonctions

---

- Résumé de l'épisode précédent
- Les entrées-sorties
- Les fonctions

# Plan du cours 5 – Entrées-sorties et fonctions

---

- Résumé de l'épisode précédent
- Les entrées-sorties
- Les fonctions

# Résumé de l'épisode précédent

---

## Tableaux :

- pour stocker un ensemble de valeurs **de même type**
- **une** valeur par case
- **nombre de cases fixé** à l'initialisation du tableau
- boucle pour **parcourir le tableau**

## Boucles :

- boucle **Tant que** et boucle **Pour tout**
- attention à l'**initialisation** et la **condition d'arrêt** (premier et dernier passage dans la boucle)

# Plan du cours 5 – Entrées-sorties et fonctions

---

- Résumé de l'épisode précédent
- Les entrées-sorties
- Les fonctions

# Les entrées-sorties

## **Entrées-sorties dans la communication ordinateur – utilisateur :**

- périphériques de saisie d'entrées : clavier, souris, joystick, webcam, Wii remote, Kinect...
- périphérique d'affichage des sorties : écran, vidéo-projecteur, imprimante...

Différent des **variables d'entrée / variable de sortie** dans la **“communication entre algorithmes”**.

# La souris

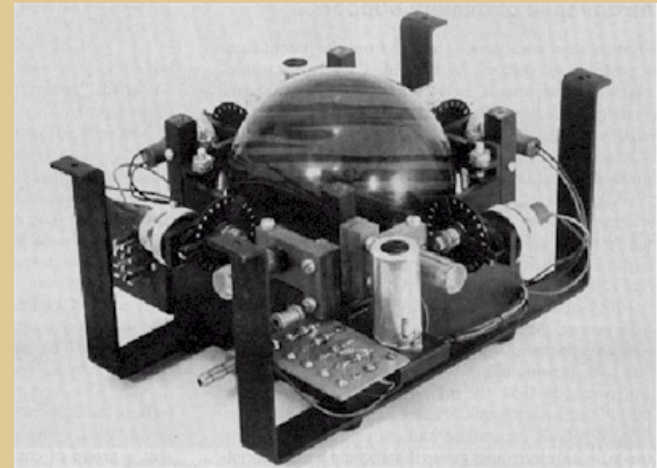
*La “minute culturelle”*

L'invention de la souris

1952 Trackball (boule de commande)  
Tom Cranston et Fred Longstaff  
(Marine Royale Canadienne)

1963 Souris mécanique  
Douglas Engelbart et Bill English  
(Stanford Research Institute)

1977 Souris optique  
Jean-Daniel Nicoud et André Guignard  
(Ecole polytechnique fédérale de Lausanne)





# La souris

*La “minute culturelle”*

L'invention de la souris

1952 Trackball (boule de commande)  
Tom Cranston et Fred Longstaff  
(Marine Royale Canadienne)

1963 Souris mécanique  
Douglas Engelbart et Bill English  
(Stanford Research Institute)

1977 Souris optique  
Jean-Daniel Nicoud et André Guignard  
(Ecole polytechnique fédérale de Lausanne)



# La souris

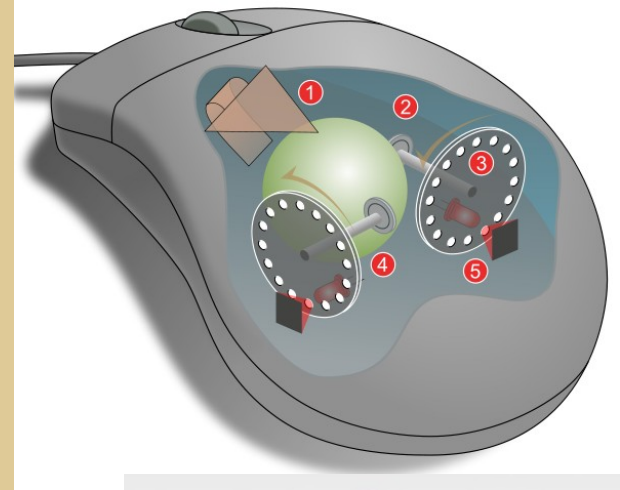
## *La "minute culturelle"*

### L'invention de la souris

1952 Trackball (boule de commande)  
Tom Cranston et Fred Longstaff  
(Marine Royale Canadienne)

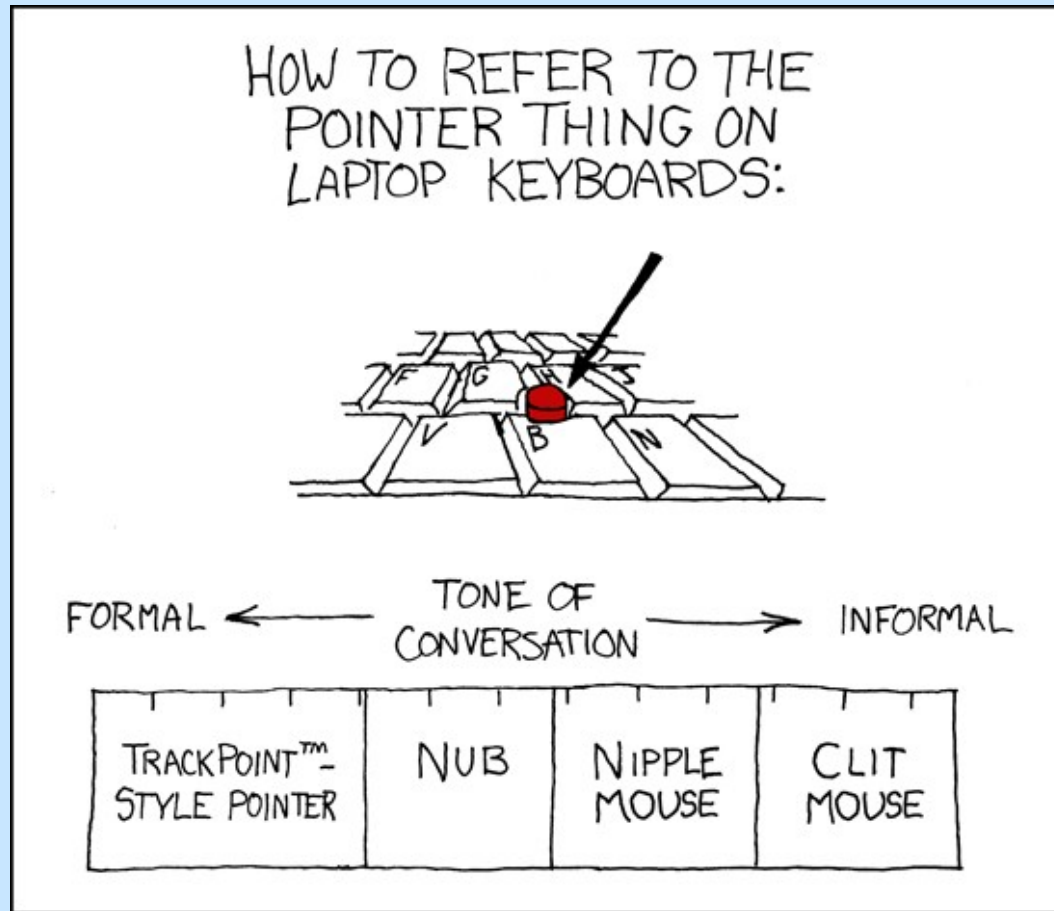
1963 Souris mécanique  
Douglas Engelbart et Bill English  
(Stanford Research Institute)

1977 Souris optique  
Jean-Daniel Nicoud et André Guignard  
(Ecole polytechnique fédérale de Lausanne)



# Autres périphériques d'entrée

La "minute xkcd"



<http://xkcd.com/243>

<http://xkcd.free.fr?id=243>

# Autres périphériques d'entrée

*La "minute culturelle"*

## Tracking fingers with the Wii Remote

jcl5m

21 vidéos

S'abonner



J'aime



+ Ajouter à

Partager



2 893 269



Ajoutée par jcl5m le 8 nov. 2007

Données  
transmises par  
la WiiRemote :


image de  
**profondeurs**

# Autres périphériques d'entrée

*La "minute culturelle"*

**Control your 3D application with Kinect**

SimplySim3D 12 vidéos  S'abonner



J'aime  + Ajouter à  Partager

8 546

Ajoutée par SimplySim3D le 10 janv. 2011

Données  
transmises par  
la Kinect :

image de  
**couleurs +  
profondeurs**

# Les entrées-sorties

## Entrées-sorties dans la communication ordinateur – utilisateur

Quel **type de données** utiliser en **algorithmique** pour coder les entrées-sorties ?

Périphérique	Type de données transmises
Clavier	
Souris à 1 bouton	
Webcam	
Kinect	
Ecran	

# Les entrées-sorties

## Entrées-sorties dans la communication ordinateur – utilisateur

Quel **type de données** utiliser en **algorithmique** pour coder les entrées-sorties ?

Périphérique	Type de données transmises
Clavier	chaîne de caractères
Souris à 1 bouton	
Webcam	
Kinect	
Ecran	

# Les entrées-sorties

## Entrées-sorties dans la communication ordinateur – utilisateur

Quel **type de données** utiliser en **algorithmique** pour coder les entrées-sorties ?

Périphérique	Type de données transmises
Clavier	chaîne de caractères
Souris à 1 bouton	deux entiers (abscisse et ordonnée) + un booléen (clic ou pas)
Webcam	
Kinect	
Ecran	



# Les entrées-sorties

## Entrées-sorties dans la **communication ordinateur – utilisateur**

Quel **type de données** utiliser en **algorithmique** pour coder les entrées-sorties ?

Périphérique	Type de données transmises
Clavier	chaîne de caractères
Souris à 1 bouton	deux entiers (abscisse et ordonnée) + un booléen (clic ou pas)
Webcam	image, donc tableau de tableaux de couleurs RGB
Kinect	
Ecran	

# Les entrées-sorties

## Entrées-sorties dans la communication ordinateur – utilisateur

Quel **type de données** utiliser en **algorithmique** pour coder les entrées-sorties ?

Périphérique	Type de données transmises
Clavier	chaîne de caractères
Souris à 1 bouton	deux entiers (abscisse et ordonnée) + un booléen (clic ou pas)
Webcam	image, donc tableau de tableaux de couleurs RGB
Kinect	image + tableau de tableaux d'entiers (profondeur)
Ecran	Si ligne de commande :

# Les entrées-sorties

## Entrées-sorties dans la communication ordinateur – utilisateur

Quel **type de données** utiliser en **algorithmique** pour coder les entrées-sorties ?

Périphérique	Type de données transmises
Clavier	chaîne de caractères
Souris à 1 bouton	deux entiers (abscisse et ordonnée) + un booléen (clic ou pas)
Webcam	image, donc tableau de tableaux de couleurs RGB
Kinect	image + tableau de tableaux d'entiers (profondeur)
Ecran	Si ligne de commande : chaîne de caractères  Si interface graphique :

# Les entrées-sorties

## Entrées-sorties dans la communication ordinateur – utilisateur

Quel **type de données** utiliser en **algorithmique** pour coder les entrées-sorties ?

Périphérique	Type de données transmises
Clavier	chaîne de caractères
Souris à 1 bouton	deux entiers (abscisse et ordonnée) + un booléen (clic ou pas)
Webcam	image, donc tableau de tableaux de couleurs RGB
Kinect	image + tableau de tableaux d'entiers (profondeur)
Ecran	Si ligne de commande : chaîne de caractères  Si interface graphique : image, donc tableau de tableaux de couleurs RGB

# Les entrées-sorties

Entrées-sorties dans la communication ordinateur – utilisateur

en pseudo-code  
(ou en Java dans  
DM2.java)

Quel **type de données** utiliser en **algorithmique** pour coder les entrées-sorties ?

Périphérique	Type de données transmises
Clavier	chaîne de caractères  chaîne de caractères  <b>reponseALaQuestion</b> ( <i>questionAAfficher</i> ) affiche la question <i>questionAAfficher</i> , renvoie une chaîne de caractères.
Ecran	Si ligne de commande : chaîne de caractères

# Les entrées-sorties

## Entrées-sorties dans la communication ordinateur – utilisateur

en pseudo-code  
(ou en Java dans  
DM2.java)

Quel **type de données** utiliser en **algorithmique** pour coder les entrées-sorties ?

Périphérique	Type de données transmises
Clavier	<p>chaîne de caractères</p> <p>chaîne de caractères</p> <p><b>reponseALaQuestion</b>(<i>questionAAfficher</i>)</p> <p>affiche la question <i>questionAAfficher</i> et renvoie une chaîne de caractères.</p> <p>Exemple : <b>reponseALaQuestion</b>("Quel est votre nom") me laisse taper mon nom au clavier et renvoie "Gambette"</p>
Ecran	<p>Si ligne de commande :</p> <p>chaîne de caractères</p> <p>chaîne de caractères</p> <p><b>Affiche</b>(<i>chaineAAfficher</i>)</p> <p>affiche la chaîne de caractères <i>chaineAAfficher</i> et ne renvoie rien.</p>

# Les entrées-sorties

en Java

## Entrées-sorties dans la communication ordinateur – utilisateur

Quel **type de données** utiliser en **algorithmique** pour coder les entrées-sorties ?

Périphérique	Type de données transmises
Clavier	chaîne de caractères  <pre>Scanner lectureClavier = new Scanner(System.in); String stringLu = lectureClavier.next();</pre> LectureClavier.next() renvoie une chaîne de caractères.
Ecran	Si ligne de commande : chaîne de caractères

# Les entrées-sorties

en Java

## Entrées-sorties dans la communication ordinateur – utilisateur

Quel **type de données** utiliser en **algorithmique** pour coder les entrées-sorties ?

Périphérique	Type de données transmises
Clavier	<p>chaîne de caractères</p> <pre>Scanner lectureClavier = new Scanner(System.in); String stringLu = lectureClavier.next();</pre> <p>LectureClavier.next() renvoie une chaîne de caractères.</p> <pre>int intLu = lectureClavier.nextInt();</pre> <p>LectureClavier.nextInt() renvoie un entier.</p>
Ecran	<p>Si ligne de commande : chaîne de caractères</p>



# Les entrées-sorties

## Entrées-sorties dans la communication ordinateur – utilisateur

en pseudo-code  
(ou en Java dans  
DM2.java)

Quel **type de données** utiliser en **algorithmique** pour coder les entrées-sorties ?

Périphérique	Type de données transmises
Clavier	<p>chaîne de caractères</p> <p>chaîne de caractères</p> <p><b>reponseALaQuestion</b>(<i>questionAAfficher</i>) affiche la question <i>questionAAfficher</i>, renvoie une chaîne de caractères.</p>
Ecran	<p>Si ligne de commande : chaîne de caractères</p> <p>chaîne de caractères</p> <p><b>Affiche</b>(<i>chaineAAfficher</i>) affiche la chaîne de caractères <i>questionAAfficher</i>, puis retourne à la ligne, mais ne renvoie rien.</p>

## Entrées-sorties dans la communication ordinateur – utilisateur

Quel **type de données** utiliser en **algorithmique** pour coder les entrées-sorties ?

Périphérique	Type de données transmises
Clavier	<p>chaîne de caractères</p> <pre>Scanner lectureClavier = new Scanner(System.in); String stringLu = lectureClavier.next();</pre> <p>LectureClavier.next() renvoie une chaîne de caractères.</p> <pre>int intLu = lectureClavier.nextInt();</pre> <p>LectureClavier.nextInt() renvoie un entier.</p>
Ecran	<p>Si ligne de commande :</p> <p>chaîne de caractères</p> <pre>String chaineAAfficher="blabla"; System.out.println(chaineAAfficher);</pre> <p>affiche la chaîne de caractères <i>questionAAfficher</i>, puis retourne à la ligne, mais ne renvoie rien.</p>

## Entrées-sorties dans la communication ordinateur – utilisateur

Quel **type de données** utiliser en **algorithmique** pour coder les entrées-sorties ?

Périphérique	Type de données transmises
Clavier	<p>chaîne de caractères</p> <pre>Scanner lectureClavier = new Scanner(System.in); String stringLu = lectureClavier.next();</pre> <p>LectureClavier.next() renvoie une chaîne de caractères.</p> <pre>int intLu = lectureClavier.nextInt();</pre> <p>LectureClavier.nextInt() renvoie un entier.</p>
Ecran	<p>Si ligne de commande :</p> <p>chaîne de caractères</p> <pre>String chaineAAfficher="blabla"; System.out.print(chaineAAfficher);</pre> <p>affiche la chaîne de caractères <i>questionAAfficher</i>, et ne renvoie rien.</p>

# Plan du cours 5 – Entrées-sorties et fonctions

---

- Résumé de l'épisode précédent
- Les entrées-sorties
- Les fonctions

# Les fonctions

## *La "minute mathématique"*

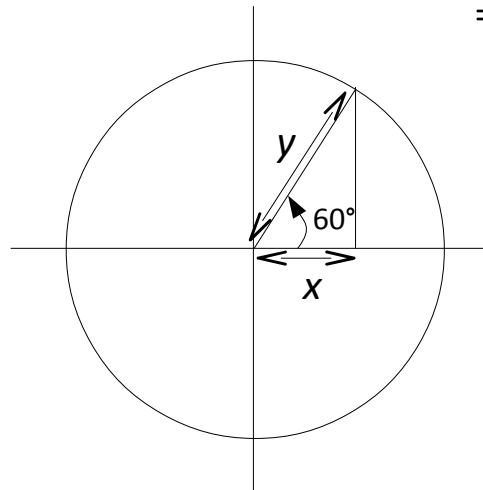
fonction	exemple	entrées possibles	sortie
cosinus	$\text{cosinus}(1.047)=0.5$		
somme			
opposé			
inverse			
différence			
estPositif			
partieEntière			
moyenne			
min			

# Les fonctions

## La "minute mathématique"

fonction	exemple	entrées possibles	sortie
cosinus	$\text{cosinus}(1.047)=0.5$		
somme			
opposé			
inverse			
différence			
estPositif			
partieEntière			
moyenne			
min			

$$\begin{aligned}\text{cosinus}(1.047) &\approx \text{cosinus}(\pi/3) = \text{cosinus}(60^\circ) \\ &= x/y \\ &= 0.5/1\end{aligned}$$



# Les fonctions

## *La "minute mathématique"*

<b>fonction</b>	<b>exemple</b>	<b>entrées possibles</b>	<b>sortie</b>
cosinus	$\text{cosinus}(1.047)=0.5$	flottant	flottant
somme			
opposé			
inverse			
différence			
estPositif			
partieEntière			
moyenne			
min			

# Les fonctions

## *La "minute mathématique"*

fonction	exemple	entrées possibles	sortie
cosinus	$\text{cosinus}(1.047)=0.5$	flottant	flottant
somme	$\text{somme}(2,3)=5$	2 entiers	entier
opposé			
inverse			
différence			
estPositif			
partieEntière			
moyenne			
min			



# Les fonctions

## *La "minute mathématique"*

fonction	exemple	entrées possibles	sortie
cosinus	cosinus(1.047)=0.5	flottant	flottant
somme	somme(2,3)=5	2 entiers	entier
opposé			
inverse			
différence			
estPositif			
partieEntière			
moyenne			
min			

Les "nombres à virgule" se notent avec un point aux Etats-Unis

La virgule sépare les paramètres d'une fonction

# Les fonctions

## *La "minute mathématique"*

fonction	exemple	entrées possibles	sortie
cosinus	$\text{cosinus}(1.047)=0.5$	flottant	flottant
somme	$\text{somme}(2,3)=5$	2 entiers 2 flottants	entier flottant
opposé			
inverse			
différence			
estPositif			
partieEntière			
moyenne			
min			

# Les fonctions

## *La "minute mathématique"*

fonction	exemple	entrées possibles	sortie
cosinus	$\text{cosinus}(1.047)=0.5$	flottant	flottant
somme	$\text{somme}(2,3)=5$	2 entiers 2 flottants	entier flottant
opposé	$\text{opposé}(4)=-4$	entier flottant	entier flottant
inverse			
différence			
estPositif			
partieEntière			
moyenne			
min			

# Les fonctions

## *La "minute mathématique"*

fonction	exemple	entrées possibles	sortie
cosinus	$\text{cosinus}(1.047)=0.5$	flottant	flottant
somme	$\text{somme}(2,3)=5$	2 entiers 2 flottants	entier flottant
opposé	$\text{opposé}(4)=-4$	entier flottant	entier flottant
inverse	$\text{inverse}(10)=0.1$	flottant	flottant
différence			
estPositif			
partieEntière			
moyenne			
min			

# Les fonctions

## *La "minute mathématique"*

fonction	exemple	entrées possibles	sortie
cosinus	$\text{cosinus}(1.047)=0.5$	flottant	flottant
somme	$\text{somme}(2,3)=5$	2 entiers 2 flottants	entier flottant
opposé	$\text{opposé}(4)=-4$	entier flottant	entier flottant
inverse	$\text{inverse}(10)=0.1$	flottant	flottant
différence	$\text{différence}(2,3)=-1$	2 entiers 2 flottants	entier flottant
estPositif			
partieEntière			
moyenne			
min			

# Les fonctions

## *La "minute mathématique"*

fonction	exemple	entrées possibles	sortie
cosinus	$\text{cosinus}(1.047)=0.5$	flottant	flottant
somme	$\text{somme}(2,3)=5$	2 entiers 2 flottants	entier flottant
opposé	$\text{opposé}(4)=-4$	entier flottant	entier flottant
inverse	$\text{inverse}(10)=0.1$	flottant	flottant
différence	$\text{différence}(2,3)=-1$	2 entiers 2 flottants	entier flottant
estPositif	$\text{estPositif}(-5)=\text{FAUX}$	entier	booléen
partieEntière			
moyenne			
min			

# Les fonctions

## La "minute mathématique"

fonction	exemple	entrées possibles	sortie
cosinus	$\text{cosinus}(1.047)=0.5$	flottant	flottant
somme	$\text{somme}(2,3)=5$	2 entiers 2 flottants	entier flottant
opposé	$\text{opposé}(4)=-4$	entier flottant	entier flottant
inverse	$\text{inverse}(10)=0.1$	flottant	flottant
différence	$\text{différence}(2,3)=-1$	2 entiers 2 flottants	entier flottant
estPositif	$\text{estPositif}(-5)=\text{FAUX}$	entier	booléen
partieEntière	$\text{partieEntière}(5.6)=5$	flottant	entier
moyenne			
min			

$\text{PartieEntière}(10)=\lfloor 10 \rfloor = 10$

$\text{PartieEntière}(5.6)=\lfloor 5.2 \rfloor = 5$

# Les fonctions

## *La "minute mathématique"*

fonction	exemple	entrées possibles	sortie
cosinus	$\text{cosinus}(1.047)=0.5$	flottant	flottant
somme	$\text{somme}(2,3)=5$	2 entiers 2 flottants	entier flottant
opposé	$\text{opposé}(4)=-4$	entier flottant	entier flottant
inverse	$\text{inverse}(10)=0.1$	flottant	flottant
différence	$\text{différence}(2,3)=-1$	2 entiers 2 flottants	entier flottant
estPositif	$\text{estPositif}(-5)=\text{FAUX}$	entier	booléen
partieEntière	$\text{partieEntière}(5.6)=5$	flottant	entier
moyenne	$\text{moyenne}(2,4,6)=4$	3 flottants	flottant
min			



# Les fonctions

## *La "minute mathématique"*

fonction	exemple	entrées possibles	sortie
cosinus	$\text{cosinus}(1.047)=0.5$	flottant	flottant
somme	$\text{somme}(2,3)=5$	2 entiers 2 flottants	entier flottant
opposé	$\text{opposé}(4)=-4$	entier flottant	entier flottant
inverse	$\text{inverse}(10)=0.1$	flottant	flottant
différence	$\text{différence}(2,3)=-1$	2 entiers 2 flottants	entier flottant
estPositif	$\text{estPositif}(-5)=\text{FAUX}$	entier	booléen
partieEntière	$\text{partieEntière}(5.6)=5$	flottant	entier
moyenne	$\text{moyenne}(2,4,6)=4$	3 flottants	flottant
min	$\text{min}(\{6,2,4,3\})=2$	tableau d'entiers tableau de flottants	entier flottant

# Les fonctions

## *La "minute mathématique"*

fonction	exemple	entrées possibles	sortie
somme	$\text{somme}(2,3)=5$	flottant entier	flottant entier
opposé	$\text{opposé}(4)=-4$	flottant	flottant
différence	$\text{différence}(2,3)=-1$	2 entiers 2 flottants	entier flottant

Fonction **différence**  
Entrées : 2 entiers  $a$  et  $b$   
Sortie : entier  
Début  
    Renvoyer ...  
Fin

# Les fonctions

## La "minute mathématique"

fonction	exemple	entrées possibles	sortie
somme	$\text{somme}(2,3)=5$	flottant entier	flottant entier
opposé	$\text{opposé}(4)=-4$	flottant	flottant
différence	$\text{différence}(2,3)=-1$	2 entiers 2 flottants	entier flottant

### Fonction **différence**

Entrées : 2 entiers  $a$  et  $b$

Sortie : entier

Début

    Renvoyer **somme**( $a$ ,**opposé**( $b$ ))

Fin

# Les fonctions

## La "minute mathématique"

fonction	exemple	entrées possibles	sortie
somme	$\text{somme}(2,3)=5$	flottant entier	flottant entier
opposé	$\text{opposé}(4)=-4$	flottant	flottant
différence	$\text{différence}(2,3)=-1$	2 entiers 2 flottants	entier flottant

### Fonction **différence**

Entrées : 2 flottants  $a$  et  $b$

Sortie : flottant

Début

    Renvoyer **somme**( $a$ ,**opposé**( $b$ ))

Fin

# Les fonctions

---

Les **fonctions** = les **algorithmes**

Un ou plusieurs paramètres en entrée...

Une sortie...

# Les fonctions

---

Les **fonctions** = les **algorithmes**

Un ou plusieurs paramètres en entrée...  
ou aucun paramètre en entrée : fonction **Tournoi**.

Une sortie...  
ou aucune sortie : fonction **Tournoi**

# Les fonctions

Les **fonctions** = les **algorithmes**

Un ou plusieurs paramètres en entrée...  
ou aucun paramètre en entrée : fonction **Tournoi**.

Une sortie...  
ou aucune sortie : fonction **Tournoi**

Algorithme **Tournoi**

**Entrées** : aucune

**Sortie** : aucune

**Variables** : tableau de chaînes de caractères *Equipes*, tableaux d'entiers *Points* et *Valeurs*, entiers *Equipe1*, *Equipe2*, *pointsChampion*

Début

1.  $Equipes \leftarrow \{ "PSG", "MHSC", "OL", "OM" \}$
2.  $Points \leftarrow \{ 0, 0, 0, 0 \}$
3.  $Valeurs \leftarrow \{ 3, 4, 2, 2 \}$
4.  $Equipe1 \leftarrow 1$
5. Tant que  $Equipe1 < 5$  faire :
6.      $Equipe2 \leftarrow Equipe1 + 1$
7.     Tant que  $Equipe2 < 5$  faire :
8.          $Points \leftarrow \text{MetPointsAJour}(Equipe1, Equipe2, Points, Valeurs)$
9.          $Equipe2 \leftarrow Equipe2 + 1$
10.     Fin tant que
11.      $Equipe1 \leftarrow Equipe1 + 1$
12. Fin tant que

# Les fonctions

Les **fonctions** = les **algorithmes**

Un ou plusieurs paramètres en entrée...  
ou aucun paramètre en entrée : fonction **Tournoi**.

Une sortie...  
ou aucune sortie : fonction **Tournoi**

Algorithme **Tournoi**

**Entrées** : aucune

**Sortie** : aucune

**Variables** : tableau de chaînes de caractères *Equipes*, tableaux d'entiers *Points* et *Valeurs*, entiers *Equipe1*, *Equipe2*, *pointsChampion*

Début

1. *Equipes*  $\leftarrow$  {"PSG","MHSC","OL","OM"}
2. *Points*  $\leftarrow$  {0,0,0,0}
3. *Valeurs*  $\leftarrow$  {3,4,2,2}
4. *Equipe1*  $\leftarrow$  1
5. Tant que *Equipe1* < 5 faire :
6.     *Equipe2*  $\leftarrow$  *Equipe1* + 1
7.     Tant que *Equipe2* < 5 faire :
8.         *Points*  $\leftarrow$  MetPointsAJour(*Equipe1*, *Equipe2*, *Points*, *Valeurs*)
9.         *Equipe2*  $\leftarrow$  *Equipe2* + 1
10.     Fin tant que
11.     *Equipe1*  $\leftarrow$  *Equipe1* + 1
12. Fin tant que

Matches à faire jouer :  
1-2 1-3 1-4 2-3 2-4 3-4



# Les fonctions

Les **fonctions** = les **algorithmes**

Un ou plusieurs paramètres en entrée...  
ou aucun paramètre en entrée : fonction **Tournoi**.

Une sortie...  
ou aucune sortie : fonction **Tournoi**

Algorithme **Tournoi**

**Entrées** : aucune

**Sortie** : aucune

**Variables** : tableau de chaînes de caractères *Equipes*, tableaux d'entiers *Points* et *Valeurs*, entiers *Equipe1*, *Equipe2*, *pointsChampion*

Début

1.  $Equipes \leftarrow \{ "PSG", "MHSC", "OL", "OM" \}$
2.  $Points \leftarrow \{ 0, 0, 0, 0 \}$
3.  $Valeurs \leftarrow \{ 3, 4, 2, 2 \}$
4.  $Equipe1 \leftarrow 1$
5. Tant que  $Equipe1 < 5$  faire :
6.      $Equipe2 \leftarrow Equipe1 + 1$
7.     Tant que  $Equipe2 < 5$  faire :
8.          $Points \leftarrow \text{MetPointsAJour}(Equipe1, Equipe2, Points, Valeurs)$
9.          $Equipe2 \leftarrow Equipe2 + 1$
10.     Fin tant que
11.      $Equipe1 \leftarrow Equipe1 + 1$
12. Fin tant que

Trace :

Equipe1	1	1	1	2	2	3	4
Equipe2	2	3	4	3	4	4	5

Matches à faire jouer :  
1-2 1-3 1-4 2-3 2-4 3-4

# Les fonctions en Java

---

Toujours une fonction `main()` qui ne renvoie rien

Déclaration des fonctions après la fonction `main()`

# Les fonctions en Java

Toujours une fonction main() qui ne renvoie rien et qui prend en entrée les paramètres du programme

Déclaration des fonctions après la fonction main()

```
public class TP1{
    public static void main(String[] arg) {
        int i,j;
        i=5;
        j=34;
        System.out.print("i+1="+inc(i)+" , i="+i+" ,
            j="+j+" , somme : "+addition(i,j));
    }
    public static int inc(int i){
        i=i+1;
        return i;
    }
    public static int addition(int i, int j){
        return i+j;
    }
}
```

# Les fonctions en Java

Toujours une fonction `main()` qui ne renvoie rien et qui prend en entrée les **paramètres du programme**

Déclaration des fonctions après la fonction `main()`

```
public class TP1{
    public static void main(String[] arg) {
        int i,j;
        i=5;
        j=34;
        System.out.print("i+1
j="+j+", somme : "+ad
    }
    public static int inc(in
        i=i+1;
        return i;
    }
    public static int addition(int i, int j){
        return i+j;
    }
}
```

En ligne de commande :

```
java TP1 toto 1 10.5
```

Le tableau `arg` est alors :

```
{"toto","1","10.5"}
```

# Les fonctions en Java

Toujours une fonction `main()` **qui ne renvoie rien**  
et qui **prend en entrée les paramètres du programme**

Déclaration des fonctions après la fonction `main()`

```
public class TP1{
    public static void main(String[] arg) {
        int i,j;
        i=5;
        j=34;
        System.out.print("i+1="+inc(i)+" , i="+i+" ,
            j="+j+" , somme : "+addition(i,j));
    }
    public static int inc(int i){
        i=i+1;
        return i;
    }
    public static int addition(int i, int j){
        return i+j;
    }
}
```


# Les fonctions en Java

Toujours une fonction `main()` **qui ne renvoie rien**  
et qui **prend en entrée les paramètres du programme**

Déclaration des fonctions après la fonction `main()`

```
public class TP1{
    public static void main(String[] arg) {
        int i,j;
        i=5;
        j=34;
        System.out.print("i+1="+inc(i)+" , i="+i+" ,
        j="+j+" , somme : "+addition(i,j));
    }
    public static int inc(int i){
        i=i+1;
        return i;
    }
    public static int addition(int i, int j){
        return i+j;
    }
}
```

**type de variable renvoyée par la fonction**



# Les fonctions en Java

Toujours une fonction `main()` **qui ne renvoie rien**  
et qui **prend en entrée les paramètres du programme**

Déclaration des fonctions après la fonction `main()`

```
public class TP1{
    public static void main(String[] arg) {
        int i,j;
        i=5;
        j=34;
        System.out.print("i+1="+inc(i)+" , i="+i+" ,
            j="+j+" , somme : "+addition(i,j));
    }
    public static int inc(int i) {
        i=i+1;
        return i;
    }
    public static int addition(int i, int j) {
        return i+j;
    }
}
```

**entrées du programme précédées de leur type**


**type de variable renvoyée par la fonction**

# Les fonctions en Java

Toujours une fonction main() **qui ne renvoie rien**  
et qui **prend en entrée les paramètres du programme**

Déclaration des fonctions après la fonction main()

```
public class TP1{
    public static void main(String[] arg) {
        int i,j;
        i=5;
        j=34;
        System.out.print("i+1="+inc(i)+" , i="+i+" ,
        j="+j+" , somme : "+addition(i,j));
    }
    public static int inc(int i){
        i=i+1;
        return i;
    }
    public static int addition(int i, int j){
        return i+j;
    }
}
```

 **appel de la fonction addition**



# Les fonctions en Java

**Visibilité** des variables : toute variable déclarée à l'intérieur d'une fonction n'est valable **que dans cette fonction** et **ne peut pas être utilisée ailleurs**.

## Variables locales

```
public class TP1{
    public static void main(String[] arg){
        int i,j;
        i=5;
        j=34;
        System.out.print("i+1="+inc(i)+" , i="+i+" ,
        j="+j+" , somme : "+addition(i,j));
    }
    public static int inc(int i){
        i=i+1;
        return i;
    }
    public static int addition(int i, int j){
        return i+j;
    }
}
```

# Les fonctions en Java

**Visibilité** des variables : toute variable déclarée à l'intérieur d'une fonction n'est valable **que dans cette fonction** et **ne peut pas être utilisée ailleurs**.

## Variables locales

```
public class TP1{
    public static void main(String[] arg) {
        int i, j;
        i=5;
        j=34;
        System.out.print("i+1="+inc(i)+" , i="+i+" ,
        j="+j+" , somme : "+addition(i, j));
    }
    public static int inc(int i){
        i=i+1;
        return i;
    }
    public static int addition(int i, int j){
        return i+j;
    }
}
```

**pas la même variable *i*  
même si elles ont la  
même valeur !**



# Les fonctions en Java

**Visibilité** des variables : toute variable déclarée à l'intérieur d'une fonction n'est valable **que dans cette fonction** et **ne peut pas être utilisée ailleurs**.

## Variables locales

```
public class TP1{
    public static void main(String[] arg) {
        int i,j;
        i=5;
        j=34;
        System.out.print("i+1="+inc(i)+" , i="+i+" ,
            j="+j+" , somme : "+addition(i,j));
    }
    public static int inc(int i){
        i=i+1;
        return i;
    }
    public static int addition(int i, int j){
        return i+j;
    }
}
```

**définition** de la fonction addition

# Les fonctions en Java

**Visibilité** des variables : toute variable déclarée à l'intérieur d'une fonction n'est valable **que dans cette fonction** et **ne peut pas être utilisée ailleurs**.

## Variables locales

```
public class TP1{
    public static void main(String[] arg) {
        int i,j;
        i=5;
        j=34;
        System.out.print("i+1="+inc(i)+" , i="+i+" ,
            j="+j+" , somme : "+addition(i,j));
    }
    public static int inc(int i){
        i=i+1;
        return i;
    }
    public static int addition(int i, int j){
        return i+j;
    }
}
```

déclaration de la fonction addition

définition de la fonction addition

# Les fonctions en Java

**Visibilité** des variables : toute variable déclarée à l'intérieur d'une fonction n'est valable **que dans cette fonction** et **ne peut pas être utilisée ailleurs**.

## Variables locales

```
public class TP1{
    public static void main(String[] arg) {
        int i,j;
        i=5;
        j=34;
        System.out.print("i+1="+inc(i)+" , i="+i+" ,
            j="+j+" , somme : "+addition(i,j));
    }
    public static int inc(int i){
        i=i+1;
        return i;
    }
    public static int addition(int i, int j){
        return i+j;
    }
}
```

déclaration de la fonction addition

renvoi du résultat en sortie

définition de la fonction addition

# Les fonctions en Java

**Visibilité** des variables : toute variable déclarée à l'intérieur d'une fonction n'est valable **que dans cette fonction** et **ne peut pas être utilisée ailleurs**.

## Variables locales

```
public class TP1{
    public static void main(String[] arg) {
        int i,j;
        i=5;
        j=34;
        System.out.print("i+1="+inc(i)+" , i="+i+" ,
            j="+j+" , somme : "+addition(i,j));
        }
        appel de la fonction addition
    public static int inc(int i){
        i=i+1;
        return i;
        }
        déclaration de la fonction addition
    public static int addition(int i, int j) {
        return i+j; renvoi du résultat en sortie
    }
}
définition de la fonction addition
```

# Séance d'exercices d'entraînement

---

Une autre avant l'exam en janvier

Possibilité d'envoyer l'exercice supplémentaire par mail

Possibilité d'envoyer ou demander tout autre exercice par mail