

DUT MMI – IUT de Marne-la-Vallée

03/12/2015

M1202 - Algorithmique

Cours 4
Périphériques d'entrée-sortie,
Fonctions

Plan du cours 4 – Les fonctions, les entrées/sorties

- Résumé de l'épisode précédent
- Les entrées/sorties
- Écriture de fonctions en Java, variables locales
- Les fonctions, entrées et sortie
- Combiner plusieurs fonctions pour en créer une nouvelle

Plan du cours 4 – Les fonctions, les entrées/sorties

- Résumé de l'épisode précédent
- Les entrées/sorties
- Écriture de fonctions en Java, variables locales
- Les fonctions, entrées et sortie
- Combiner plusieurs fonctions pour en créer une nouvelle

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 :

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

La boucle “for” / “Pour tout...”

Pour parcourir tous les entiers entre deux valeurs entières.

En pseudo-code :

Pour tout entier i de 1 à 42 faire :

...

Fin Pour

En Java :

```
for (int i=1; i<43; i++) {
```

```
    ...
```

```
}
```

La boucle “for” / “Pour tout...”

Pour parcourir tous les entiers entre deux valeurs entières.

En pseudo-code :

Pour tout entier i de 1 à 42 faire :

...

Fin Pour

En Java :

```
for(int i=1;i<43;i++) {  
    ...  
}
```

En Java avec while :

```
int i=1;  
while (i<43) {  
    ...  
    i++;  
}
```

La boucle “for” / “Pour tout...”

Pour parcourir tous les entiers entre deux valeurs entières.

En pseudo-code :

Pour tout entier i de 1 à 42 faire :

...

Fin Pour

En Java :

```
for(int i=1, i<43, i++) {  
    ...  
}
```

En Java avec while :

```
int i=1;  
while (i<43) {  
    ...  
    i++;  
}
```

Plan du cours 4 – Les fonctions, les entrées/sorties

- Résumé de l'épisode précédent
- Les entrées/sorties
- Écriture de fonctions en Java, variables locales
- Les fonctions, entrées et sortie
- Combiner plusieurs fonctions pour en créer une nouvelle

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

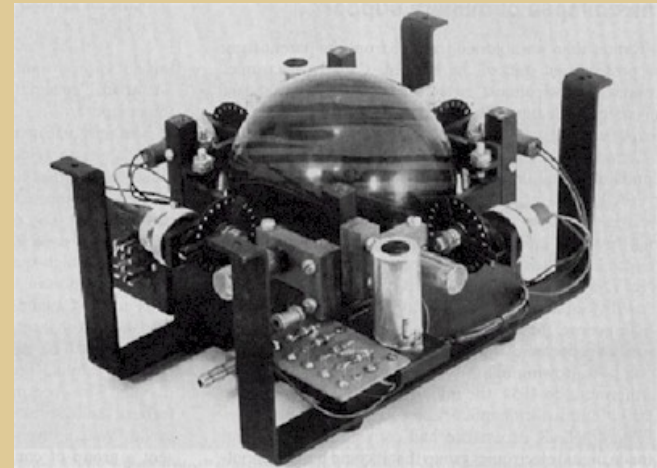
La “minute culturelle”

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

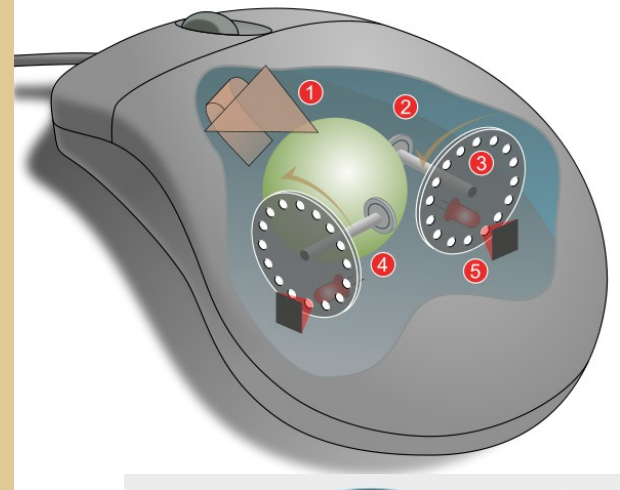
La "minute culturelle"

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 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



0:41 / 2:41 360p

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

en pseudo-code

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 chaîne de caractères reponseALaQuestion (<i>questionAAfficher</i>) affiche la question <i>questionAAfficher</i> et renvoie une chaîne de caractères. |
| Ecran | Si ligne de commande : chaîne de caractères |

Les entrées-sorties

en pseudo-code

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> <p>chaîne de caractères</p> <p>reponseALaQuestion(<i>questionAAfficher</i>)</p> <p>affiche la question <i>questionAAfficher</i> et renvoie une chaîne de caractères.</p> <p>Exemple : reponseALaQuestion("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>Affiche(<i>chaineAAfficher</i>)</p> <p>affiche la chaîne de caractères <i>chaineAAfficher</i> et 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 | chaîne de caractères <code>Scanner lectureClavier = new Scanner(System.in);</code> <code>String chaineLue = lectureClavier.next();</code> LectureClavier.next() renvoie une chaîne de caractères. |
| Ecran | Si ligne de commande : chaîne de caractères |

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 chaineLue = lectureClavier.next();</pre> <p>LectureClavier.next() renvoie une chaîne de caractères.</p> <pre>int entierLu = lectureClavier.nextInt();</pre> <p>LectureClavier.nextInt() renvoie un entier.</p> |
| Ecran | <p>Si ligne de commande : chaîne de caractères</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 chaineLue = lectureClavier.next();</pre> <p>LectureClavier.next() renvoie une chaîne de caractères.</p> <pre>int entierLu = 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 chaineLue = lectureClavier.next();</pre> <p>LectureClavier.next() renvoie une chaîne de caractères.</p> <pre>int entierLu = 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 4 – Les fonctions, les entrées/sorties

- Résumé de l'épisode précédent
- Les entrées/sorties
- Écriture de fonctions en Java, variables locales
- Les fonctions, entrées et sortie
- Combiner plusieurs fonctions pour en créer une nouvelle

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

Une sortie...
ou aucune

Les fonctions en Java – La fonction `main`

Toujours une fonction `main` qui ne renvoie rien

Déclaration des fonctions après la fonction `main`

Les fonctions en Java – La fonction main

Algorithme pour **construire une maison** ?

→ faire appel à un **maître d'oeuvre**

Que fait le maître d'oeuvre ?

→ il réunit les informations sur la maison qui va être construite

→ il trouve des artisans

→ à chaque artisan, il donne des informations sur ce qu'il attend, et récupère le résultat

→ chaque artisan peut lui-même sous-traiter une partie de son travail à un autre artisan : il donne les informations sur ce qu'il attend, et récupère le résultat

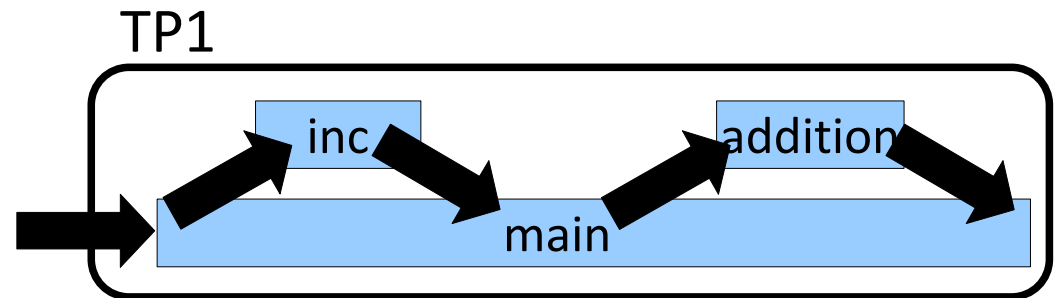
→ la maison se construit

La construction de la maison : le programme TP1

Le maître d'oeuvre : la fonction main

L'artisan 1 : la fonction inc

L'artisan 2 : la fonction addition



```
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 – La fonction `main`

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 – La fonction `main`

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 10.5
```

Le tableau `arg` est alors :

```
{"toto", "1", "10", "10.5"}
```

Les fonctions en Java – La fonction `main`

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 – Autres fonctions

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 – Autres fonctions

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



**entrées de la fonction
précédées de leur type**




Les fonctions en Java – Autres fonctions

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 – Autres fonctions

Seule **la fonction main** s'exécute quand on exécute le programme TP1.

Les **autres fonctions** ne s'exécutent que si elles sont **appelées** pendant l'exécution de 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 inc

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
```

Plan du cours 4 – Les fonctions, les entrées/sorties

- Résumé de l'épisode précédent
- Les entrées/sorties
- Écriture de fonctions en Java, variables locales
- Les fonctions, entrées et sortie
- Combiner plusieurs fonctions pour en créer une nouvelle

Les fonctions

La "minute mathématique"

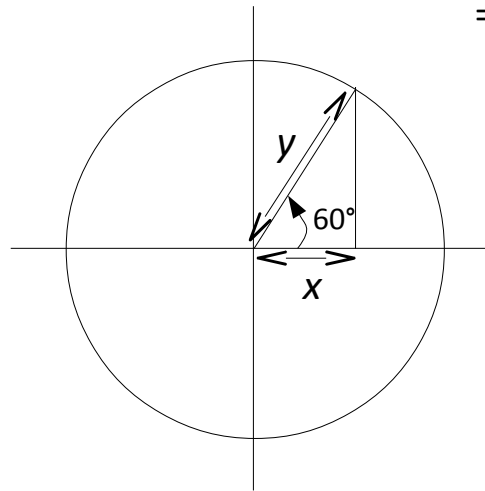
| fonction | exemple | entrées possibles | sortie |
|---------------|-----------------------------|-------------------|--------|
| cosinus | $\text{cosinus}(1.047)=0.5$ | | |
| somme | | | |
| opposé | | | |
| inverse | | | |
| différence | | | |
| estPositif | | | |
| partieEntière | | | |
| moyenne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

| fonction | exemple | entrées possibles | sortie |
|---------------|-----------------------------|-------------------|----------|
| 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 |

Plan du cours 4 – Les fonctions, les entrées/sorties

- Résumé de l'épisode précédent
- Les entrées/sorties
- Écriture de fonctions en Java, variables locales
- Les fonctions, entrées et sortie
- Combiner plusieurs fonctions pour en créer une nouvelle

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

Attention, si on utilise la fonction somme qui renvoie un flottant, le type de sortie n'est pas correct pour la fonction différence !

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