

Algorithmes et Pensée Computationnelle

Fonctions, mémoire et exceptions

Le but de cette séance est d'approfondir vos connaissances en programmation. Au terme de cette séance, l'étudiant sera capable de :

- définir une fonction et l'utiliser dans un programme,
- utiliser des bibliothèques contenant des fonctions prédéfinies,
- connaître quelle est la portée d'une variable,
- comprendre comment fonctionne une pile d'exécution (call stack),
- gérer des exceptions.

1 Variables et Fonctions

Question 1: (🕒 5 minutes) Les fonctions (fonctions basiques) (Java ou Python)

Définissez une fonction nommée `ping()` qui, lorsqu'elle est appelée, affiche "pong".

💡 Conseil

Référez vous aux diapositives du cours pour la création et l'appel des fonctions.

>_ Solution

Python :

```
1 def ping() :  
2     print("pong")  
3  
4 ping()
```

Java :

```
1 public class Main {  
2     static void Ping() {  
3         System.out.println("Pong");  
4     }  
5     public static void main(String[] args) {  
6         Ping();  
7     }  
8 }
```

Question 2: (🕒 5 minutes) Les Fonctions (Fonction multiplication) (Java ou Python)

Définissez une fonction nommée `multiplicateur()` qui prend deux arguments *multiple1* et *multiple2*, les multiplie et retourne le résultat. Stockez le résultat de `multiplicateur(2, 3)` dans une variable *resultat* et affichez la.

💡 Conseil

- Référez vous aux diapositives du cours pour la création et l'appel des fonctions.
- Pour retourner une valeur au lieu de l'afficher, utilisez le mot-clé `return` (pour Python et Java).

>_ Solution

Python :

```
1 def multiplicateur(multiple1, multiple2):
2     return multiple1 * multiple2
3
4
5 if __name__ == "__main__":
6     resultat = multiplicateur(2, 3)
```

Java :

```
1 public class Main {
2     public static int multiplicateur(int multiple1, int multiple2){
3         return multiple1*multiple2;
4     }
5
6     public static void main(String[] args) {
7         int resultat = multiplicateur(1, 2);
8     }
9 }
```

Question 3: (🕒 5 minutes) Les Fonctions (Fonctions Aire et Périmètre) (Java ou Python)

Définissez deux fonctions nommées `aire()` et `perimetre()` qui prennent un argument (`rayon`) et renvoient respectivement l'aire et le périmètre d'un cercle. Stockez les résultats dans des variables `aire` et `perimetre` et affichez le contenu de ces variables.

💡 Conseil

- Référez vous aux diapositives du cours pour la création et l'appel des fonctions.
- Pour retourner une valeur au lieu de l'imprimer, utilisez le mot clé `return` (pour Python et Java).
- Pour rappel, le périmètre d'un cercle s'obtient en utilisant la formule $P = 2 * \pi * r$ et l'aire s'obtient en utilisant la formule $A = r^2 * \pi$.

>_ Solution

Python :

```
1 import math
2
3 def aire(rayon):
4     return (rayon**2)*math.pi
5
6 def perimetre(rayon):
7     return 2*math.pi*rayon
8
9 if __name__ == '__main__':
10     rayon = 10
11     aire = aire(rayon)
12     perimetre = perimetre(rayon)
13     print("L'aire d'un cercle de rayon {} est égale à
14           {}".format(rayon, aire))
15     print("Le périmètre d'un cercle de rayon {} est égal à
16           {}".format(rayon, perimetre))
```

Java :

```
1 public class Main {
2
3     static double aire(int rayon){
4         return Math.pow(rayon, 2)*Math.PI;
5     }
6
7     static double perimetre(int rayon){
8         return 2*Math.PI*rayon;
9     }
10
11     public static void main(String[] args) {
12         int rayon = 5;
13         double aire = aire(rayon);
14         double perimetre = perimetre(rayon);
15         System.out.println("L'aire d'un cercle de rayon "+rayon+" est é
16                             gale à "+aire);
17         System.out.println("Le périmètre d'un cercle de rayon
18                             "+rayon+" est égal à "+perimetre);
19     }
20 }
```

Question 4: (🕒 5 minutes) Portée des variables (Python)

Qu'affiche le programme suivant ?

```
1 x = 2
2
3 def fonction():
4     x = 3
5
6     def fonction2():
7         global x
8         print("x=" + str(x))
9
10    fonction2()
11    print("x=" + str(x))
12
13 print(fonction())
```

Conseil

- Le mot-clé `global` permet d'accéder aux variables globales (définies à l'extérieur de la fonction).
- Soyez attentif au type des éléments retournés par chacune des fonctions.

>_ Solution

x=2
x=3
None

Question 5: (🕒 10 minutes) Portée des variables (Java)

Qu'affiche le programme suivant ?

```
1 public class Main {
2     static int a = 0;
3     public static void f() {
4         a += 2;
5         System.out.println("a = " + a);
6     }
7
8     public static void g() {
9         int b = 3;
10        System.out.println("a = " + a);
11        System.out.println("b = " + b);
12    }
13
14    public static void main(String[] args) {
15        f();
16        g();
17        //System.out.println("b = " + b);
18    }
19 }
```

Que se passerait-il si on décommente la ligne 17 (enlever les //)?

Conseil

Tenir compte de l'endroit où chaque variable est définie et quelles opérations sont réalisées sur celle-ci.

>_ Solution

On obtiendra une erreur de compilation du programme car la variable `b` est créée à l'intérieur de la méthode `g()` et n'existe donc qu'au sein de celle-ci. La variable `b` n'étant pas définie en dehors de la méthode `g()`, celle-ci ne peut pas faire l'objet d'un appel en dehors de la fonction `g()`.

2 Gestion des exceptions

Question 6: (🕒 5 minutes) Qu'affiche le programme suivant ? Types d'erreurs (Python)

```
1 value1 = "Algorithms"
2 value2 = 4
3
4 try:
5     size = len(value1)
```

```

6     result = size/value2
7     print(f"Le résultat de la division est: {result}")
8
9 except Exception as error:
10    print("On ne peut pas effectuer l'opération")
11
12 try:
13     result = value1/2
14     print(f"Le résultat de la division est: {result}")
15
16 except TypeError as error:
17    print("On ne peut pas diviser une chaîne de caractères")

```

Conseil

Utilisée sur une chaîne de caractères, la fonction `len()` renvoie le nombre de caractères de la chaîne.

Solution

Le résultat de la division est: 2.5
On ne peut pas diviser une chaîne de caractères.

Question 7: (🕒 5 minutes) Qu'affiche le programme suivant ? **Types d'erreurs (Python)**

```

1 value1 = 4
2 value2 = ""
3
4 try:
5     count = len(value2)
6     result= value1/count
7 except ZeroDivisionError as error:
8     print("Nous ne pouvons pas diviser un nombre par 0")

```

Conseil

La chaîne de caractères vide est représentée par `""`

Solution

Nous ne pouvons pas diviser un nombre par 0.

Question 8: (🕒 5 minutes) Qu'affiche le programme suivant ? **Types d'erreurs (Python)**

```

1 value1 = "Algorithms"
2 value2 = 4
3
4
5 try:
6     decimal = float(value2)
7 except ValueError as error:
8     print("Nous ne pouvons pas convertir un entier en décimal")
9
10 finally:
11     try:
12         value2 = int(value1)
13     except ValueError as error:
14         print("Nous ne pouvons pas convertir une chaîne de caractères en
    nombre")

```

💡 Conseil

- La fonction `float()` permet de convertir une variable en nombre à décimal.
- La fonction `int()` permet de convertir une variable en nombre entier.

>_ Solution

Nous ne pouvons pas convertir une chaîne de caractères.

Question 9: (🕒 5 minutes) **Gestion d'erreurs (Java)** Le programme suivant est incorrect. Que devez-vous modifier pour qu'il fonctionne correctement ?

```
1 public class Question9 {
2
3     public static void division(int a, int b) throws ArithmeticException{
4         if(b==0) {
5             throw new ArithmeticException();
6         }else{
7             float result = a/b;
8             System.out.println("Le résultat de la division de " + a + "/" + b
9 + " = " + result);
10        }
11    }
12
13    public static void main(String args[]){
14        int value1 = 2;
15        int value2 = 4;
16        try {
17            division(value1,value2);
18            value2 = 0;
19            division(value1,value2);
20        }catch (IndexOutOfBoundsException err){
21            System.out.println("Nous ne pouvons pas effectuer une division par
22            0");
23        }
24    }
25 }
```

💡 Conseil

Référez-vous à la diapositive 26 du cours de cette semaine.

>_ Solution

À l'intérieur de la fonction `division`, lorsque `b` est égal à 0, le programme lève une exception de type `ArithmeticException`. Mais dans le `main`, l'erreur qui est interceptée est `IndexOutOfBoundsException`. Pour corriger cette erreur, une des solutions est de remplacer `IndexOutOfBoundsException` par `ArithmeticException`.