Algorithmes et Pensée Computationnelle

Fonctions, gestion de la mémoire et des exceptions - Exercices de base

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

- Utiliser des librairies contenant des fonctions prédéfinies,
- définir une fonction et l'utiliser dans un programme,
- connaître quelle est la portée d'une variable,
- comprendre comment fonctionne la gestion de la mémoire,
- 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". Appelez la plusieurs fois et observez le résultat.

Conseil

>_ Solution

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

Python: def ping(): 2 print("pong") ping() ping() Java: public class Main { static void Ping(){ 3 System.out.println("Pong"); 4 5 public static void main(String[] args) { Ping(); 7 Ping();

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

Définissez une fonction nommée multiplicateur() qui prend deux arguments *multiple_1* et *multiple_2*, les multiplie et retourne le résultat. Stockez le résultat de multiplicateur(2,3) dans une variable *resultat* et affichez la.

Conseil

8 9 }

- Référez vous au 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).

>_ Solution

Python:

Java:

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

Définissez deux fonctions nommées aire() et perimètre() 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 au 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:

Java:

Question 4: (O 5 minutes) **Portée des variables**

Qu'affiche le programme suivant?

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

Consei

- 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

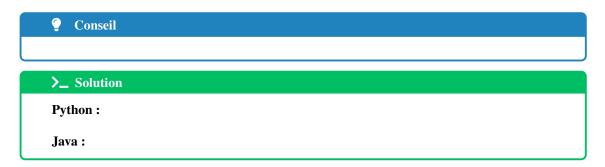
x=3

None

2 Gestion de la mémoire

Question 5: (5 minutes) Template (langage à utiliser)

Exercice 1 - structure



3 Gestion des exceptions

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

```
value1 = "Algorithms"
     value2 = 4
 3
 4
     try:
 5
       size = len(value1)
 6
       result = size/value2
       print(f"Le résultat de la division est: {result}")
 7
     except Exception as error:
 9
10
       print("On ne peut pas effectuer l'opération")
11
12
     try:
13
       result = value 1/2
       print(f''Le résultat de la division est: {result}'')
14
15
16
     except TypeError as error:
       print("On ne peut pas diviser une chaine de caractère")
17
```

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: (**1** *5 minutes*) Qu'affiche le programme suivant? **Types d'erreurs (Python)**

```
value1 = "Algorithms"
 2
     value2 = 4
 3
 4
 5
 6
       decimal = float(value2)
 7
     except ValueError as error:
       print("Nous ne pouvons pas convertir un entier en décimal")
 8
 9
10
     finally:
11
       trv:
         value2 = int(value1)
12
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*) Est ce que le programme suivant s'exécute correctement? Si la réponse est non, expliquez pourquoi et comment vous corrigerez le programme? **Gestion d'erreurs (Java)**

```
public class Question9 {
 1
 3
       public static void division(int a, int b) throws ArithmeticException{
 4
          if(b==0)
 5
            throw new ArithmeticException();
 6
7
          }else{
            float result = a/b;
 8
            System.out.println("Le résultat de la division de " + a + "/" + b + "= " + result);
 9
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
          \big\} catch (IndexOutOfBoundsException\ err) \big\{
            System.out.println("Nous ne pouvons pas effectuer une division par 0");
21
22
23
     }
```

Conseil

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

>_ Solution

Le programme suivant est faux, le type d'erreur que la fonction division() retourne est différent de celui qu'on intercepte dans le catch.

Pour corriger cela, il suffit de remplacer IndexOutOfBoundsException par ArithmeticException, qui correspond au type d'erreur retourné par la fonction division.