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 librairies contenant des fonctions prédéfinies,
- connaître quelle est la portée d'une variable,
- comprendre comment fonctionne une pile d'exécution (call stack)

Attention

Les exceptions n'ayant pu être présentées en cours, les exercices portant sur cette notion seront inclus dans les feuilles d'exercices de la semaine prochaine.

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 :

```python
def ping() :
    print("pong")

ping()
```

Java :

```java
public class Main {
    static void Ping(){
        System.out.println("Pong");
    }
    public static void main(String[] args) {
        Ping();
    }
}
```

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 :

```python
def multiplicateur(multiple1, multiple2):
    return multiple1 * multiple2


if __name__ == "__main__":
    resultat = multiplicateur(2, 3)
```

Java :

```java
public class Main {
    public static int multiplicateur(int multiple1, int multiple2){
        return multiple1*multiple2;
    }

    public static void main(String[] args) {
        int resultat = multiplicateur(1, 2);
    }
}
```

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 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 :

```python
import math

def aire(rayon):
  return (rayon**2)*math.pi

def perimetre(rayon):
  return 2*math.pi*rayon

if __name__ == '__main__':
    rayon = 10
    aire = aire(rayon)
    perimetre = perimetre(rayon)
    print("L'aire d'un cercle de rayon {} est égale à {}".format(rayon, aire))
    print("Le périmètre d'un cercle de rayon {} est égal à {}".format(rayon, perimetre))
```

Java :

```java
public class Main {

    static double aire(int rayon){
        return Math.pow(rayon, 2)*Math.PI;
    }

    static double perimetre(int rayon){
        return 2*Math.PI*rayon;
    }

    public static void main(String[] args) {
        int rayon = 5;
        double aire = aire(rayon);
        double perimetre = perimetre(rayon);
        System.out.println("L'aire d'un cercle de rayon "+rayon+" est égale à "+aire);
        System.out.println("Le périmètre d'un cercle de rayon "+rayon+" est égal à "+perimetre);
    }
}
```

Question 4: (*5 minutes*) **Portée des variables** (Python)
Qu'affiche le programme suivant ?

```python
x = 2

def fonction():
    x = 3

    def fonction2():
        global x
        print("x=" + str(x))

    fonction2()
    print("x=" + str(x))

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()`.