

Exercice_22-params

SimpleCalculationWithTwoInts
+ printAll(a : int, b : int) : void
+ calculateSumMultiple(a : int, b : int, multiple : int) : int
+ calculateSumOdd(a : int, b : int) : int
+ calculateSumEven(a : int, b : int) : int
+ calculatePower(a : int, b : int) : double
+ GCD_search(a : int, b : int) : int
+ GCD_Euclid(a : int, b : int) : int
+ GCD_OptimizedEuclid(a : int, b : int) : int
+ LCM_search(a : int, b : int) : int
+ LCM_shortcut(a : int, b : int) : int

Créez la classe `SimpleCalculationsWithTwoInts` dans le projet `Exercice-22_params`. La classe n'a pas d'attributs.

- Définissez la méthode `printAll` qui affiche tous les entiers compris entre les valeurs des paramètres `a` et `b` (valeurs incluses).
- Définissez la méthode `calculateSumMultiples` qui calcule et retourne la somme de tous les entiers **multiples** de la valeur du paramètre `multiple` compris entre les valeurs des paramètres `a` et `b` (valeurs incluses).
- Définissez la méthode `calculateSumOdd` qui calcule et retourne la somme de tous les entiers **impairs** compris entre les valeurs des paramètres `a` et `b` (valeurs incluses).
- Définissez la méthode `calculateSumEven` qui calcule et retourne la somme de tous les entiers **pairs** compris entre les valeurs des paramètres `a` et `b` (valeurs incluses).
- Définissez la méthode `calculatePower` qui calcule et retourne la puissance a^b à l'aide de multiplications successives (donc : ne pas utiliser `Math.power()` !)
- Définissez la méthode `GCD_search` qui calcule et retourne le PGCD (plus grand commun diviseur) des valeurs des paramètres `a` et `b` en utilisant la méthode par essais successifs.
- Définissez la méthode `GCD_Euclid` qui calcule et retourne le PGCD (plus grand commun diviseur) des valeurs des paramètres `a` et `b` en utilisant l'algorithme d'Euclide par soustractions.
- Définissez la méthode `GCD_OptimizedEuclid` qui calcule et retourne le PGCD (plus grand commun diviseur) des valeurs des paramètres `a` et `b` en utilisant l'algorithme d'Euclide par division et reste.
- Définissez la méthode `LCM_search` qui calcule et retourne le PPCM (plus petit commun multiple) des valeurs des paramètres `a` et `b` en utilisant la méthode par essais successifs.
- Définissez la méthode `LCM_shortcut` qui calcule et retourne le PPCM (plus petit commun multiple) des valeurs des paramètres `a` et `b` en utilisant le raccourci mathématique.

Note : vous pouvez aussi utiliser `pA` et `pB` (`pMultiple`) comme noms de paramètres !