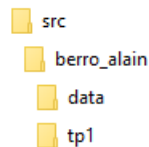


Programmation Impérative en Java

Consignes

- Vous êtes libres d'utiliser l'environnement de développement (IDE) que vous souhaitez. Les enseignants ne vous apporteront aucune aide sur ce sujet.
- Vous développerez avec le JDK 21 ou supérieur donc voici la [documentation](#).
- Lors de la remise de travaux, vous nous donnerez simplement une archive (.zip) du répertoire "src" contenant les sources de votre code. Celui-ci devra être structuré obligatoirement comme le schéma ci-dessous. "src" contiendra un répertoire avec votre "nom_prénom". Ce dernier pourra contenir éventuellement un répertoire "data" pour le stockage de fichiers de données utiles à votre travail et des répertoires comme "tp1" contenant uniquement les sources de vos codes Java.



- Dans tous les exercices ci-après, vous développerez un sous-programme et vous le testerez. Pour cela vous devez créer une classe nommée "AppProgImpJava" dans votre répertoire "tp1" et structurée de la manière suivante :

```

package berro_alain.tp1;

public class AppProgImpJava
{
    /*
     * Zone pour les sous-programmes.
     */

    /**
     * Retourne si un nombre entier est diviseur d'un autre.
     */
    public static boolean estDiviseur (int nb, int diviseur)
    {
        return nb % diviseur == 0;
    }

    /*
     * Programme principal.
     */
    public static void main (String[] args)
    {
        /*----- Zone pour tester vos sous-programmes -----*/

    }

} /*----- Fin de la classe AppProgImpJava -----*/
  
```


Travail à faire

Tous les exercices doivent être réalisés, codés proprement, commentés et correctement testés dans le programme principal.

Exercices

Réalisez les sous-programmes suivants :

- 1) Retourne si un nombre entier est diviseur d'un autre.
 - 2) Retourne le plus grand de trois nombres réels donnés en paramètres.
 - 3) Convertit une durée, donnée en secondes, en heures, minutes et secondes et retourne le résultat sous forme d'une chaîne de caractères. Par exemple, 8036 s = 2 h 13 m 56 s, 3645 s = 1 h 45 s, 1450 s = 24 m 10 s.
[String]
 - 4) Retourne la somme des nombres divisibles par 7 compris dans un intervalle donné.
[for, invocation]
 - 5) Détecte le premier chiffre et le dernier chiffre d'une chaîne de caractères et retourne le nombre entier formé par ces 2 chiffres auxquels on ajoute 1. Par exemple : "a1b2c3d4e5f" donne le nombre 15+1=16, "pqr3stu8vwx" 39 et "treb7uchet" 77+1=78. Supposez que la chaîne a au moins un chiffre.
[charAt, Integer.parseInt()]
 - 6) Retourne si un nombre donné est premier.
[while]
 - 7) Recherche et affiche tous les nombres premiers compris dans un intervalle donné.
[invocation]
 - 8) Retourne le salaire d'un employé en fonction de son grade et de son ancienneté (sujet du cours).
[switch]
 - 9) Vérifie si un nombre entier, représentant un code de carte bleu de 16 chiffres, est valide au sens de l'[algorithme de Luhn](#).
[long]

L'algorithme de Luhn somme chacun des chiffres du code de carte bleue, en multipliant par deux un chiffre sur deux à partir de l'avant dernier et en se déplaçant de droite à gauche. Si un de ces doubles est plus grand que neuf (par exemple 2×8=16) alors il faut le ramener à un chiffre compris entre 1 et 9 en lui soustrayant 9.
Ensuite, l'algorithme effectue la somme de tous les chiffres obtenus. Si le dernier chiffre de cette somme est nul, alors le nombre original est valide.
Pour tester votre algo: 972487086 est valide, 8763 est valide et 927487086 est non valide.
-  de transformer le nombre entier (long) en String ou en tableau.
- 10) Retourne le pgcd de deux nombres entiers a et b sachant que $\text{pgcd}(a,a) = a$ et si $a > b$ alors $\text{pgcd}(a,b) = \text{pgcd}(a-b,b)$ sinon $\text{pgcd}(a,b) = \text{pgcd}(a,b-a)$.
[récursivité]

Exercices sur les tableaux

Créez une nouvelle classe Java, développez les sous-programmes ci-dessous et testez-les. Nous travaillerons avec des tableaux d'entiers.

- 11) Crée et retourne un tableau initialisé aléatoirement avec des nombres compris dans un intervalle donné. Le nombre d'éléments du tableau est passé en paramètre.

[`Math.random()`]

- 12) Retourne le contenu du tableau sous forme d'une chaîne de caractères.

[`String` ou `Arrays.toString()`]

- 13) Retourne la copie exacte d'un tableau passé en paramètre.

[`for` ou `System.arraycopy()`]

- 14) Retourne le nombre d'occurrences d'un nombre donné dans un tableau.

- 15) Retourne la position de la plus petite valeur dans un tableau.

- 16) Vérifie et retourne si un tableau donné est trié par ordre croissant.