

L'objectif de ce TP est de vous entraîner à la déclaration et l'utilisation de vecteurs en ArrayList, ainsi qu'au codage de premiers algorithmes.

## Avant de commencer...

- Lisez entièrement ce sujet
- Ouvrez un terminal et placez-vous dans votre répertoire R1.01
- Lancez **IJ** avec la commande **idea**, puis créez un nouveau projet **TP5\_A**

## Partie A : Vecteurs de Integer

### A1. Déclaration et initialisation

1.1. Dans le projet TP5\_A créez une classe java Vecteurs\_de\_Integer

1.2. Dans cette classe, ajoutez et codez les fonctions suivantes :

a) Saisie contrôlée d'un entier strictement positif

```
public static int saisieEntPos() {  
    // {} => {résultat = un entier > 0, saisi par l'utilisateur}
```

b) Vecteur d'entiers contenant tous les diviseurs d'un entier donné, strictement positif

```
public static ArrayList<Integer> lesDiviseurs(int unEnt) {  
    //{unEnt > 0} => {résultat = vecteur contenant les diviseurs de unEnt  
    // y compris 1 et unEnt}
```

1.3. Insérez une procédure **main**, dans laquelle vous ajouterez :

- ✓ la déclaration d'une variable de type **int**, nommée **unEnt** et initialisée par appel de la fonction **saisieEntPos()**
- ✓ la déclaration d'un **ArrayList<Integer>**, nommé **lesDiv**
- ✓ les instructions suivantes :
  - initialisation de **lesDiv** avec les diviseurs de **unEnt**
  - affichage des diviseurs de **unEnt** (↔ affichage de **lesDiv**)

1.4. Exécutez et testez

EXEMPLES DE TRACES D'EXÉCUTION

```
Entrez un entier strictement positif : 28  
Diviseurs de 28 : [1, 2, 4, 7, 14, 28]
```

```
Entrez un entier strictement positif : -5  
-> saisie invalide, recommencez...  
Entrez un entier strictement positif : 100  
Diviseurs de 100 : [1, 2, 4, 5, 10, 20, 25, 50, 100]
```

### A2. Utilisation : Entiers "particuliers"

ENTIERS POSITIFS PARTICULIERS

- Un entier positif est **PREMIER** s'il n'est divisible que par 1 et par lui-même - *Exemple : 29*
- Un entier positif est **PARFAIT** s'il est égal à la moitié de la somme de ses diviseurs – *Exemple : 28*
- Un entier positif est **SUBLIME** si le nombre de ses diviseurs est un nombre parfait et si la somme de ses diviseurs est aussi un nombre parfait – *Exemple : 12*

2.1. Complétez la classe Vecteurs\_de\_Integer avec les fonction suivantes :

a) Somme des éléments d'un ArrayList de Integer, ne contenant que des entiers strictement positifs

```
public static int getSommeVectEntPos(ArrayList<Integer> unVectEntPos) {  
    //{unVectEntPos non vide, tous ses éléments sont > 0}  
    //    => {résultat = somme des éléments de unVectEntPos}  
}
```

b) Entier "premier"

```
public static boolean nombrePremier(int unEnt) {  
    //{unEnt > 0} => {résultat = vrai si unEnt est un nombre premier}  
}
```

c) Entier "parfait"

```
public static boolean nombreParfait (int unEnt) {  
    //{unEnt > 0} => {résultat = vrai si unEnt est un nombre parfait}  
}
```

d) Entier "sublime"

```
public static boolean nombreSublime(int unEnt) {  
    //{unEnt > 0} => {résultat = vrai si unEnt est un nombre sublime}  
}
```

2.2. Complétez la procédure main de Vecteurs\_de\_Integer avec :

- ✓ l'affichage du nombre de diviseurs de unEnt
- ✓ l'affichage d'un message exprimant unEnt est *premier*, *parfait* ou *sublime*, ou s'il n'a aucune de ces "qualités"...

2.3. Exécutez et testez...

## Partie B : Vecteur de String

### B1. Déclaration et initialisation

1.1. Dans le projet TP5\_A créez une classe java Vecteurs\_de\_String

1.2. Insérez une procédure main dans laquelle vous déclarerez et initialiserez par lot :

- ✓ un ArrayList de String, nommé couleursFR, initialisé avec les 7 couleurs de l'arc en ciel, en français (violet, indigo, bleu, vert, jaune, orange et rouge)
- ✓ un ArrayList de String, nommé colorsEN, initialisé avec les 7 couleurs de l'arc en ciel, en anglais et dans le même ordre que les couleurs en français (purple, indigo, blue, green, yellow, orange et red)

1.3. Ajoutez les instructions d'affichage de ces deux vecteurs puis testez

### B2. Recherche dans un vecteur de String

1.1. Dans la classe Vecteurs\_de\_String déclarez et codez la fonction suivante :

```
public static int getIndice(String ch, ArrayList<String> vString) {  
    //{vString non vide, ne contenant que des chaînes différentes}  
    //    => {résultat = indice de ch dans vString, si ch trouvé,  
    //    sinon, résultat = vString.size()}  
}
```

1.2. Dans la procédure main, ajoutez une boucle où :

- ✓ l'utilisateur sera invité à saisir une couleur parmi les couleurs de l'arc en ciel en français
- ✓ dans le cas où il a respecté les consignes : le nom anglais de cette couleur sera affiché
- ✓ dans le cas où il n'a pas respecté les consignes : un message l'avertira de son "erreur"

L'exécution de cette boucle sera contrôlée par la valeur d'un caractère, que l'utilisateur sera invité à saisir en fin de chaque itération (exemple : 'o' pour continuer / 'n' pour arrêter)

1.3. Exécutez et testez