

IAD : TP4.1 — Implémentation Java

BUT/INFO

version 2021-2022

Table des matières

- [1. Émulation de table traçante](#)
- [2. Des sapins couchés](#)
- [3. Pour aller plus loin : matrices dans un écran barré](#)

Problème

Écrire 3 programmes en langage Java permettant de dessiner divers sapins de hauteur h donnée. Voici quelques exemples pour $h=5$. La hauteur sera saisie et comprise entre 0 et 40.

- Un sapin plein : programme **SapinPlein**

```
      *
     ***
    *****
   *********
  ***********
 *****
```

- Un sapin vide : programme **SapinVide**

```
      *
     * *
    *   *
   *     *
  *       *
 *         *
*****
```

- Un sapin couché : programme **SapinCouche**

```
*
**
***
****
*****
****
***
**
*

```

1. Émulation de table traçante

- Commencer par créer les commandes de base sous forme de fonctions `star()`, `right()` et `next()`.
- Enrichir les vocables qui vous sont accessibles en créant
 - la fonction `n_star`, qui prend un paramètre un entier `n` et qui affiche `n` étoiles
 - la fonction `n_right`, qui prend un paramètre un entier `n` et qui se déplace vers la droite de `n` caractères.
- Créer la fonction `genererSapinPlein`, puis la fonction `genererSapinVide`.

2. Des sapins couchés

À la manière de la matrice écran du TP3 précédent, nous voulons dans un premier temps générer une chaîne de caractères représentant le sapin, que l'on affichera éventuellement dans un second temps.

- Créer la fonction `genererStringLigne` telle que décrite dans la séance précédente. La tester.
- Créer la fonction `genererSapinCouche` prenant en paramètre un entier `h`, qui génère la chaîne de caractère correspondant à un sapin couché de hauteur `h`. La tester.

3. Pour aller plus loin : matrices dans un écran barré

Créer une nouvelle fonction `genererStringMatriceEcranBarre` (et éventuellement d'autres fonctions pouvant aider) qui puisse nous permettre de réaliser l'affichage suivant :

```
*****
**0000000000**
*0*00000000*0*
*00*000000*00*
*000*0000*000*
*0000*00*0000*
*00000*00000*
*000000*00000*
*0000000*00000*
*00000000*0000*
*000000000*000*
*0000000000*00*
*00000000000*0*
**00000000000**
*****
```