

## TP2

---

### Tableaux une dimension

---

**Exercice 1 – Les températures.** On souhaite définir une classe *Temperatures* permettant de stocker, pour un mois, des relevés de températures effectués chaque jour. On utilisera pour mémoriser les relevés un tableau d'entiers.

Programmer et tester les méthodes ci-dessous dans une méthode `public static void main(String[] args)` :

- Écrire un constructeur permettant d'initialiser un objet avec le nom d'un mois et le nombre de jours de ce mois.
- Écrire un deuxième constructeur ayant pour seul paramètre le nom du mois, son nombre de jours ayant par défaut la valeur 31.
- Définir des accesseurs permettant d'une part d'obtenir le nombre d'éléments du tableau, et d'autre part d'accéder au  $i^{me}$  élément du tableau.
- Définir un accesseur permettant d'accéder au tableau dans sa totalité, en prenant soin de recopier explicitement les données.
- Définir des modificateurs pour :
  - Donner une valeur au  $i^{me}$  élément du tableau,
  - Initialiser globalement le tableau à partir d'un tableau déjà construit (les éventuelles anciennes valeurs seront alors perdues).
- Pour optimiser les allocations, définir un nouveau constructeur qui prendra en paramètres le nom du mois et un tableau de relevés déjà construit utilisé pour l'initialisation.
- Écrire une méthode `public int moyenne()` permettant de calculer la moyenne des températures sur un mois.
- Redéfinir la méthode `public String toString()` pour permettre l'affichage des températures selon le format suivant :  
`Nomdumois[moyenne] : tempJ1, tempJ2, tempJ3, ..., tempDernierJour`

**Exercice 2 – Tris.** Programmer et tester les algorithmes de tri suivants :

- Le tri à bulle,
- Le tri insertion,
- Le tri fusion,
- Le tri rapide.

Vous trouverez toutes les informations utiles pour les différents tris ici : [https://fr.wikipedia.org/wiki/Algorithme\\_de\\_tri#Exemples\\_d.27algorithmes\\_de\\_tri](https://fr.wikipedia.org/wiki/Algorithme_de_tri#Exemples_d.27algorithmes_de_tri)

**Exercice 3 – Triangles de Pascal le retour.** On peut aussi calculer le triangle de Pascal avec un tableau.

La valeur d'un élément du triangle de Pascal est égale à la somme de l'élément au-dessus à gauche et juste au-dessus.

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
    
```

Écrire un programme qui calcule le triangle de Pascal à l'aide de deux tableaux à une dimension.

---

### Tableaux à 2 dimensions

---

**Exercice 4 – Températures le retour.** Définir une classe `TemperaturesAnnuelles` dans laquelle les relevés seront stockés sous la forme d'un tableau de `Temperatures` (chaque mois correspondant à un objet de l'exercice 1). Les relevés sont transmis chaque mois sous la forme d'un tableau d'entiers.

- Écrire une méthode permettant d'ajouter un nouveau relevé à la suite du dernier.
- Écrire une méthode `public int moyenne()` qui affiche la température moyenne sur l'année.
- Écrire une méthode `public String toString()` qui affiche les températures de l'année.

**Exercice 5 – Puissance 4.** Le jeu Puissance 4 est connu. Si vous ne le connaissez pas Wikipédia vous aidera. Écrire une classe permettant à deux joueurs de s'affronter. La taille de la grille est de 6 rangées par 7 colonnes.

La classe `LireClavier` présente dans le git vous permet de récupérer ce que le joueur va saisir au clavier. La méthode `int lireEntier()` retourne l'entier saisi par l'utilisateur.

- Créer la classe `GameP4` ayant en attribut :
  - un tableau représentant la grille du jeu,
  - un booléen représentant le joueur en train de jouer.
- Écrire un constructeur qui initialise une nouvelle partie.
- Écrire une méthode `private void placerPion()` qui demande au joueur de saisir la colonne où il souhaite jouer, place le pion dans la grille en prenant en compte la gravité. S'il n'y a plus de place dans cette colonne, on demande au joueur de saisir une nouvelle colonne.
- Écrire une méthode `private boolean victoire()`, qui détecte si un joueur a gagné.
- Écrire une méthode `public boolean jouer()`, qui permet à deux joueur de s'affronter. Si le joueur gagne cette méthode retourne `true`.

On souhaite maintenant, permettre aux joueurs de faire plusieurs parties et de comptabiliser leur scores. Écrire une nouvelle classe `Puissance4` proposant cela.