

Le tutorat commence la semaine du 3 février, les détails sont sur DidEL.

1 Personnes

On définit :

- une classe `Date` qui aura trois attributs privés `jour`, `mois` et `annee` de type entier, et
 - une classe `Personne` qui aura deux attributs privés `anniversaire` de type `Date` et `nom` de type `String`.
1. Écrivez la classe `Date` avec un constructeur public sans argument qui choisit des valeurs aléatoires pour les trois attributs. Les valeurs choisies doivent être correctes, par exemple la valeur du `mois` est comprise entre 1 et 12.
 2. Écrivez la classe `Personne` avec le même type de constructeur.
 3. Écrivez une méthode publique de prototype `public boolean egal(Date d)` qui teste si la date `this` représente le même jour de l'année que la date `d`.

2 Tableaux de personnes de taille variable

On définit une classe `TabPersonnes` pour représenter des tableaux de taille variable contenant des valeurs de type `Personne`.

1. Écrivez la classe `TabPersonnes` avec les attributs et le constructeur public adaptés.
2. Écrivez des méthodes publiques permettant d'accéder à un élément du tableau et d'ajouter des éléments au tableau.
3. Que faut-il modifier pour obtenir une classe `TabDates` qui représente des tableaux de valeurs de type `Date` ?

3 Le problème des anniversaires

Écrivez trois programmes, permettant de stocker en mémoire des personnes choisies aléatoirement et de répondre, de manière expérimentale, aux questions suivantes :

1. Combien de personnes faut-il choisir aléatoirement avant de trouver trois personnes qui sont nées le même jour de l'année ?
2. On suppose qu'on choisit aléatoirement 365 personnes. Combien d'anniversaires différents auront-ils ? (on ignore l'année de naissance)
3. Combien de personnes faut-il choisir aléatoirement avant de trouver pour chaque jour de l'année au moins une personne qui est née ce jour-là.

Chaque programme devra simuler le choix aléatoire des personnes et la vérification de leurs anniversaires. Faites attention à la manière dont vous stockez dans la mémoire les personnes choisies aléatoirement.

4 Afficher des mots

1. Écrivez une méthode `fusion` qui à un tableau de taille variable (i.e., `ArrayList`) de chaînes de caractères, ajoute les éléments d'un tableau de taille fixe de chaînes de caractères (i.e., `String[]`). On suppose que les éléments du `ArrayList` reçu comme paramètre sont triés dans l'ordre alphabétique et on demande que la même propriété soit vraie à la sortie de cette fonction. Vous pouvez utiliser la méthode `compareTo` de la classe `String` :
 - `int compareTo(String anotherString)` : compare `this` avec `anotherString` par rapport à l'ordre alphabétique : la chaîne de caractères `this` est avant `anotherString` si le résultat est négatif et après sinon.
2. Écrivez un programme qui demande à l'utilisateur de rentrer des phrases (jusqu'au moment où il tape "stop") et qui affiche une liste alphabétique de tous les mots écrits par l'utilisateur. Tous les mots doivent être convertis en minuscules. Utilisez `ArrayList` pour stocker les mots. Les méthodes suivantes de la classe `String` peuvent vous aider :
 - `String[] split(String sep)` : renvoie le tableau des sous-chaînes de `this` délimitées par la chaîne de caractères donnée comme paramètre. En particulier, `w.split("\\s+")` renvoie le tableau des sous-chaînes de `w` délimitées par *des* espaces.
 - `String toLowerCase()` : renvoie la conversion en minuscules du mot `this`.