

Département d'Informatique

Informatique 2 SPC

2016-2017

TD1 – Types composés et tris

1. Mesures de températures

- Définir un type **TemperaturesCaptees** composé de 4 champs : un entier représentant l'**identifiant** unique du capteur ayant réalisé les mesures, de trois réels représentant (en °C) les mesures sur une journée de la **température maximum**, la **température minimum** et la **température à midi**. Pensez à définir un ou des constructeurs.
- Écrire la fonction **saisirTemperaturesCaptees** qui demande à l'utilisateur de saisir au clavier les valeurs et qui retourne cette donnée. On s'assurera que les différentes températures saisies ne sont pas en dehors de la place [-50°C,100°C] et que la température maximum est supérieure ou égale à la température minimum et que la température à midi est bien comprise entre la température maximum et la température à minimum.
- Écrire la fonction **ecartMaxTemperaturesCaptees** qui prend en entrée une donnée de type **TemperaturesCaptees** et retourne la valeur absolue de la différence entre la température maximum et la température minimum.
- Écrire la fonction **afficheTemperaturesCaptees** qui prend en entrée une donnée de type **TemperaturesCaptees** et affiche celle-ci sous la forme :

`Id_capteur   *   temp à midi   *   temp max   *   temp min   *   ecartMax`

2. Températures d'une ville

- Définir le type **TemperaturesVille** composé de 3 champs : une chaîne représentant le **nom de la ville**, un entier représentant le **nombre de capteurs installés** dans la ville (supérieur ou égal à 1) et un tableau de **TemperaturesCaptees** représentant **les températures mesurées** par les différents capteurs. Pensez à définir un ou des constructeurs.
- Écrire la fonction **saisirTemperaturesVille** qui demande à l'utilisateur de saisir au clavier le nom de la ville, le nombre de capteurs (supérieur ou égal à 1) et toutes les températures mesurées et qui retourne cette donnée.
- Écrire la fonction **moyenneTemperatureAMidi** qui prend une donnée de type **TemperaturesVille** et qui calcule la moyenne sur les différents capteurs installés dans la ville.
- Écrire la fonction **moyenneEcartMaxTemperature** qui prend une donnée de type **TemperaturesVille** et qui calcule la moyenne sur les écarts maximum de températures des différents capteurs installés dans la ville.
- Écrire la fonction **afficheTemperaturesVille** qui prend en entrée une donnée de type **TemperaturesVille** et affiche celle-ci sous la forme :

Nom_ville	-	nb capteurs	-	temp à midi	-	moyEcartMax	
Id_capteur_1	*	temp à midi	*	temp max	*	temp min	* ecartMax
...							
Id_capteur_n	*	temp à midi	*	temp max	*	temp min	* ecartMax

- Écrire le programme principal qui, en faisant appel aux types et fonctions précédentes, réalise la saisie des températures des différents capteurs installés dans une ville et affiche la moyenne des températures mesurées à midi, la moyenne des écarts sur l'ensemble des capteurs installés et toutes les données des capteurs.

3. Tris

Dans cet exercice nous souhaitons classer plusieurs villes selon différentes méthodes en fonction de données de leur relevé de températures.

- Écrire la fonction **saisirTableauTemperaturesVille** qui demande à l'utilisateur de saisir au clavier le nombre de villes et les données nécessaires pour construire et retourner un tableau de **TemperaturesVille** (le tableau aura une taille correspondant au nombre de villes – taille qui sera supérieure ou égale à 1).
- Écrire la fonction **afficheTableauTemperaturesVille** qui prend en entrée un tableau de **TemperaturesVille** et affiche les données de chaque élément. Les affichages de chaque élément étant séparés par une ligne de "#####".
- Écrire la fonction **triVilleParNombreDeCapteurs** qui prend en entrée un tableau de **TemperaturesVille** et qui trie celui-ci par ordre croissant du nombre de capteurs installés par ville. On réalisera ici une implémentation manuelle de la méthode de tri en utilisant un tri par recherche du minimum.
- Écrire la fonction **triVilleParTemperatureAMidi** qui prend en entrée un tableau de **TemperaturesVille** et qui trie celui-ci par ordre croissant de la moyenne des températures à midi des différents capteurs installés par ville. On utilisera ici l'interface **Comparable** et la fonction **Arrays.sort()**.
- Écrire la fonction **triVilleParMoyEcartMaxDecroissant** qui prend en entrée un tableau de **TemperaturesVille** et qui, en utilisant une classe implémentant **Comparator** et la fonction **Arrays.sort()**, réalise le tri d'un tableau de **TemperaturesVille** selon l'ordre décroissant de la moyenne des écarts max de température.
- Écrire le programme principal qui, en faisant appel aux fonctions précédentes, réalise la saisie du tableau de **TemperaturesVille**, qui affiche le tableau initial, puis qui appelle les différentes méthodes de tri et affiche après chaque tri le contenu du tableau.