

SEMAINE 4 : TP2 - HASHMAP

CLASSE HASHMAP

Consulter la Javadoc pour pouvoir utiliser correctement la classe `java.util.HashMap`

NOTES

On souhaite pouvoir accéder rapidement aux résultats des étudiants en spécifiant leur nom.

Créer une classe **Notes** dont la méthode `main()` :

1) Crée une **HashMap** avec :

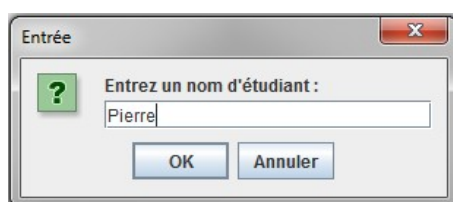
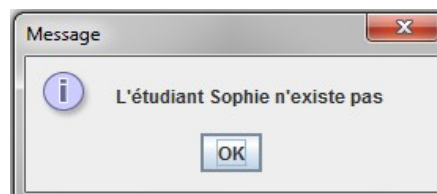
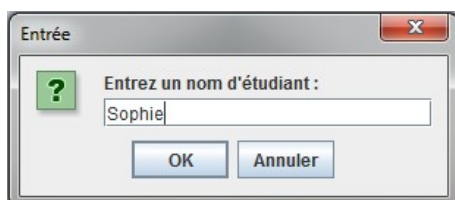
- 5 étudiants (Pierre, Paul, Jacques, Vincent , Francois),
- et leurs 5 notes (notes entre 10 et 20 attribuées arbitrairement).
 - les clés sont les noms, les valeurs sont des tableaux de 5 notes

Pour simplifier nous utiliserons l'initialisation directe de tableau (`tab={..., ..., ...}`) pour fournir les données :

1. créer un tableau de noms,
2. créer un tableau de tableaux de notes,
3. les enregistrer dans la **HashMap**

2) Propose une boîte de dialogue pour afficher les notes d'un étudiant à partir de son nom

- la syntaxe du **for** vue au TP1 est applicable aux tableaux
- un message d'erreur est affiché si le nom fourni n'est pas présent
 - (on boucle alors pour une nouvelle saisie)



3) Que se passe-t-il si on tente d'ajouter les résultats d'un deuxième étudiant s'appelant Pierre aussi ?

MODULES

On souhaite désormais conserver les résultats des étudiants mais ceux-ci ne suivent pas tous les mêmes modules.

Il faut donc conserver, en plus de la note, l'intitulé du module auquel elle correspond, de plus le nombre de modules suivis, et donc de notes dépend de chaque étudiant.

Créer une classe *Etudiant* associant un nom d'étudiant et une **HashMap** de ses notes par modules.

- une seule note par module
- on accédera à la note d'un module en spécifiant son intitulé.

Créer une nouvelle *HashMap*, dont les clés sont les noms des étudiants, permettant d'accéder simplement à leurs notes par modules.

Modifier le main en conséquence.

AGENDA - CONTACTS

On souhaite améliorer le type **Agenda** que l'on a mis au point précédemment.

- 1) On souhaite associer aux événements un ensemble de **Contacts**, et faire des recherches particulières.

Un **Contact** est constitué d'un **nom** et d'un **email**.

A un événement sont associés un ou plusieurs contacts.

- a) Ecrire la classe **Contact**,
- b) Modifier la classe **Evenement**.

- 2) Ajouter à la classe **Agenda** les fonctionnalités suivantes :

- a. Retourner tous les contacts concernés par un événement particulier,
- b. Retourner tous les contacts concernés par des événements dans une période donnée,
- c. Retourner tous les événements associés à un contact particulier.

*On notera bien qu'il s'agit de **retourner** les éléments et non de les **afficher** directement
=> retourner une ArrayList.*