

Objectif

L'application permet de gérer des notes d'un groupe d'élèves. Un groupe aura un maximum de 25 élèves. Les notes des élèves sont des entiers entre 0 et 100. Au lancement un fichier "notes.txt" fourni est chargé.

DA

1234501

1234502

1234503

1234504

1234505

1234506

1234507

1234508

1234509

1234510

1234511

1234512

1234513

1234514

1234515

1234516

1234517

Ajouter

Supprimer

Modifier

DA

1234501

Examen 1

51

Examen 2

61

TP1

71

TP2

81

OK

Annuler

Tris

Tri ascendant DA

DA	Examen1	Examen2	TP1	TP2	Total
1234501	51	61	71	81	65 %
1234502	52	62	72	82	66 %
1234503	53	63	73	83	67 %
1234504	11	22	55	44	31 %
1234505	99	84	96	98	93 %
1234506	51	61	71	81	65 %
1234507	52	62	72	82	66 %
1234508	53	63	73	83	67 %
1234509	11	22	55	44	31 %
1234510	99	84	96	98	93 %
1234511	51	61	71	81	65 %
1234512	52	62	72	82	66 %
1234513	53	63	73	83	67 %
1234514	11	22	55	44	31 %
1234515	99	84	96	98	93 %
1234516	51	61	71	81	65 %
1234517	52	62	72	82	66 %
1234518	53	63	73	83	67 %
1234519	11	22	55	44	31 %
1234520	99	84	96	98	93 %
1234521	51	61	71	81	65 %
1234522	52	62	72	82	66 %
1234523	53	63	73	83	67 %
1234524	53	63	73	83	67 %
1234525	99	84	96	98	93 %

Moyenne 54,88% 60,04% 74,12% 79,16% 65,84%

Note minimum 11% 22% 55% 44% 31%

Note maximum 99% 84% 96% 98% 93%

Nombre d'élèves 25

Quitter

- La sélection d'un DA garnit les champs de données correspondants (TextFields).
- Les boutons "Ajouter" et "Modifier" rendent accessibles les champs de données. Une fois les données traitées (bouton OK ou Annuler) les champs redeviennent inaccessibles. Les champs de données sont validés avant traitement, toute note doit être entre 0 et 100.
- Le bouton "Suppression" ainsi que l'opération DnD vers la poubelle suppriment le DA sélectionné.
- Pour tout ajout, modification et suppression le GridPane et les statistiques sont mis à jour.
- Le stage n'est pas redimensionnable.
- Demande de confirmation pour enregistrer avant de quitter.

Les tris du GridPane utilisent l'algorithme SSS par indirection.

Tris

Tri ascendant DA

Tri ascendant DA

Tri descendant DA

Tri ascendant note finale

Tri descendant note finale

Le total (la note finale) est calculé selon les pondérations suivantes:

Évaluation	Pondération
Examen 1	25%
Examen 2	30%
TP 1	20%
TP 2	25%

Méthode de la Classe Utilitaires

1. **supprimer**, supprime une ligne d'un tableau 2D d'entiers, ajuste et retourne le nombre d'éléments.
2. **ajouter**, ajoute une ligne (DA et les notes) à un tableau 2D d'entiers, ajuste et retourne le nombre d'éléments.
3. **modifier**, modifie les données (DA et les notes) d'une ligne d'un tableau 2D d'entiers.
4. **moyenneEval**, retourne la moyenne d'une colonne des N premiers éléments d'un tableau 2D d'entiers, , si N est à zéro retourner -1.
5. **minEval**, retourne la valeur minimum d'une colonne des N premiers éléments d'un tableau 2D d'entiers, si N est à zéro retourner -1.
6. **maxEval**, retourne la valeur maximum d'une colonne des N premiers éléments d'un tableau 2D d'entiers, , si N est à zéro retourner -1.

Déclarations globales

```
private static final int NB_ELEVES = 25;
private static final int NB_EVALS = 4;

private static int nbEleves = 0; //Compteur du nombre d'élèves
private static int[][] tabNotes = new int[NB_ELEVES][NB_EVALS + 2];
private static int[] index = new int[NB_ELEVES];

//Pour accéder au tableau utiliser des constantes pour les colonnes, par exemple:
private static final int DA = 0;
private static final int EXA1 = 1;

tabNotes[i][DA]    plutôt que tabNotes[i][0]
tabNotes[i][EXA1] plutôt que tabNotes[i][1]
```

Remise

Remettre dans un fichier **zip le projet NetBeans du TP** par LÉA. Tout travail remis sans commentaire méritera la note zéro. Vous devez commenter en haut du fichier java l'objectif du TP, l'auteur et la date. Toutes méthodes et variables doivent être commentées selon le standard vu en classe.

Correction

Fonctionnement du programme	80
Chargement et sauvegarde des données	10
Accès aux champs	05
Ajout	15
Modification	15
Suppression	15
Tris	10
Statistiques	10
Commentaires	10
Contenu des commentaires et qualité du français	
Aération Indentation	05
Lisibilité du code	
Identificateurs	05
Méthodes, paramètres et variables	

Ce travail représente 20% de la note finale.

France Beaudoin

Annexe

Fichier notes.txt:

```
1234501 51 61 71 81 65
1234502 52 62 72 82 66
1234503 53 63 73 83 67
1234504 11 22 55 44 31
1234505 99 84 96 98 93
1234506 51 61 71 81 65
1234507 52 62 72 82 66
1234508 53 63 73 83 67
1234509 11 22 55 44 31
1234510 99 84 96 98 93
1234511 51 61 71 81 65
1234512 52 62 72 82 66
1234513 53 63 73 83 67
1234514 11 22 55 44 31
1234515 99 84 96 98 93
1234516 51 61 71 81 65
1234517 52 62 72 82 66
1234518 53 63 73 83 67
1234519 11 22 55 44 31
1234520 99 84 96 98 93
1234521 51 61 71 81 65
1234522 52 62 72 82 66
1234523 53 63 73 83 67
1234524 53 63 73 83 67
1234525 99 84 96 98 93
```

Exemple de découpage du code:

```
//-----VALIDATION-----
private boolean validerDA(){...17 lines }
private boolean validerExamen1(){...17 lines }
private boolean validerExamen2(){...17 lines }
private boolean validerTP1(){...17 lines }
private boolean validerTP2(){...17 lines }

//-----ACCES / SCENE / STATS -----
private void visibiliteErreurs(boolean vis){...7 lines }
private void disableDetailsDA(boolean acces){...10 lines }
private void viderDetailsDA(){...7 lines }

//Préparer la scène au lancement de l'application
public void preparerScene(){...24 lines }
public void calculerStats(){...31 lines }

//-----FICHIERS-----
public static void ecrireFichierText(String nomFichier) throws IOException {...17 lines }

//Lire fichier notes.txt et remplir tabNotes
public static void lireFichier(String nomFichier) throws FileNotFoundException, IOException {...24 lines }

//-----GRIDPANE-----
public void fixerGridLignes(){...8 lines }
public void afficherGridEntete(){...9 lines }
public void afficherGridContenuTrie(){...20 lines }
public void viderGridContenu(){...6 lines }

//-----TRIS INDEX SSS-----
public void permuter(int[] index, int i, int j){...5 lines }
public void triAscDA(int[] index, int[][] tab, int nb){...15 lines }
public void triDesDA(int[] index, int[][] tab, int nb){...15 lines }
public void triAscTotal(int[] index, int[][] tab, int nb){...15 lines }
```