

DM2 (obligatoire) – Un outil d'apprentissage des trombinoscopes

Méthode de travail pour ce DM :

Ce DM sert à vous préparer à l'examen final, et à savoir programmer (en autonomie) un outil. Je vous conseille donc de le préparer de manière individuelle. Toutefois, si vous choisissez de travailler dessus à plusieurs, je l'autorise, **sous réserve que vous respectiez les conditions suivantes** :

- indiquez-moi avec qui vous avez travaillé pour préparer ce DM.
- rédigez le devoir de manière individuelle. Vous pouvez réfléchir aux questions ensemble et en discuter ensemble, mais la phase d'écriture doit être personnelle.

Tout manquement manifeste à ces conditions sera sanctionné par des points de pénalité.

Lisez le sujet du DM jusqu'à la fin avant de commencer.

Objectifs :

Le but de ce DM est d'obtenir un outil qui va permettre d'apprendre le trombinoscope de la promo, en suivant les « questions intermédiaires » ci-dessous. L'idée est que l'utilisateur :

- ouvre un fichier image contenant un trombinoscope où les visages sont numérotés (le fichier

<http://www.lirmm.fr/~gambette/ENSIUT/DM2.png>)

- exécute dans un terminal le programme que vous allez réaliser. Le programme lui demande sur combien d'étudiants il veut se tester, puis fait les tests de la manière suivante : il tire un numéro au hasard entre 1 et 58, demande à l'utilisateur à quel(le) étudiant(e) le numéro correspond, vérifie si la réponse est vraie ou fausse, et calcule le pourcentage d'erreurs de l'utilisateur.

Exemple :

```
C:\Windows\system32\cmd.exe
C:\IUT\2011-DM2>javac DM2.java
C:\IUT\2011-DM2>java DM2
Combien de questions ?
4
Question 1/4 : Quel est le nom de l'etudiant numero 12 ?
Justine Collot
Bonne reponse !
Vous avez actuellement 100% de bonnes reponses
Question 2/4 : Quel est le nom de l'etudiant numero 37 ?
Marie Helluy
Bonne reponse !
Vous avez actuellement 100% de bonnes reponses
Question 3/4 : Quel est le nom de l'etudiant numero 33 ?
Lucie Lam
Erreur, il s'agissait de Julie Lam.
Vous avez actuellement 66% de bonnes reponses
Question 4/4 : Quel est le nom de l'etudiant numero 28 ?
Anthony Lé
Erreur, il s'agissait de Anthony Le.
Vous avez actuellement 50% de bonnes reponses
C:\IUT\2011-DM2>
```



Remarquez que pour obtenir un affichage correct et pour plus de simplicité, tous les accents ont été enlevés.

Outils à disposition :

Vous considérez que vous avez à disposition les algorithmes suivants :

- l'algorithme **Affiche** prend en entrée une chaîne de caractères et l'affiche dans la ligne de commande.
- l'algorithme **reponseALaQuestion** prend en entrée une chaîne de caractères, l'affiche dans la ligne de commande en demandant à l'utilisateur d'entrer du texte, et renvoie une chaîne de caractères contenant le texte entré par l'utilisateur.
- l'algorithme **ChainesEgales** prend en entrée deux chaînes de caractères et renvoie VRAI si elles sont égales, FAUX sinon (en Java, on ne peut pas écrire directement « `if (chaine1==chaine2) ...` » pour vérifier que deux chaînes sont égales, il faut écrire « `if (ChainesEgales (chaine1, chaine2) ...` »).
- l'algorithme **nbAleatoire** prend en entrée deux entiers *i* et *j* et renvoie un entier aléatoire entre *i* et *j* inclus.
- l'algorithme **convertitChaineEnEntier** prend en entrée une chaîne de caractères *chaine* qui contient un entier (par exemple "42") et renvoie cet entier (dans l'exemple : 42). Il « convertit » donc une chaîne de caractères en entier.

Le code actuel du fichier <http://www.lirmm.fr/~gambette/ENSIUT/DM2.java> vous montre un exemple d'utilisation de ces différents algorithmes.

Consigne :

Pour chaque question intermédiaire, vous pouvez choisir de répondre soit toujours en pseudo-code, soit toujours en Java, dans un fichier texte ou Word ou OpenOffice (disons qu'il s'appelle *Reponses.odt*).

Pour la question 8, ajoutez votre réponse (l'algorithme complet en pseudo-code) à la fin du fichier *Reponses.odt*.

Pour la question 9, remplacez l'intérieur de l'algorithme **main** (qui commence à la ligne « `public static void main (String[] arg)` ») par votre algorithme en Java, dans le fichier

<http://www.lirmm.fr/~gambette/ENSIUT/DM2.java>. Ajoutez le code de l'algorithme **NomEtudiant** en Java juste après le code de l'algorithme **main** dans le fichier *DM2.java*.

Envoyez-moi par mail à philippe.gambette@gmail.com les deux fichiers *DM2.java* et *Reponses.odt*.

Questions intermédiaires (barème entre parenthèses) :

- 1- Écrivez les instructions qui permettent de demander à l'utilisateur sur combien de questions il veut se tester, récupérer la réponse, et stockent le résultat sous forme d'un entier (2 points).
- 2- Écrivez l'instruction qui tire un nombre aléatoire pour décider quel sera le numéro demandé à l'utilisateur (2 points).
- 3- Supposez que vous avez à disposition un entier n et le nom de l'étudiant correspondant dans une chaîne de caractères *etudiant* (par exemple $n=42$ et *etudiant*="Cyril Provost"). Écrivez les instructions qui permettent de poser la question à l'utilisateur du programme en utilisant les variables n et *etudiant*, récupérer sa réponse à la question, et voir si elle correspond à la chaîne de caractères stockée dans *etudiant* (2 points).
- 4- Écrivez un algorithme **NomEtudiant** qui prend en entrée un entier n qui est un numéro d'étudiant (entre 1 et 58) et renvoie une chaîne de caractères *chaine* constituée du prénom puis du nom de l'étudiant. Pour cela, vous pouvez :
 - * soit utiliser un tableau (qui va stocker les prénoms et noms des étudiants, dont vous récupérerez le contenu de la n -ième case)
 - * soit utiliser des tests en fonction de la valeur de n .Vous utiliserez le fichier <http://www.lirmm.fr/~gambette/ENSIUT/DM2.txt> pour récupérer la correspondance entre les numéros et les prénoms/noms. Il est possible que le code de cette question soit assez long et qu'il contienne plus de 58 lignes (3 points).
- 5- En utilisant la valeur de l'entier que vous avez défini à la question 1 (qui compte le nombre de questions), ainsi que les questions 2, 3 et 4, écrivez une boucle qui : a) tire un numéro au hasard ; b) pose la question à l'utilisateur ; c) affiche la réponse donnée par l'utilisateur ; d) affiche la réponse qu'il aurait dû donner (3 points).
- 6- Ajoutez le code qui permet de tester si la réponse de l'utilisateur est correcte, et selon le cas, fait les mises à jour nécessaires pour afficher le taux de bonnes réponses (1 point).
- 7- Quelle fonctionnalité ajouteriez-vous à ce programme ? Choisissez une fonctionnalité que vous sauriez programmer (au moins en pseudo-code), mais **ON NE VOUS DEMANDE PAS DE LA PROGRAMMER EN JAVA NI DE DONNER SON ALGORITHME EN PSEUDO-CODE** (1 point).

Questions finales :

- 8- Écrivez en pseudo-code l'ensemble de l'algorithme **TestTrombinoscope** en combinant les réponses aux questions intermédiaires. Cet algorithme doit répondre aux « objectifs » définis au début de ce DM. Il ne prend rien en entrée, ne renvoie rien en sortie, mais interagit avec l'utilisateur par la ligne de commande. (3 points)
- 9- Insérez dans le fichier *DM2.java* le code Java nécessaire pour obtenir un programme fonctionnel qui répond aux objectifs. (3 points)