

Projets_6

Tous les projets seront réalisés en Java, il seront notés avec des notes de 0 à 6

1. Réaliser un programme d'interrogation sur le langage JAVA. L'utilisateur soumet au programme le nom d'une classe et le programme répond en affichant
 1. si la classe existe, la description de la classe et de son utilisation et propose un exemple qui fonctionne.
 2. sinon, un message indiquant que la classe est inconnue

Le programme reconnaîtra 10 classes, au choix.

2. Réaliser un programme Java pour l'affichage graphique des états par lesquels peuvent passer les processus. Lorsque l'utilisateur pose la souris sur un des états affichés, le programme affichera une « bulle » avec une explication décrivant l'état.
3. Réaliser un programme Java pour l'analyse textuelle. Le programme lira un fichier texte (*.txt) et établira :
 1. une liste ordonnée alphabétiquement des mots qui sont utilisés dans le texte, ainsi que leur fréquence
 2. une liste croissante du nombre de lettres dans les mots et leur fréquence (nombre de mots composés d'une lettre, nombre de mots composés de deux lettres, etc....). Un graphique et la moyenne seront également affichés
4. Réaliser un programme Java qui construit une table avec les noms des étudiants du groupe. La table sera sauvegardée sur le disque après chaque modification. Une entrée dans cette table comprendra le nom et le prénom de l'étudiant, le titre du dernier spectacle auquel il a assisté, le type de spectacle (film, théâtre, concert, opéra, cirque...), un code d'autorisation (user ou administrateur) et un mot de passe lui permettant de changer le contenu de la table.

L'auteur du programme sera aussi administrateur. Il sera le seul à pouvoir éliminer une entrée de la table.

5. Écrire un programme Java pour la consultation du tableau de Mendeleev. Les paramètres utilisés pour interroger seront la désignation, le symbole, le poids atomique et la position dans le tableau (exprimée par la ligne *et* la colonne) de l'élément. L'utilisateur fournira une ou deux des informations ci-dessus et le programme lui fournira les informations manquantes. En cas de demande erronée le programme répondra « élément inconnu ».

Vous utiliserez le tableau de Mendeleev proposé par WIKIPEDIA :

[Tableau de Mendeleev](#)

6. On obtient un anagramme en changeant l'ordre des lettres dans un mot ou une proposition (exemples : bilan / Liban, conversation / conservation) Écrire un programme Java qui vérifie que deux séquences de caractères sont l'une l'anagramme de l'autre, sans tenir compte de la casse (majuscules/minuscules).
7. Écrire un programme qui calcule le jour de la semaine d'une date quelconque. La date sera fournie par l'utilisateur sous la forme jj/mm/AAAA (format européen). Ne pas oublier de tenir compte des années bissextiles.

8. Écrire un programme Java qui trouve le chemin le plus économique parmi les chemins possibles entre deux destinations. Tous les parcours comportent 4 étapes et 4 seulement, dont les coûts sont inscrits dans un tableau avec 5 lignes et 5 colonnes comme celui ci-dessous donné en exemple. L'utilisateur du programme doit pouvoir saisir les coûts de chaque étape (25 valeurs) et ensuite lancer la recherche.

12	100	24	85	32
18	55	42	55	11
48	56	12	12	47
11	54	32	16	65
7	14	96	47	45

Les chemins doivent être composés de cases qui se « touchent » (sur la même ligne ou sur la diagonale). Ils seront affichés dans le tableau, en colorant avec la même couleur les cases par lesquelles il faut passer. Si plusieurs chemins coûtent le même prix, il faut les afficher tous en utilisant plusieurs tableaux.

9. Écrire un programme Java qui permet d'illustrer la [conjecture de Syracuse](#). L'utilisateur saisira un nombre entier et le programme calculera et affichera la série générée par cette conjecture. Le programme comprendra tous les tests nécessaires pour éviter une saisie incorrecte.
10. Écrire un programme qui affiche les cartes d'un paquet de cartes à jouer. Les règles de l'affichage seront saisies par l'utilisateur, parmi les suivantes :
- Toutes les cartes dans ordre croissant des valeurs (1,2,... V,D,R) et des couleurs (Trèfle, Carreau, Cœur, Pique)
 - Toutes les cartes dans l'ordre inverse des valeurs et des couleurs
 - Toutes les cartes dans un ordre aléatoire (comme après avoir battu les cartes)
 - Seulement 13 cartes tirées au hasard, mais le nombre de cartes de chaque couleur spécifié à la saisie.
11. Écrire un programme qui exécute l'algorithme des banquiers pour la prévention des blocages. Exécuter ce programme sur des données saisies par l'utilisateur dans des tableaux comportant au minimum 6 processus et 5 ressources. Le programme devra mettre en évidence les blocages potentiels en cas de décisions inappropriées d'affectation des ressources.
12. Écrire un programme qui détecte un cycle dans un graphe d'allocation des ressources et prévient l'utilisateur du danger de blocage. La structure du graphe d'allocation sera saisie par l'utilisateur sous forme de tables :
1. Table des utilisations en cours, avec une ligne chaque processus et une colonne pour chaque ressource.
 2. Table des ressources sollicitées, avec une ligne pour chaque processus et une colonne pour chaque ressource
 3. Vecteur des ressources disponibles avec une entrée pour chaque ressource.

Le programme devra afficher l'état des tables après chaque opération et signaler l'existence d'un cycle bloquant

