

TRAVAUX PRATIQUES

JAVA – BASE DU LANGAGES

Afin de réaliser ces exercices de mise en œuvre, les notions suivantes seront utilisées : la notion de variables, les structures de contrôle, les structures itératives, les tableaux, les procédures, les paramètres et retour de méthodes.

Également, il vous est demandé de réaliser ces exercices en respectant les notions de conventions de nommage de JAVA et la structure d'organisation du code.

L'interface utilisateur se fera à travers la console pour les demandes de saisies, l'affichage des résultats et messages.

Le travail est à réaliser en autonomie mais l'entraide n'est pas interdite. Le travail est à réaliser au plus tard lundi 5 août 2024 - 17h sous forme d'un dossier archive contenant vos sources.

EXERCICE 1

- Ecrivez une application permettant la résolution d'une équation du second degré
 - $ax^2 + bx + c = 0$.
- On saisira les **entiers** a , b et c .
- La(les) solution(s) seront de type réel.
- Vous afficherez le résultat pour produire un message de ce type reprenant l'équation et l'affichage de la/les solutions de l'équation.

EXERCICE 2

- Ecrivez une application permettant le calcul d'un nombre X élevé à la puissance Y .
- Les **entiers positifs** X et Y seront saisis.
- Ecrivez une fonction ***long puissance(int x, int y)*** qui admet deux paramètres x et y de type entier et qui renvoie le résultat de x élevé à la puissance y .

EXERCICE 2

- Ecrivez une application permettant la recherche **dichotomique** d'une variable entière X dans un tableau d'entiers *TabEntiers* de taille TAILLE_MAX **déjà trié**.
- Utilisez cette fois les structures itératives et de contrôle que vous jugerez nécessaires.
- Déclarez un tableau de 10 entiers. Rangez-y 10 valeurs négatives, positives ou nulles dans **l'ordre croissant**.
- Afficher ce tableau avant et après rangement.
- Ecrivez une fonction ***int rechercherEntier (int tab[] , int x)*** qui recherche la **position** de la variable x dans le tableau *tab* en mettant en œuvre la recherche dichotomique.
- Si x est trouvée dans *tab*, la fonction renvoie le rang du tableau auquel elle se trouve.
- Si x n'existe pas dans le tableau, la fonction renvoie -1 indiquant ainsi l'absence de cette valeur au programme appelant.

EXERCICE 3

- Ecrivez un algorithme en pseudocode qui permet de remplir aléatoirement un tableau d'entiers de TAILLE éléments dans le module principal puis qui affiche la valeur entière maximum de ce tableau d'entiers.
- La recherche de l'entier maximum présent dans le tableau sera confiée à une fonction ***lireMaxTabEntiers*** qui recevra en argument le tableau et qui renverra l'entier maxi lu dans ce tableau.
- Afficher votre tableau puis l'élément maximum à l'écran
- Codez votre solution.

EXERCICE 4

- Ecrivez un algorithme en pseudocode permettant le rangement, dans un tableau *TabMultiplication*, des tables de multiplication de 1*1 jusqu'à 9*9.
- Le nombre des colonnes (10) et le nombre des lignes (10) seront consignés dans deux constantes LIGNES et COLONNES.
- Le tableau, une fois rempli dans le module principal, devra permettre de générer, à la suite d'un dialogue, une réponse de la forme : « **Le produit de : »**, X, « **par : »**, Y « **est : »**, **PRODUIT**
- Les variables entières X et Y ayant fait, au préalable, l'objet d'une saisie interactive
- Toute saisie incorrecte produira l'affichage suivant : **Veillez entrer un entier !**
- Le tableau devra être rempli selon le format suivant :

	1	2	...	8	9
1	1	2	...	8	9
2	2	4	...	16	18
...
8	8	16	...	64	72
9	9	18	...	72	81

- Vous utiliserez les structures de contrôle **itératives** les mieux adaptées.
- Naturellement, vous exploiterez le tableau pour extraire la valeur située à l'intersection des ligne/colonne correspondant aux valeurs saisies.
- La réponse sera de la forme : **Le résultat de 8 x 6 = 48**
- Après un calcul et la production de son affichage, vous proposerez à l'utilisateur le dialogue suivant : **Souhaitez-vous encore un calcul ?**
- Codez votre solution.

EXERCICE 5

- Ecrivez un algorithme en pseudocode permettant de **trier un tableau d'entiers**.
- Au sein du module principal, ce tableau sera déclaré et rempli avec des entiers négatifs, positifs ou nuls, sans ordre particulier dans le module principal et transmis dans une procédure *trierTableau*.
- *trierTableau* va donc recevoir en unique argument le tableau d'entiers à trier. On utilisera la **méthode de tri par insertion**.
- Ecrivez le code de la fonction *trierTableau(...)* dans la classe *Utilitaires*, similairement aux solutions proposées précédentes.
- Ecrivez également une procédure permettant d'afficher dans une boîte de dialogue le tableau comme dans les exemples précédents.
- Vous utiliserez cette procédure pour afficher le tableau d'entiers non trié, puis ce même tableau trié après l'appel à *trierTableau()*.
- Lorsque vous coderez votre pseudocode, écrivez un module pour remplir aléatoirement le tableau d'entiers.
- *Bonus* : Ajoutez un booléen à la méthode de tri pour trier selon un ordre croissant ou décroissant en fonction de la valeur de ce booléen.

EXERCICE 6

- Ecrivez un algorithme en pseudocode permettant de réaliser une calculatrice pour les 4 opérations arithmétiques suivantes : + , - , * , / .
- Dans le module principal, on proposera à l'utilisateur un menu l'invitant à saisir :
 - Le caractère associé à l'opération choisie. (une des quatre opérations arithmétiques)
 - Les 2 valeurs entières pour lesquelles il souhaite réaliser l'opération
- Ecrivez une procédure '*calculer*' qui admet les trois paramètres suivants :
 - **Reel** calculer(**Char** operateur, **Reel** operande1, **Reel** operande2)
- On voit que la fonction *calculer(...)* reçoit les paramètres nécessaires pour effectuer une opération arithmétique :
 - Le premier est le caractère correspondant à l'opération : ce sera '+' ou '-' ou '*' ou encore '/'.
 - Le deuxième sera le premier opérande pour le calcul.
 - Le troisième sera le second opérande pour le calcul.
 - La fonction *calculer()*, après avoir effectué le calcul, retournera celui-ci au module appelant. Ce résultat sera de type réel.
- Vous mettrez en œuvre toutes les dialogues nécessaires, ainsi que les contrôles après saisies, requis pour une utilisation standard de la calculatrice.
- Codez votre solution.