

Exercices sur les méthodes

Maria-Virginia Aponte, François Barthélémy

NFA031

Exercice 1 adaptation de programme

Le programme suivant calcule la division entière d'un nombre par un autre en utilisant la soustraction pour calculer le résultat sans utiliser l'opérateur `/`.

```
package nfa031;
public class Meth_1{
    public static void main(String[] args){
        int x,y,cour,res;
        Terminal.ecrireString("Entrez le nombre x: ");
        x = Terminal.lireInt();
        Terminal.ecrireString("Entrez le nombre y: ");
        y = Terminal.lireInt();
        res = 0;
        cour = x;
        while(cour>=y){
            cour=cour-y;
            res = res + 1;
        }
        Terminal.ecrireString("'" + x + "/" + y + "= ");
        Terminal.ecrireIntln(res);
    }
}
```

Modifiez le programme pour que le calcul de la division soit réalisé par une fonction à deux paramètres, `x` et `y` (en gardant le même algorithme de calcul avec une boucle).

Exercice 2 fonctions mathématiques

Ecrire un programme avec les fonctions `carre` et `cube` qui calculent respectivement le carré et le cube (ou puissance 3) d'un nombre de type `double`. La méthode `main` doit tester ces deux fonctions sur plusieurs exemples.

Exercice 3 affichage d'un rectangle

Question 1

Écrivez une méthode qui prend en paramètre une largeur et une hauteur, qui affiche un rectangle composé d'étoiles ayant cette largeur et cette hauteur.

Par exemple, si la largeur est 6 et la hauteur est 4, l'affichage sera le suivant :

```
*****
*****
*****
*****
```

Vous écrierez également une méthode main permettant de tester cette méthode d'affichage.

Question 2

Même question, mais cette fois, le rectangle ne sera pas forcément composé d'étoiles, mais d'un certain caractère passé en paramètre à la méthode.

Exercice 4 système de validation pour une école

Contexte

Une école primaire souhaite automatiser la validation des inscriptions d'élèves. Vous devez créer un système de méthodes de validation pour vérifier que les données saisies lors des inscriptions sont cohérentes et respectent les règles de l'établissement.

Partie 1 : Méthodes de base

Créez les méthodes de validation suivantes dans une classe `ValidationEcole` :

Listing 1 – Signatures des méthodes à implémenter

```
// Valide qu'une année de naissance est plausible pour un élève
public static boolean estAnneeNaissanceValide(int anneeNaissance,
int anneeActuelle)

// Vérifie si un élève a l'âge requis pour une classe donnée
public static boolean estAgeApproprieeClasse(int age, String classe)

// Valide qu'un nom n'est pas vide et contient au moins 2 caractères
public static boolean estNomValide(String nom)
```

Contraintes à respecter :

- **Années de naissance valides** : entre 1990 et l'année actuelle incluses
- **Âges par classe** :
 - "CP" : 6-7 ans
 - "CE1" : 7-8 ans
 - "CE2" : 8-9 ans
 - "CM1" : 9-10 ans
 - "CM2" : 10-11 ans
- **Noms valides** : minimum 2 caractères, pas de chaîne vide ou nulle

Partie 2 : Validation combinée

Créez une méthode qui utilise les méthodes précédentes pour valider toutes les données d'un élève en une seule fois :

Listing 2 – Méthode de validation globale

```
// Valide toutes les données d'un élève
public static boolean estEleveValide(String nom, int anneeNaissance,
String classe, int anneeActuelle)
```

Cette méthode doit retourner **true** seulement si toutes les validations individuelles réussissent.

Partie 3 : Programme de test

Écrivez un programme principal qui :

1. Demande les informations d'un élève (nom, année de naissance, classe)
2. Utilise vos méthodes de validation
3. Affiche des messages d'erreur spécifiques pour chaque problème détecté
4. Confirme l'inscription si toutes les données sont valides
5. Calcule et affiche l'âge de l'élève

Exercice 5 fonctions sur les tableaux

Question 1

Ecrire une fonction qui cherche si un élément appartient à un tableau de **char**. Le caractère recherché et le tableau seront les deux paramètres de la fonction. Le résultat sera un booléen.

Question 2

Ecrire une fonction qui compte le nombre d'occurrences d'un caractère dans un tableau, c'est à dire le nombre de fois où un élément apparaît dans un tableau de caractères. Le caractère recherché et le tableau seront les deux paramètres de la fonction.

	caractère	tableau	résultat
Exemples :	'z'	{ 'a', 'b', 'c' }	0
	'a'	{ 'a', 'b', 'c' }	1
	'b'	{ 'a', 'b', 'c', 'b', 'd', 'b', 'a', 'd' }	3

Question 3

Ecrire une fonction qui prend deux tableaux de caractères en paramètres et qui teste si tous les éléments du premier tableau apparaissent au moins une fois dans le deuxième tableau. Il est possible d'utiliser dans le corps de cette fonction la fonction écrite pour la réponse à la question 1.