

TP 1 : Java

Exercice 1

1. Ecrire une classe **Personne** permettant de décrire une personne, sachant que l'on souhaite avoir autant d'informations que dans la phrase suivante :
"M. Bruno SALVES est né le 12 Janvier 1990 et il est célibataire."
2. Ajouter un constructeur à la classe **Personne** permettant de renseigner toutes les informations relatives à une personne.
3. Ajouter à la classe **Personne**, une méthode de nom **retourneInfos**. Cette méthode doit retourner une chaîne de caractères similaire à la phrase donnée dans l'énoncé de la question 1.
4. Ecrire un programme qui déclare 3 variables de type **Personne**, crée 3 instances de **Personne** pour les affecter dans les variables et affiche les informations les concernant.
5. Ajouter une méthode **age** qui renvoie l'âge de la personne en fonction d'une année donnée en paramètre.
6. Ajouter à la classe **Personne** un attribut **conjoint** et examiner les conséquences que cela peut avoir sur l'ensemble du code.
7. Ajouter une méthode **marier(Personne p)** qui permet de marier une personne à une autre. Modifier la méthode **retourneInfos** de façon que le nom (traditionnel de la femme devienne):
"[nom de l'époux] née [nom de jeune fille]", par exemple :
si Mlle BEN SALEH se marie avec M. BEN MOHMAED, son nom deviendra Mme "BEN MOHAMED née BEN SALEH"
8. Modifier votre programme pour faire marier des personnes et afficher par la suite leurs informations

Exercice 2

1. Définissez une classe **Individu** composée d'un **nom**, d'une **adresse**, et d'un numéro de **telephone**. Cette classe doit contenir les méthodes suivantes :
 - un constructeur permettant d'initialiser toutes les informations relatives à un **Individu**,
 - les accesseurs et les mutateurs des différents champs.
 - affichage permettant d'afficher toutes les informations relatives à un **Individu**
2. Modifiez la classe **Individu** pour pouvoir :
 - afficher le nombre d'objets créés,
 - gérer la liste des individus créés dans un tableau d'individus,
 - afficher la liste des individus créés,

3. Ecrivez un programme permettant de créer des individus et d'afficher par la suite le nombre d'instances créées et de d'afficher la liste des individus créés

Exercice 3

1. Donnez le code de la classe **Mesure** qui modélise des mesures réelles (températures par exemple) :

- une mesure a une **Valeur** de type réel.
- le constructeur de la classe **Mesure** permet d'instancier un objet de cette classe avec la valeur réelle fournie en paramètre. Si la valeur fournie est en-dehors de l'intervalle [-100,100] (Min et Max), le message "mesure hors intervalle" est affiché et la valeur n'est pas prise en compte.
- **NbMesures**, le nombre d'instances de la classe **Mesure** est géré par la classe ainsi que **SommeMesures** la somme des valeurs des mesures instanciées.
- la méthode **afficherMoyenne** calcule et affiche la moyenne de toutes les mesures instanciées, le nombre de mesures est également affiché. S'il n'y a pas de mesures, le message "zéro mesures" est affiché.

2. Donnez le code source d'un programme Java nommé **EntreeMesures** permettant à l'utilisateur d'entrer un nombre arbitraire de valeurs au clavier. Ces valeurs sont utilisées pour instancier des **Mesures**.

Quand la valeur 0 est entrée, le programme fait appel à **afficherMoyenne** avant de terminer. Voici un exemple d'affichage lors de l'exécution du programme :

```
Entrez une mesure :120
Mesure hors intervalle
Entrez une mesure :10
Entrez une mesure :20
Entrez une mesure :30
Entrez une mesure :0
La moyenne des 3 mesures est :20
```

Exercice 4

1. Créez la classe **Livre** ayant pour attribut obligatoire le nom de l'auteur, le titre et comme attribut optionnel le nombre de pages et l'éditeur. Ces valeurs seront initialisées via l'usage d'un ou de plusieurs constructeurs. Les attributs seront privés, et on utilisera des méthodes d'accès publiques. On veut également une méthode **toString()** renvoyant une chaîne de caractères contenant les informations sur le livre.

2. Modifiez la classe pour pouvoir gérer le nombre de livres instanciés. Pour cela on utilisera la variable **private static int numLivre** pour stocker cette information. Modifiez le ou les constructeurs de manière à ce que cela fonctionne.

3. Ajoutez le champs **numero** à chaque livre afin de stocker l'ordre de sa création. Modifier en conséquence la méthode **toString()** et le ou les constructeurs pour gérer ce numéro.

4. Ecrivez un programme permettant de créer des Livres et de d'afficher leurs informations.