

## Série de TP n° 2 – java

### Classes et Objets

- Objectifs : implémentation de classes en java, instanciation d'objets et manipulation de tableaux d'objet, Manipulation de chaînes de caractères.
- Dans cette série, chaque programme doit comporter au moins deux classes dont une qui contient la méthode main.
- **Les messages de saisie et d'affichage doivent être explicites à l'utilisateur du Programme.**

#### Exercice 1

On définit une classe Etudiant caractérisée par les attributs et méthodes suivants :

##### Attributs

matricule : Nombre entier long représentant le matricule d'un étudiant.

nom, prenom et spec : Chaînes de caractères (String) représentant respectivement, le nom, prénom et la spécialité d'un étudiant

Sec : caractère, représente la section d'un étudiant

Note1, Note2, Note3 : Notes de l'étudiant dans trois modules de même coefficient

##### Méthodes

void afficher () : affiche les informations d'un étudiant,

float moyenne () : calcule la moyenne d'un étudiant,

void setSec (char) : modifie la section d'un étudiant,

String getNom () : renvoie le nom de l'étudiant

1. Implémenter la classe Etudiant en Java.
2. Définir une autre classe ProgEtudiant dans laquelle on ajoute la méthode main :
3. Déclarer un objet Etudiant, nommé E1 et l'instancier
  - Quelle sont les valeurs de ses attributs ?
  - Quelles sont les modifications à apporter pour avoir un objet avec des valeurs précises ?
4. Créer deux objets Etudiant E1 et E2, tel que E1 = (512555, ~~Ⓜ~~Azi~~Ⓜ~~, ~~Ⓜ~~Amine~~Ⓜ~~, ~~Ⓜ~~ISIL~~Ⓜ~~, 'A', 13, 12) et E2 = (512004, ~~Ⓜ~~Ziane~~Ⓜ~~, ~~Ⓜ~~Maroua~~Ⓜ~~, ~~Ⓜ~~ISIL~~Ⓜ~~, 'B', 14, 15).
5. Modifier la section de l'étudiant ~~Ⓜ~~Azi~~Ⓜ~~ de A à B.
6. Calculer les moyennes des deux étudiants et afficher leurs descriptions complètes.
7. Affecter E1 à E2. Si on invoque la méthode afficher avec E2, quel résultat va-t-on avoir ? Justifier votre réponse en donnant le contenu des zones mémoires et des références et objets créés.

#### Exercice 2

On considère une entité **HEURE** représentant l'heure décrite par la donnée de trois champs heure, minute et seconde.

1. Donner une implémentation objet de l'entité **HEURE** sus-décrite. Prévoir une méthode VERIF qui vérifie si les données sont correctes ( $0 \leq \text{heure} \leq 23$  ;  $0 \leq \text{minute} \leq 60$  ;  $0 \leq \text{seconde} \leq 60$ ), une méthode d'affichage et les autres méthodes nécessaires au traitement de la question 2.
2. Etant donnée une heure (H, M, S) entrée au clavier, écrire un programme java qui calcule et affiche l'heure (H :M :S) après un nombre S de secondes.

### Exemple

Si l'heure est égale à 10 : 58 : 45 après S = 125 secondes, il sera 11 : 00 : 50

### Exercice 3

Dans cet exercice, on désire créer et manipuler un carnet d'adresses (postales).

1. Donnez l'implémentation d'une entité **Adresse** (numéro, nomRue, codePostal, ville). Prévoir une méthode qui affiche l'entité et une méthode qui la modifie.
2. Donnez l'implémentation d'une entité **Personne** (nom, prénom, adresse). Prévoir une méthode qui affiche l'entité Personne et une méthode qui modifie l'adresse de la personne.
3. On demande d'écrire un programme qui :
  - Crée un tableau P de n éléments de type Personne et remplit le tableau avec des informations entrées au clavier.
  - Recherche et affiche toutes les personnes dont le code postal est égal à CP donné.
  - Modifie l'adresse d'une personne de nom donné et affiche le contenu de tout le tableau.

### Exemple de données de Test

Nom	Prénom	numRue	nomRue	codePostal	ville
Amine	Amine	20	R1	10000	Alger
Khaled	Khaled	15	R2	11111	Blida
Walid	Walid	100	R1	10000	Alger
Yasmine	Yasmine	13	R3	12000	Blida
Maria	Maria	200	R4	10000	Alger

Pour la recherche : CP = 10000

Pour la modification : Nom = walid nouvelle adresse = (26, R3, 12000, Blida)

### Exercice 4

Dans cet exercice, on désire programmer les congés des employés d'une société.

1. Donnez l'implémentation d'une entité **Date** (jour, mois, année). Prévoir une méthode qui affiche l'entité et une méthode qui compare deux dates.
2. Donnez l'implémentation d'une entité **CongéEmp** (nom, prénom, datedeb, datefin) qui décrit la période de congé d'un employé. Prévoir une méthode qui affiche l'entité et des méthodes qui modifient les dates de congé.

3. On demande d'écrire un programme qui :

- Crée un tableau **C** de **n** éléments de type **CongéEmp** et remplit le tableau avec des informations entrées au clavier.
- Recherche et affiche tous les employés qui sont en congé à une date d donnée.
- Modifie la période de congé d'un employé (nouvelle datedeb et nouvelle datefin); et affiche ensuite le contenu de tout le tableau.

### Exemple de données de Test

Nom	Prénom	datedeb	Datefin
Amine	Amine	5/5/2009	20/5/2009
Khaled	Khaled	1/6/2009	15/1/2009
Walid	Walid	25/5/2009	10/6/2009
Yasmine	Yasmine	15/5/2009	31/5/2009
Maria	Maria	6/6/2009	16/6/2009

Pour la recherche : date = 27/05/2009

Pour la modification : nom = « Maria » datedeb = 10/06/09 datefin = 25/6/2009

**Indication :** Méthode S1. **compareTo(S2)** définie dans java.lang.String : compare deux chaînes de caractères.

### Exercice 5

On voudrait programmer un jeu simple qui consiste à générer des valeurs aléatoires par le système et attribuer des points aux joueurs selon la valeur générée.

Un joueur est décrit par un *numéro*, un *nom* et un *score* (initialement égal à 0 et incrémenté à chaque fois selon la règle du jeu).

### Description du jeu

Un joueur introduit au clavier une valeur **val** entière quelconque, on considère alors l'intervalle **I** = [**val-1000**, **val+1000**] qui encadre la valeur val. Le système génère ensuite, **dix** (10) valeurs **aléatoires**. A chaque fois que la valeur générée appartient à l'intervalle **I**, le joueur gagne un point de plus dans son score.

1. Donnez une implémentation de la classe **Joueur**. Prévoir une méthode de saisie, une méthode d'affichage, et une méthode PartieJeu qui implémente une partie de jeu pour un joueur.
2. Ecrire un programme qui crée deux joueurs **J1** et **J2**, implémente une partie de jeu pour chacun et affiche ensuite le joueur gagnant (celui qui a le meilleur score).

On considère maintenant, une partie de jeu avec 20 joueurs créés un par un et stockés dans un vecteur **V**. Le jeu se déroulera de la même manière que précédemment pour chaque joueur un par un, le score de chacun d'eux doit être enregistré.

3. Ecrire le programme qui implémente la partie du jeu avec les 20 joueurs, enregistre les joueurs avec les scores dans le vecteur, les *trie* et les affiche ensuite dans *l'ordre décroissant* de leurs scores respectifs.
4. Que faut-il modifier si on déclare l'attribut « score » avec le niveau de visibilité « private » ?