



Contrôle (Module : Java)
(2heures)

Exercice 1 : (16 points) :

Un établissement public veut organiser un concours pour recruter un technicien spécialisé (Bac+2) en informatique, ce concours comportera deux tests : écrit et oral. L'avis du concours est affiché sur le site officiel de l'établissement.

La gestion des candidats qui ont soumis leurs dossiers est réalisée via une application java (Ensemble de classes). Les informations concernant un candidat sont : le numéro de la carte d'identité nationale (cin), le nom (nom), le prénom (prenom), le diplôme du Baccalauréat (bac) et le diplôme du Technicien spécialisé (tc). On considère les classes suivantes :

```
Public class Diplome{
    private String type;
    private String specialite;
    private int annee;
    private double note;
    Diplome(String t,String s,int a,double n)
    {type=t;specialite=s;annee=a;note=n;}
    public String toString()
    {String ch="Type: "+type+" Spécialité: "+specialite+"
      Année : "+annee+" Note: "+note; return(ch);}
    String getType(){return type;}
    String getSpecialite(){return specialite;}
    String getAnnee(){return annee;}
    String getNote(){return note;}
}

public class Personne{
    private String nom, prenom, cin;
    Personne(String n,String p,String c)
    {nom=n ;prenom=p ;cin=c ;}
    Static String clavier()
    {Scanner cl = new Scanner(System.in) ; return cl.nextLine();}
    public String toString()
    { String ch=" Nom : "+nom+" Prénom : "+prenom+" CIN : "+cin+"
      return ch ;}
    String getNom(){return nom;}
    String getPrenom(){return prenom;}
    String getCIN(){return cin;}
}
```

1) Ecrire une classe **Candidat** qui hérite de la classe **Personne** contenant: (3pts)

- Le nom du candidat (**nom** : String)
- Le prénom du candidat (**prenom** : String).
- Le numéro de la carte d'identité nationale du candidat (**cin** :String).
- Le diplôme du Baccalauréat (**bac** :Diplome).
- Le diplôme de Technicien Spécialisé (**tc** :Diplome).
- Un constructeur **Candidat** pour initialiser le **nom**, le **prenom** et le **cin** du candidat.
- Une méthode **lire()** pour compléter les informations du candidat (**bac** et **tc**).
- La méthode **toString()** qui retourne une chaîne de caractères contenant toutes les informations du candidat.

- ✓ 2) Ecrire une classe **Concours** répondant au cahier des charges suivant : (3pts)
- ✓ ■ Le nombre maximum des candidats (**max**: int).
 - ✓ ■ Un tableau des candidats (**table**[]): **Candidat**).
 - ✓ ■ Le nombre des candidats enregistrés (**nbe**: int).
 - ✓ ■ Un constructeur qui prend en paramètre un entier pour initialiser le nombre maximum des candidats qui peuvent participer concours.
 - ✓ ■ Une méthode **boolean ajouteCandidat(Candidat can)** qui ajoute un candidat au tableau « **table** », la méthode renvoie **true** si l'ajout s'est correctement effectué, **false** sinon.
 - ✓ ■ Une méthode **afficher(Candidat[] tab, int d)** qui permettra d'afficher l'ensemble des candidats enregistrés dans un tableau de type **Candidat**.
- ✓ 3) Réaliser une application **JAVA** pour tester la classe **Concours**. (2pts)
- 4) Enrichir la classe **Concours** d'une méthode **int NoteCandidat(Candidat can)** qui permet de retourner une notation de présélection calculée selon le barème mentionné dans le tableau suivant : (2pts)
- | | 10≤note<12 | 12≤note<14 | 14≤note<16 | 16≤note |
|-----------------------|------------|------------|------------|---------|
| Baccalauréat | 1 | 2 | 3 | 4 |
| Technicien Spécialisé | 2 | 4 | 6 | 8 |
- ✓ 5) Ajouter à la classe **Concours** une méthode **int[] NotationCandidats()** qui retourne un tableau des entiers contenant la notation des candidats selon le barème ci-dessus. (2pts)
- ✓ 6) Ajouter à la classe **Concours** une méthode **void suppression()** qui supprime les candidats hors profile. les candidats admis sont les étudiants qui ont un baccalauréat dont la spécialité est (« sciences mathématiques » ou bien « sciences expérimentales ») et un diplôme de technicien spécialisé dont la spécialité doit figurer le mot (« informatique »). (2pts)
- 7) Ajouter à la classe **Concours** une méthode **void ordreMerite()** qui affiche les candidats admis pour passer le concours par ordre de mérite. (2pts)

Exercice 2 : (4 points)

On suppose qu'on a une compétition internationale des athlètes dont l'objectif est de faire un ensemble de tours autour d'un circuit, chaque athlète est identifié par les informations suivantes : un numéro (**num**) attribué par le comité d'organisation, le nom (**nm**), le prénom (**pm**), l'âge (**ag**) et le symbole du pays (**sym**). La simulation de la compétition sera réalisée à l'aide des threads, où chaque athlète est un thread. Sur l'écran d'affichage, et pour chaque tour, il faut afficher le numéro, le symbole du pays et le nombre de tours concernant un athlète. A la fin de la compétition, les trois premiers athlètes doivent être affichés (affichage des numéros des athlètes). (Le nombre des athlètes est 8 et le nombre de tours est 10).


```
public class Candidat extends Personne
```

```
{  
    Diplome tc, bac;
```

```
    Candidat(String nom, String prenom, String ein)  
    {  
        super(nom, prenom, ein);  
    }  
}
```

```
public void lire()  
{  
    System.out.println("Votre Diplome de Bac");  
    System.out.println("de type de diplome :");  
    String tp = clavier();  
    System.out.println("Entrez votre specialite :");  
    String cp = clavier();  
    System.out.println("Entrez l'annee");  
    int Annee = Integer.parseInt(clavier());  
    System.out.println("Entrez la note");  
    double note = Integer.parseInt(clavier());  
    bac = new Diplome(tp, cp, an, nt);  
    System.out.println("Entrez des informations  
sur votre diplome du technicien specialise");  
    tc = new Diplome(tp, cpe, ann, nt);  
}
```

```
✓ public void String toString()  
    { String ch;  
      ch = super.toString() + " " + doc.toString() + "  
        de.toString();  
      return ch;  
    }
```

```
public class Concours
```

```
{  
    int max;
```

```
    Candidat[] table;
```

```
    int nbe;
```

```
    Concours (int max):
```

```
    {  
        this.max = max;
```

```
        table = new Candidat[max];
```

```
        nbe = 0;
```

```
    }  
    public boolean ajoutCandidat(Candidat can)  
    {  
        boolean res = false;
```

```
        if (nbe < max)
```

```
        {  
            table[nbe] = can; nbe++;
```

```
            res = true;
```

```
        }
```

```
        return res;
```

```
    }  
    public void afficher(Candidat[] tab, int d)
```

```
    {
```

```
        for (int i = 0; i < d; i++)
```

```
        {  
            System.out.println (tab[i].
```

```
                toString());
```

```
        }  
    }  
    public int getNbe() { return nbe; }
```

→ Classe Principale

```
public class Principale  
{
```

```
    public static void main (String[] args)  
    {
```

```
        System.out.println("Veuillez entrer le nombre maximum des  
        candidats :");
```

```
        int max = Integer.parseInt(Personne.clavier());
```

```
        System.out.println("Veuillez entrer le nombre de  
        candidat enregistrés");
```

```
        int nbe = Integer.parseInt(Personne.clavier());
```

```
        Concours c = new Concours(max);
```

```
        int i = 1;
```

```
        boolean reussite = true;
```

```
        Candidat can;
```

```
        while (i <= nbe && reussite == true)  
        {
```

```
            System.out.println("Veuillez entrer le nom de candidat :");
```

```
            String nm = Personne.clavier();
```

```
            System.out.println("Veuillez entrer le prenom de  
            candidat :");
```

```
            String pr = Personne.clavier();
```

```
            System.out.println("Veuillez entrer le numero de car  
            nationale CIN :");
```

```
            String cin = Personne.clavier();
```

```
            can = new Candidat(nm, pr, cin);
```

can.lire();

reussite = c.ajouteCandidat(can);
i++;

}

System.out.println(" des candidats enregistrés sont:");

c.afficher(c.table, c.getNbe());

}

}

40/

```
public int NoteCandidat(Candidat can)
```

```
{
    int not = 0; t = 0;
```

```
if (can.bac.getType().equals("bacataueret"))
```

```
{
    t = 1;
    can.bac.getNote();
}
```

```
can.bac.getNote() <= 2 & can.bac.getNote() >
```

```
if (can.bac.getNote() >= 10 && can.bac.getNote() < 12)
```

```
{
    n = 1;
```

```
}
```

```
else if (can.bac.getNote() >= 12 && can.bac.getNote() < 14)
```

```
{
```

```
n = 2;
```

```
}
```

```
else if (can.bac.getNote() >= 14 && can.bac.getNote() < 16)
```

```
{
```

```
n = 3;
```

```
}
```

```
else if (can.bac.getNote() >= 16) { notation = 4; }
```

```
}
```

```
else if (can.bac
```


double noteTC = can.tc.getNote();

if (noteTC >= 10 && noteTC < 12) t = 2;

else if (noteTC >= 12 && noteTC < 14) t = 4;

else if (noteTC >= 14 && noteTC < 16) t = 6;

else if (noteTC >= 16) t = 8;

return t + b;

}

5/

public int[] NotationCandidates()

{

int[] notation = new

for (int i = 0 ; i < nbe ; i++)

{

notation[i] = NoteCandidat(table[i]);

}

return notation;

}

69

```
public void suppression()
```

```
{
```

```
    boolean b1 = false;
```

```
    boolean b2 = false;
```

```
    (int i = 0; i < nbe; i++)
```

```
    {
```

```
        if (table[i].bac.getSpecialite().equals("sciences  
mathématiques") || table[i].bac.getSpecialite().  
            equals("sciences expérimentales"))
```

```
            b1 = true;
```

```
        if (table[i].tc.getSpecialite().indexOf  
            ("Informatique") != -1)
```

```
            b2 = true;
```

```
        if (b1 == false || b2 == false)
```

```
        {
```

```
            for (int j = i + 1; j < nbe; j++)
```

```
            {
```

```
                table[j] = table[j + 1];
```

```
                nbe--;
```

```
            }
```

```
        }
```

79
public void ordreMerite()

```
{
    int i, j, y;
    int a[];
    Candidat x;
    suppression();
    a = NotationCandidat();
    for (i=0; i < nbe-1; i++)
    {
        max = i;
        for (j=i; j < nbe; j++)
        {
            if (a[j] > a[max])
                max = j;
            if (max != i)
            {
                y = a[i];
                a[i] = a[max];
                a[max] = y;
                x = table[i];
                table[i] = table[max];
                table[max] = x;
            }
        }
    }
}
```

```
System.out.println("Candidats convoqués pour passer  
le concours :");
{ afficher(table, nbe);
  //methode
}
```