



-----  
**Contrôle (Module : Java)**  
**(1h30min)**  
-----

E:

O **Exercice 1 :**

d  
z  
c

Dans le cadre de la digitalisation des services de la Faculté des Sciences et Techniques de Béni Mellal, l'établissement veut développer une plateforme pour gérer le service « Estudiantine » via une application Java, qui offre aux étudiants la possibilité de : consulter les emplois du temps et les avis, demander l'attestation d'inscription et les relevés des notes, ...etc. L'accès à la plateforme nécessite d'avoir un compte. La création d'un compte exige la saisie de quelques informations, à savoir : le code national de l'étudiant (cne), le nom (nm), le prénom (pr), le nom d'utilisateur (login), le mot de passe (pwd) et d'autres. L'application contiendra un ensemble de classes, à savoir : Compte, Etudiant et GestionFST.

**1) Ecrire la classe Etudiant :**

(6pts)

- o Le code national de l'étudiant (cne : int)
- o Le nom de la filière (fil : String )
- o Les notes des six semestres (notes : double[])
- o Un constructeur Etudiant(int,String) pour initialiser le code national de l'étudiant et la filière
- o Une méthode String clavier() qui permet de saisir et retourner une chaîne de caractères
- o Une méthode void saisirNotes() qui permet de saisir les notes des six semestres
- o Une méthode String toString() qui retourne une chaîne de caractères contenant la filière et les notes des semestres
- o Une méthode boolean valideSemestres() qui retourne true dans le cas où l'étudiant a validé tous les semestres et false dans le cas inverse. Un semestre est validé si la note du semestre est supérieure ou égale à 10.

**2) Ecrire une classe Compte répondant au cahier des charges suivant :**

(7pts)

- o Le code national de l'étudiant (cne : int)
- o Le nom de l'étudiant (nm : String)
- o Le prénom de l'étudiant (pr : String)
- o Le nom de l'utilisateur (login : String)
- o Le mot de passe (pwd : String)
- o Un constructeur Compte(int) pour initialiser le code national de l'étudiant.
- o Une méthode void creerCompte() qui permet de saisir le reste des attributs. Pour plus de sécurité, il faut appliquer le chiffrement au mot de passe avant de le stocker.
- o Une méthode void toString() qui retourne une chaîne de caractères contenant les attributs de la classe.
- o Une méthode String chiffrer(String,int) qui retourne une chaîne de caractères chiffrée d'une chaîne de caractères passée en paramètre et qui ne contient que les alphabets. Le principe du chiffrement applique le décalage à droite circulaire selon un nombre de pas passé en paramètre. Exemple : Pour une chaîne de caractères "FST" et un pas de décalage (3), le chiffrement à retourner est : "CPQ".



- Les alphabets :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

- Les alphabets après un décalage à droite circulaire d'un pas=3 :

X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

- Pour la chaîne de caractères à chiffrer : "FST"
  - La position de 'F' dans les alphabets est : 6, mais après le décalage la position 6 correspond à 'C'.
  - La position de 'S' dans les alphabets est : 19, mais après le décalage la position 19 correspond à 'P'.
  - La position de 'T' dans les alphabets est : 20, mais après le décalage la position 20 correspond à 'Q'.

NB : - `toCharArray()` : une méthode de `String` qui permet de convertir `String` en `char[]`  
- `toString()` : une méthode de `Character` qui permet de convertir `char[]` en `String`

### 3) Ecrire la classe `GestionFST` :

(7pts)

- o Le nombre maximum des comptes (`max : int`)
- o La liste des comptes (`GComptes : List<Compte>`)
- o La liste des étudiants (`GEtud : List<Etudiant>`)
- o Un constructeur `GestionFST(int)` pour initialiser le nombre maximum des comptes
- o Une méthode boolean `ajouterCompte(Compte c)` qui permet d'ajouter un compte à la liste `GComptes`
- o Une méthode boolean `ajouterEtud(Etudiant e)` qui permet d'ajouter un étudiant à la liste `GEtud`
- o Une méthode `int rechercherEtud(String,String)` qui permet de retourner le cne d'un étudiant identifié par le nom d'utilisateur et le mot de passe passés en paramètres
- o Une méthode `void relevNotes(String,String)` qui crée un fichier représentant le relevé des notes d'un étudiant identifié par le nom d'utilisateur et le mot de passe passés en paramètres et qui a validé tous les semestres