

Exercice 1 :

1. On reprend la classe Etudiant donnée dans l'énoncer
- 2.

```
class Exec
{
    public static void main(String[] args) {
        Etudiant etu = new Etudiant("CHAPUS","Louka",21);

        etu.ajouterNote(17);
        etu.ajouterNote(15);

        etu.afficherNotes();
    }
}
```

Figure [1] classe Exec

```
louka@surface-laptop: /mnt/c/Users/louka/Desktop/Github/esirem/3A/Java/TP4$ java Exec
17      15
```

Figure [2] Résultat après exécution

On ajoute simplement les notes à l'objet etu et on les affiche.

- 3.

On risque d'ajouter des notes en dehors de la plage de note autorisé, soit en dessous de 0 ou au-dessus de 20. Il suffit d'ajouter une condition sur la note dans la fonction ajouterNote.

4.

```
public void ajouterNote (int note ) {  
    if(note < 0 || note > 20){  
        note = 0;  
    }  
    notes [ nbNotes ] = note ;  
    nbNotes ++;  
}
```

Figure [3] Correction de la méthode ajouterNote

On modifie la méthode ajouterNote pour que si la note est plus petite que 0 ou plus grande que 20, alors on met la note par défaut à 0, et on ajoute la note au tableau des notes.

Exercice 2 :

1.

```
public void ajouterNote (int note ) throws ErreurNotes {  
    if ( ( nbNotes >= nbMatières ) || ( note < 0 || note > 20 ) ) {  
        throw new ErreurNotes ();  
    }  
    notes [ nbNotes ] = note ;  
    nbNotes ++;  
}
```

Figure [4] Méthode ajouterNote mise à jour

Cette fois-ci on utilise la classe ErreurNotes pour renvoyer l'erreur dans le programme si jamais les conditions ne sont pas remplies alors on crée un nouvel objet ErreurNotes et dans le constructeur on à le message d'erreur à afficher.

2.

```
class Exec  
{  
    public static void main(String[] args) {  
        Etudiant etu = new Etudiant("CHAPUS","Louka",21);  
  
        try {  
            etu . ajouterNote (12) ;  
            etu . ajouterNote (9);  
            etu . ajouterNote (7);  
            etu . ajouterNote (19) ;  
        }  
        catch ( ErreurNotes e){  
            System . out . println (" Gestionnaire d'exceptions demandeE ... ");  
        }  
    }  
}
```

Figure [5] Classe Exec

```
Vous avez sollicité un gestionnaire d'exceptions ? Je ne suis pas tout à fait prêt , mais je répond quand même  
Gestionnaire d'exceptions demandé ...
```

Figure [6] Après exécution de la classe Exec

On a un message d'erreur car on essaye d'ajouter 4 notes à l'étudiant alors qu'il ne peut avoir que 3 notes au total.

Exercice 3 :

1.

```
public void ajouterNote (int note ) throws ErreurNotes {  
    if ( ( nbNotes >= nbMatières ) || ( note < 0 || note > 20 ) ) {  
        throw new ErreurNotes (nbNotes,nbMatières,note);  
    }  
    notes [ nbNotes ] = note ;  
    nbNotes ++;  
}
```

Figure [7] modification de la méthode ajouterNote

Maintenant on passe simplement les valeurs des notes, le nombre de notes et le nombre de matière dans la création de l'objet ErreurNote pour que celui-ci détermine la cause exacte de l'erreur et qu'il affiche le bon message d'erreur.

2.

```
catch ( ErreurNotes e){  
    System . out . println ( " Gestionnaire d'exceptions demandé ... " );  
    System . out . println ( e.recuperMessageErreur() );  
}
```

Figure [8] Modification de la classe Exec

```
Gestionnaire d'exceptions demandé ...  
Nombre de notes à saisir supérieur au nombre de matières .
```

Figure [9] Résultat après exécution

On appelle simplement la méthode `recuperMessageErreur` de la classe `ErreurNotes` et on l'affiche grâce au `println`. On voit bien que l'erreur provient du nombre de notes et non pas de la valeur des notes.