

TD 3

1. Exceptions

Soit le programme suivant :

```
public class Exercicel {
    private final int MAX = 99;
    private int tableau[] = new int[MAX];

    public Exercicel (int n) {
        for (int i=0; i<n; i++)
            tableau[i] = i;
    }

    public int sommeDesNPremiers(int n) {
        int resultat = 0;

        for (int i=0; i<n ; i++)
            resultat += tableau[i];
        return resultat;
    }

    public static void main(String args[]) {
        Exercicel ex1 = new Exercicel(10);
        int n = Integer.parseInt(args[0]);
        int resultat;

        resultat = ex1.sommeDesNPremiers(n);
        System.out.print("La somme des "+args[0]+" premiers nombres ");
        System.out.println("est : "+resultat);
    }
}
```

1. Donner la liste des erreurs d'exécutions possibles
2. Proposer des modifications permettant de gérer toutes ces erreurs (en utilisant uniquement les Exceptions).

2. Exceptions : mots de passe

Ecrire un programme qui teste l'efficacité d'un mot de passe saisi par l'utilisateur

1. Implémenter les exceptions
 - a. *MDPVide*, qui se produit lorsque l'utilisateur saisi un mot de passe vide
 - b. *MDPTaille*, qui se produit lorsque l'utilisateur saisi un mot de passe composé de moins de 8 caractères ou de plus de 30 caractères
 - c. *MDPForce*, qui se produit lorsque le mot de passe ne contient que des lettres, i.e.

< la chaine >.matches("[a-zA-Z]*") renvoie true

2. Ecrire la méthode **saisirMotDepasse** qui récupère et renvoie le mot de passe saisi par l'utilisateur. La méthode lève une des exceptions précédentes si le mot de passe saisi n'est pas correct.

Pour rappel, la lecture d'une chaîne de caractères se fait avec
String chaine = new Scanner(System.in).next() ;

3. Ecrire un programme Main, qui demande en boucle un mot de passe à l'utilisateur jusqu'à recevoir un mot de passe correct

3. Cours

- Qu'est ce que la sérialisation ?
- Expliquer les étapes pour la mettre en place.

4. La sérialisation

1. Expliquer le fonctionnement de la sérialisation.
2. Indiquer le contenu du fichier après exécutions des opérations suivantes
 - a. Point P= new Point(1,2) ;
 - b. Ouverture d'un accès à un fichier F
 - c. Sérialisation de P dans le fichier F
 - d. P.translater(1,1)
 - e. Sérialisation de P dans le fichier F
 - f. Fermeture du fichier F

5. Clonage et Sérialisation

Soient les classes suivantes

```
public class Patronyme {
    private String prenom;
    private String nom; public
    Patronyme(String p, String n) {
        this.prenom = p;
        this.nom = n;
    }
}
```

```
public class Personne{
    private Patronyme patronyme;
    private int age;

    public Personne(Patronyme p,int a) {
        this.patronyme = patronyme;
        this.age = age;
    }
}
```

1. Proposer les modifications permettant de cloner en profondeur un objet de type Personne
2. Proposer les modifications permettant de sérialiser un objet de type Personne
3. Ecrire un programme qui
 - a. Crée une personne p0
 - b. crée un clone p1 de p0
 - c. sérialise p1 dans un fichier "save.pers"
 - d. Désérialise dans une variable p2 l'objet de type personne contenu dans "save.pers"
4. Que renvoie l'instruction System.out.println(p1.equals(p2))
5. Comment y remédier ?