



TP4 - JAVA : Gestion des exceptions

BACHOUR Peter

26 Mai 2020

Table des matières

1	Introduction	2
2	Définition	3
3	Contrôle de la saisie des notes	4
4	Déclenchement d'une exception	5
5	Transmission d'informations au gestionnaire d'exceptions	6
6	Conclusion	7

Table des figures

1	Classe Exec	4
2	Fonction ajouterNote(int note)	4
3	Fonction ajouterNote(int note) avec Exception	5
4	Try et Catch des exceptions	5
5	Fonction ajouterNote(int note) avec passage d'information	6
6	Try et Catch des exceptions avec affichage du type	6

1 Introduction

Dans le cadre du quatrième TP en JAVA, nous allons nous focaliser sur la gestion des exceptions ainsi que comment try et catch les erreurs possible.

2 Définition

La gestion des exceptions en Java se fait par l'utilisation de trois mots clefs `try`, `catch`, `throw`. Le mot clef `try` sert à définir un bloc dans lequel les exceptions sont susceptibles d'être capturées. La capture des exceptions est confiée à des gestionnaires d'exception. Après la définition de ce bloc se trouve la liste des gestionnaires d'exception. Un gestionnaire d'exception est défini en utilisant le mot réservé `catch`. La liste des gestionnaires d'exception ressemble à un ensemble de fonctions surchargées. Une exception est levée en utilisant le mot clef `throw`.

3 Contrôle de la saisie des notes

Comme la classe Etudiant le montre, chaque étudiant a le droit d'avoir jusqu'à 3 notes. Alors les risques qu'on pourra affronter seront alors où l'ajout de plus que 3 notes à un étudiant ou l'ajout d'une note interdite à un étudiant (plus petit que 0 ou plus grand que 20).

```
Etudiant etu = new Etudiant( nom: "Bachour", prenom: "Peter", age: 21);

etu.ajouterNote(12);
etu.ajouterNote(9);

etu.afficherNotes();
```

FIGURE 1: Classe Exec

Alors pour limiter les risques, on a recours à la condition suivante :

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

FIGURE 2: Fonction ajouterNote(int note)

4 Déclenchement d'une exception

Afin de gérer plus facilement les erreurs qui apparaîtront, un message d'erreur sera affiché à l'utilisateur lorsqu'une erreur aura lieu et cela est fait grâce à la classe `ErreurNotes`. Alors afin de **throw** une nouvelle exception, on devra changer la condition dans **if** à l'inverse. Maintenant, si la condition dans le block du **if** est vrai, une nouvelle exception sera afficher et l'exécution de la fonction **ajouterNote** s'arrêtera.

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

FIGURE 3: Fonction `ajouterNote(int note)` avec Exception

Maintenant que les exceptions ont été "**throw**", il devra les "attrapper" dans le main, et cela est fait grâce au block try-catch. En effet, les fonctions sont appelées dans la partie **try**, et si une exceptions a été "attrapper", elle sera passé à la partie **catch** qui l'affichera.

```
try {
    etu.ajouterNote(12);
    etu.ajouterNote(9);
    etu.ajouterNote(7);
    etu.ajouterNote(19);
}
catch (ErreurNotes e){
    System.out.println("Gestionnaire d'exceptions demandé...");
}
```

FIGURE 4: Try et Catch des exceptions

5 Transmission d'informations au gestionnaire d'exceptions

On a pu remarquer que le try peut catch deux types d'erreurs, la première étant que la note peut être plus petite que 0 ou plus grande que 20, et la deuxième est que le nombre de note est supérieure à 3. C'est pour cela, il faudra définir le type d'exception.

```
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 5: Fonction ajouterNote(int note) avec passage d'information

```
try {  
    etu.ajouterNote(12);  
    etu.ajouterNote(9);  
    etu.ajouterNote(7);  
    etu.ajouterNote(19);  
}  
catch (ErreurNotes e){  
    System.out.println(e.recuperMessageErreur());  
}
```

FIGURE 6: Try et Catch des exceptions avec affichage du type

6 Conclusion

En conclusion, à travers ce TP, nous avons découvert la gestion des exceptions et comment transmettre les exceptions au gestionnaire.