

TP exception

- La gestion des exceptions en Java -



Sommaire

1) Objectifs :

Mettre en oeuvre les mécanismes liés aux **exceptions** en Java. Consolider les expérimentations faites dans le support de cours dédié et maîtriser définitivement tous les concepts autour des exceptions.

2) Vocabulaire utilisé :

La hiérarchie des exceptions , **exceptions non contrôlées**, **contrôlées**, les *Errors*. Le bloc *try/catch*, la clause *throw*, la clause *throws*, la clause *finally*, les exceptions de l'API Java et les exceptions créées par le développeur.

3) Environnement technique :

J2SE, JDK. Un EDI (type *Eclipse/NetBeans*).

La documentation **Java Oracle** :

<https://docs.oracle.com/javase/tutorial/essential/exceptions/index.html>

exo1: appliqué la gestion des erreurs pour lever une exception si l'utilisateur veut atteindre un indice hors d'un tableau d'int.

```
int[]tabint = new int[] { 2, 6, 5, 9, 8,10, 54, 21, 55 };
```

```
taille du Tableau: 9
Saisissez un index de votre Tableau:
12
Out of range
Press any key to exit
```

exo2: Sécuriser l'utilisation d'une calculatrice

- 1) Si un opérateur arithmétique erroné (autre que : '+', '-', '/', '*') est saisi une exception est lancée.
- 2) Pour les opérandes si une valeur non numérique est saisi une exception est lancée.
- 3) Une division par 0 peut survenir, gérer la capture de l'exception et afficher l'incident

Calculatrice
■ sc: Scanner
▲ encore: boolean
▲ valeurSaisie: String
▲ entierSaisi: int
● Calculatrice()
● saisirOperateur():char
● saisirEntier():int
● calculer(char,int,int):int
● puissance(int,int):long

exemple d'utilisation de la classe Claculatrice

```
Calculatrice calc = new Calculatrice();
char Operateur = calc.saisirOperateur();
int leftValue = calc.saisirEntier();
int rightValue = calc.saisirEntier();
int resultat = calc.calculer( Operateur, leftValue, rightValue);
if (resultat == -1)
    System.out.println("saisie incorrecte code erreur -1");
else
    out.println("resultat:"+leftValue+Operateur+rightValue+"="+ resultat);
```

```
Veillez saisir un operateur?%
|Veillez entrer un operateur valide SVP

Veillez saisir un operateur?/
Veillez saisir un entier SVP ?2
Veillez saisir un entier SVP ?0
pas de Division Par Zéro
saisie incorrecte code erreur -1

Veillez saisir un operateur?/
Veillez saisir un entier SVP ?A
|la valeur saisie n'est pas valide recommencez

Veillez saisir un operateur?/
Veillez saisir un entier SVP ?4
Veillez saisir un entier SVP ?2
resultat : 4/2 = 2
```

A noter il est préférable d'utiliser le type double notamment pour les divisions $3/2 = 1.5$
Mais avec ce type, l'exception `ArithmeticException` ne serait pas mise en valeur

exo 3 BirthDate

Surveillance de la construction d'un objet

Ecrire un programme qui impose la saisie d'une date de type String:

La saisie contient 8 digits

les 2 premiers représente le jour,

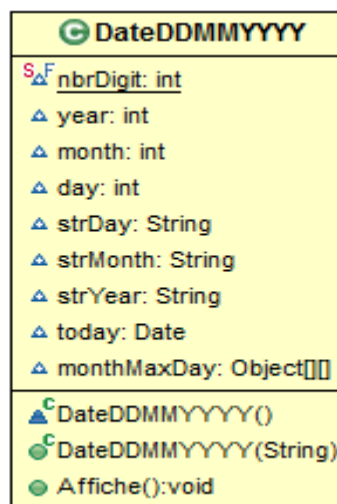
les 2 suivants représente le mois

et les 4 restants sont l'année.

Le programme doit tester la validité de la date saisie.

Si la date saisie n'est pas valide entrez la date du jour.

Méthodologie: écrire et utiliser la classe DateDDMMYYYY



Utilisation de la classe:

```
DateDDMMYYYY myBirthDate = null;
String date = sc.next(); //inputting the date in String format
try{
    myBirthDate = new DateDDMMYYYY(date) ;
}
catch (Exception e){
    System.out.println(e.getMessage());
}
finally
{
    if(myBirthDate== null)
        myBirthDate = new DateDDMMYYYY();//par default sans try catch
}
myBirthDate.Affiche();
```

écrire une classe spécifique pour les exceptions

exo 3 BirthDate (suite)

Enter any date in 8 digits (ddmmyyyy) format: 11111111111111111111

saisie incorrecte la date se forme sur 8 digits

Date in dd/mm/yyyy format = 24/02/2020

Enter any date in 8 digits (ddmmyyyy) format: 30022020

saisie jour incorrecte pour:February

Date in dd/mm/yyyy format = 24/02/2020

Enter any date in 8 digits (ddmmyyyy) format: 31042007

saisie jour incorrecte pour:April

Date in dd/mm/yyyy format = 24/02/2020

jour utilisation du programme:

