

TD 5 : Contrats, exceptions

Objectifs pédagogiques : contrats et exceptions.

5.1 Gestion des exceptions

Soit le programme Java de l'Annexe 1.

⇒ Donner les affichages du programme si on lance: `java ExempleSimple`

⇒ Même question en ajoutant les sources de l'Annexe 2 et si on lance le programme:

`java ExempleMultiple`

⇒ Une partie du programme est commentée, elle ne compile pas. Expliquez pourquoi.

5.2 Contrats pour Piles

Dans cet exercice, nous allons utiliser l'approche dite de conception par contrats pour concevoir et implémenter une classe simple pour les structures de données de piles bornées.

Le squelette de la classe est le suivant:

```
public class BoundedStack {  
    private Object[] content;  
    private int size;  
    public BoundedStack(int limit) {  
        content = new Object[(limit<0) ? 0 : limit];  
        size = 0;  
    }  
    public boolean isFull() { return getSize()==getLimit(); }  
    public boolean isEmpty() { return getSize()==0; }  
    public int getSize() { return size; }  
    public int getLimit() { return content.length; }  
    // Suite de la classe  
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13

On souhaite implémenter trois méthodes supplémentaires :

- + `top()` : `Object` qui retourne l'élément en haut de pile,
- + `push(Object o)` : `void` qui ajoute un élément dans la pile,
- + `pop()` : `Object` qui dépile l'élément en haut de pile.

5.2.1 Contractualisation

⇒ Ecrivez, de façon informelle, les contrats pour chacune de ces méthodes en précisant :

- le fournisseur des contrats,
- les clients,
- les prérequis du client,
- les garanties du fournisseur.

5.2.2 Classes d'exception et classe de test unitaire

⇒ Définissez une hiérarchie de classes d'exception permettant de prendre en compte :

- les erreurs génériques sur les piles (`StackException`)
- les erreurs spécifiques: pile vide ou pile pleine

⇒ Définissez également une classe de tests unitaires avec déclaration et initialisation (méthode `@Before`) d'une pile vide, d'une pile à moitié pleine et d'une pile pleine.

5.2.3 Implémentation

⇒ Donnez le code des trois méthodes en gérant correctement les exceptions. Pour chaque méthode, écrivez un test unitaire permettant de :

- tester les prérequis (au moins un cas prérequis assuré, un autre cas prérequis faux)
- tester les garanties (si les prérequis sont vérifiés)

5.3 Contrats pour Files

⇒ Selon le procédé décrit à l'exercice 2, **concevez et implémentez une classe `BoundedFifo`** (file bornée) proposant une sémantique de type premier entrant premier sortant (*First In First Out*) avec au moins une méthode `push()` pour ajouter un élément et `fetch()` pour retirer l'élément le plus ancien.

5.4 Annexe 1

MonException.java

```
public class MonException extends Exception
{
    public MonException(String message) {
        super(message);
    }
}
```

Générateur.java

```
public class Générateur {
    private String genstr;
    public Générateur(String genstr) {
        this.genstr = genstr;
    }

    public void tryMe(String str) throws MonException {
        if(str.equals(genstr))
            throw new MonException("Oups, l'exception est levée");
    }
}
```

ExempleSimple.java

```
public class ExempleSimple {
    public void premierTest(Générateur gen) {
        System.out.println("Premier test:");
        System.out.println("-----");
        String chaine = "alea jacta est";
        try {
            System.out.println("un");
            gen.tryMe(chaine);
            System.out.println("deux");
        } catch (MonException e) {
            System.out.println("trois");
        }
    }
}
```

```

        System.out.println("quatre");
    }

    public void secondTest(Generateur gen) {
        System.out.println("Second test:");
        System.out.println("-----");
        String chaine = "aller a la gare de l'est";
        try {
            System.out.println("un");
            gen.tryMe(chaine);
            System.out.println("deux");
        } catch (MonException e) {
            System.out.println("trois");
        }
        System.out.println("quatre");
    }

    public void troisiemeTest(Generateur gen) throws MonException {
        System.out.println("Troisieme test:");
        System.out.println("-----");
        String chaine = "alea jacta est";
        System.out.println("un");
        gen.tryMe(chaine);
        System.out.println("deux");
    }

    public void quatriemeTest(Generateur gen) throws MonException {
        System.out.println("Quatrieme test:");
        System.out.println("-----");
        String chaine = "aller a la gare de l'est";
        System.out.println("un");
        gen.tryMe(chaine);
        System.out.println("deux");
    }

    public void cinquiemeTest(Generateur gen) throws MonException {
        System.out.println("Cinquieme test:");
        System.out.println("-----");
        String chaine = "alea jacta est";
        try {
            System.out.println("un");
            gen.tryMe(chaine);
            System.out.println("deux");
        } catch (MonException e) {
            System.out.println("trois");
            throw e;
        } finally {
            System.out.println("trois bis");
        }
        System.out.println("quatre");
    }

    public void sixiemeTest(Generateur gen) throws MonException {
        System.out.println("Sixieme test:");
        System.out.println("-----");
        String chaine = "aller a la gare de l'est";
        try {
            System.out.println("un");
            gen.tryMe(chaine);
            System.out.println("deux");
        } catch (MonException e) {

```

<pre> System.out.println("trois"); throw e; } finally { System.out.println("trois bis"); } System.out.println("quatre"); } public static void main(String[] args) { Generateur gen = new Generateur("alea jacta est"); ExempleSimple exemple = new ExempleSimple(); exemple.premierTest(gen); exemple.secondTest(gen); try { exemple.troisiemeTest(gen); } catch (MonException e) { System.out.println("cinq"); } try { exemple.quatriemeTest(gen); } catch (MonException e) { System.out.println("cinq"); } try { exemple.cinquiemeTest(gen); } catch (MonException e) { System.out.println("cinq"); } try { exemple.sixiemeTest(gen); } catch (MonException e) { System.out.println("cinq"); } } } </pre>	<pre> 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 </pre>
--	--

5.5 Annexe 2

MaSousException1.java

<pre> public class MaSousException1 extends MonException { public MaSousException1(String message) { super("Sous exception 1:" + message); } } </pre>	<pre> 1 2 3 4 5 </pre>
---	------------------------

MaSousException2.java

<pre> public class MaSousException2 extends MonException { public MaSousException2(String message) { super("Sous exception 2:" + message); } } </pre>	<pre> 1 2 3 4 5 </pre>
---	------------------------

GenMultiple.java

<pre> public class GenMultiple extends Generateur { private String genstr1; } </pre>	<pre> 1 2 </pre>
--	------------------

private String genstr2;	3
	4
public GenMultiple(String gen, String gen1, String gen2) {	5
super (gen);	6
genstr1 = gen1;	7
genstr2 = gen2;	8
}	9
	10
public void tryMe(String str) throws MonException {	11
super .tryMe(str);	12
if (str.equals(genstr1))	13
throw new MaSousException1("Aie aie aie ! Ca fait mal");	14
else if (str.equals(genstr2))	15
throw new MaSousException2("Ouille, ouch, itai!");	16
}	17
}	18

ExempleMultiple.java

public class ExempleMultiple {	1
public void premierTest(Generateur gen) {	2
System.out.println("Premier test multipl:");	3
System.out.println("-----");	4
String chaine = "alea jacta est";	5
try {	6
System.out.println("un");	7
gen.tryMe(chaine);	8
System.out.println("deux");	9
} catch (MaSousException1 e) {	10
System.out.println("trois");	11
} catch (MaSousException2 e) {	12
System.out.println("trois bis");	13
} catch (MonException e) {	14
System.out.println("trois ter");	15
}	16
System.out.println("quatre");	17
}	18
	19
public void secondTest(Generateur gen) {	20
System.out.println("Second test multiple:");	21
System.out.println("-----");	22
String chaine = "alea jacta ouest";	23
try {	24
System.out.println("un");	25
gen.tryMe(chaine);	26
System.out.println("deux");	27
} catch (MaSousException1 e) {	28
System.out.println("trois");	29
} catch (MaSousException2 e) {	30
System.out.println("trois bis");	31
} catch (MonException e) {	32
System.out.println("trois ter");	33
}	34
System.out.println("quatre");	35
}	36
	37
public void troisiemeTest(Generateur gen) {	38
System.out.println("Troisieme test multi:");	39
System.out.println("-----");	40
String chaine = "chi va piano va sano";	41
try {	42

```

        System.out.println("un");
        gen.tryMe(chaine);
        System.out.println("deux");
    } catch (MaSousException1 e) {
        System.out.println("trois");
    } catch (MaSousException2 e) {
        System.out.println("trois bis");
    } catch (MonException e) {
        System.out.println("trois ter");
    }
    System.out.println("quatre");
}

public void quatriemeTest(Generateur gen) {
    System.out.println("Quatrieme test multi:");
    System.out.println("-----");
    String chaine = "Aller a saint-lazare";
    try {
        System.out.println("un");
        gen.tryMe(chaine);
        System.out.println("deux");
    } catch (MaSousException1 e) {
        System.out.println("trois");
    } catch (MaSousException2 e) {
        System.out.println("trois bis");
    } catch (MonException e) {
        System.out.println("trois ter");
    }
    System.out.println("quatre");
}

/* ne compile pas
public void cinquiemeTest(Generateur gen) {
    System.out.println("Cinquieme test multi:");
    System.out.println("-----");
    String chaine = "alea jacta ouest";
    try {
        gen.tryMe(chaine);
    } catch (MonException e) {
        System.out.println("hip");
    } catch (MaSousException1 e) {
        System.out.println("hop");
    } catch (MaSousException2 e) {
        System.out.println("hap");
    }
    System.out.println("quatre");
} */

public static void main(String[] args) {
    GenMultiple gen = new GenMultiple("alea jacta est", "chi va piano va sano", "alea
        jacta ouest");
    ExempleMultiple exemple = new ExempleMultiple();
    exemple.premierTest(gen);
    exemple.secondTest(gen);
    exemple.troisiemeTest(gen);
    exemple.quatriemeTest(gen);
    //exemple.cinquiemeTest(gen);
    //System.out.println();
}
}

```