

Préparation à E5	2 <sup>ème</sup> BTS SIO SLAM
SLAM4-5	J-C HERON
<b>CONTENU : Solutions applicatives</b>	<b>Evaluation n°1</b>
Durée max	1 heure 30
Domaines abordés	<p>D2.2 - Gestion des incidents et des demandes d'assistance</p> <p>D2.3 - Gestion des problèmes et des changements</p> <p>D4.1 - Conception et réalisation d'une solution applicative</p>
Fichier	DS1.docx

## Contexte :

Vous travaillez sur le développement de l'application Web **de report d'incidents**.

Votre chef de projet vous confie quelques tâches à effectuer, ainsi qu'un certain nombre de documents, qu'il vous appartient d'étudier attentivement, et d'utiliser avec intelligence.

## Partie 1

### Résolution d'un bogue :

Lors de la saisie d'un nouvel utilisateur, une exception est déclenchée, dont on a conservé la trace, fournie en annexe :

1. A partir des éléments qui vous sont donnés, vous devez dans un premier temps résoudre ce problème.
2. Vous apportez ensuite la certitude que le problème a été résolu de manière définitive, en implémentant un test unitaire avec PHPUnit.

## Partie 2

### Implémentation

A partir du diagramme de classes fourni en annexe, commencez l'implémentation de la classe Module :

3. Déclarez l'ensemble des membres de données.
4. Implémentez le constructeur mentionné sur le diagramme, en veillant à ce qu'il construise correctement et de manière complète une instance de Module.

Préparation à E5	2 <sup>ème</sup> BTS SIO SLAM
SLAM4-5	J-C HERON
CONTENU : Solutions applicatives	Evaluation n°1

## Partie 3

### Conception de l'interface de gestion des droits

Vous devez formaliser la logique applicative de l'interface de gestion des droits :

Les droits sont attribués pour des groupes d'utilisateur, sur un module ; chaque module contenant un ensemble de pages. L'interface doit permettre l'ajout ou la suppression de droits. Elle devra limiter les manipulations pour attribuer des droits spécifiques à un ensemble d'utilisateurs appartenant à un groupe.

5. Définir la logique applicative de l'interface de saisie des droits.
6. Préciser ensuite les liens existants entre le cas d'utilisation Gestion des droits et les autres cas d'utilisation de l'application.

Préparation à E5	2 <sup>ème</sup> BTS SIO SLAM
SLAM4-5	J-C HERON
CONTENU : Solutions applicatives	Evaluation n°1

## Contexte : Insertion d'un nouvel utilisateur

Lorsque la table utilisateur est vide, l'insertion d'un utilisateur fonctionne. C'est l'insertion d'un deuxième utilisateur qui provoque l'exception de manière systématique.

### Affichage de l'erreur d'exécution

```
object(PDOException)#22 (8) { ["message":protected]=> string(91) "SQLSTATE[23000]: Integrity
constraint violation: 1062 Duplicate entry '0' for key 'PRIMARY'" ["string":"Exception":private]=>
string(0) "" ["code":protected]=> string(5) "23000" ["file":protected]=> string(42)
"C:\xampp\htdocs\bugReport\technics\DAO.php" ["line":protected]=> int(216)
["trace":"Exception":private]=> array(4) { [0]=> array(6) { ["file"]=> string(42)
"C:\xampp\htdocs\bugReport\technics\DAO.php" ["line"]=> int(216) ["function"]=> string(7)
"execute" ["class"]=> string(12) "PDOStatement" ["type"]=> string(2) "->" ["args"]=> array(0) { } }
[1]=> array(6) { ["file"]=> string(54) "C:\xampp\htdocs\bugReport\controllers\cUtilisateur.php"
["line"]=> int(49) ["function"]=> string(6) "insert" ["class"]=> string(3) "DAO" ["type"]=> string(2)
"::" ["args"]=> array(1) { [0]=> object(Utilisateur)#6 (5) { ["login":"Utilisateur":private]=> string(4)
"unLogin" ["password":"Utilisateur":private]=> string(40)
"bcb58c91e18656345fafda2f2914264135091014" ["mail":"Utilisateur":private]=> string(12)
"moi@gmail.com" ["groupe":"Utilisateur":private]=> object(Groupe)#7 (3) {
["libelle":"Groupe":private]=> NULL ["utilisateurs":"Groupe":private]=> NULL ["id":protected]=>
string(1) "1" } ["id":protected]=> string(0) "" } } } [2]=> array(6) { ["file"]=> string(44)
"C:\xampp\htdocs\bugReport\MainController.php" ["line"]=> int(67) ["function"]=> string(3) "add"
["class"]=> string(12) "cUtilisateur" ["type"]=> string(2) "->" ["args"]=> array(1) { [0]=> &array(1) {
[0]=> string(9) "undefined" } } } [3]=> array(6) { ["file"]=> string(44)
"C:\xampp\htdocs\bugReport\MainController.php" ["line"]=> int(10) ["function"]=> string(3) "run"
["class"]=> string(14) "MainController" ["type"]=> string(2) "->" ["args"]=> array(0) { } } }
["previous":"Exception":private]=> NULL ["errorInfo"]=> array(3) { [0]=> string(5) "23000" [1]=>
int(1062) [2]=> string(37) "Duplicate entry '0' for key 'PRIMARY'" } }
```

### Structure de la table Utilisateur

 Structure

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
<input type="checkbox"/> 1	<u>id</u>	int(11)			Non	Aucune	
<input type="checkbox"/> 2	idGroupe	int(11)			Oui	NULL	
<input type="checkbox"/> 3	login	varchar(30)	utf8_general_ci		Non	Aucune	
<input type="checkbox"/> 4	password	varchar(40)	utf8_general_ci		Non	Aucune	
<input type="checkbox"/> 5	mail	varchar(255)	utf8_general_ci		Non	Aucune	

### Extrait de la classe Utilisateur

```
class Utilisateur extends BaseObject{

    private $login;
    private $password=null;
    private $mail;
    private $groupe;

    public function Utilisateur($id,$login,$password,$mail){
        $this->login=$login;
        $this->password=$password;
        $this->mail=$mail;
    }
}
```

Préparation à E5	2 <sup>ème</sup> BTS SIO SLAM
SLAM4-5	J-C HERON
CONTENU : Solutions applicatives	Evaluation n°1

### Extrait de la classe BaseObject

```
public abstract class BaseObject {
    private int id;

    public BaseObject(int id) {
        super();
        this.id=id;
    }
}
```

### Extrait de la méthode Insert de la classe DAO

```
public static function insert($instance){
    $tableName=OrmUtils::getTableName(get_class($instance));
    $keyAndValues=Reflexion::getPropertiesAndValues($instance);
    $keyAndValues=array_merge($keyAndValues,
OrmUtils::getManyToOneMembersAndValues($instance));
    $sql="INSERT INTO
".$tableName."(".SqlUtils::getInsertFields($keyAndValues).")
VALUES(".SqlUtils::getInsertFieldsValues($keyAndValues).")";
    Logger::log("insert", $sql);
    $statement=DAO::$db->prepareStatement($sql);

    try {
        foreach ($keyAndValues as $key=>$value){
            DAO::$db->bindValueFromStatement($statement,$key,$value);
        }
        $statement->execute();
    } catch (Exception $e) {var_dump($e);
        Logger::error("PDO", $e->getMessage());
    }
}
```

### Autres méthodes publiques de la classe DAO :

```
/**
 * Retourne un tableau d'objets de $className depuis la base de données
 * @param string $className
 * @param string $condition
 * @return multitype:$className
 */
public static function getAll($className,$condition='') ;

/**
 * Retourne une instance de $className depuis la base de données, à partir
 * des valeurs $keyValues de la clé primaire
 * @param String $className
 * @param Array or condition $keyValues
 */
public static function getOne($className,$keyValues) ;

/**
 * Supprime $instance dans la base de données
 * @param Classe $instance instance à supprimer
 */
public static function delete($instance) ;

/**
 * Insère $instance dans la base de données
 * @param Classe $instance instance à insérer
 */
public static function insert($instance) ;
```

Préparation à E5	2 <sup>ème</sup> BTS SIO SLAM
SLAM4-5	J-C HERON
CONTENU : Solutions applicatives	Evaluation n°1

```
/**
 * Met à jour $instance dans la base de données.
 * Attention de ne pas modifier la clé primaire
 * @param Classe $instance instance à modifier
 */
public static function update($instance) ;
```

### Exemples de tests unitaires réalisés à partir de PHPUnit

```
<?php
class FalseTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertFalse(TRUE);
    }
}
?>
```

```
<?php
class EqualsTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertEquals(array('a', 'b', 'c'), array('a', 'c', 'd'));
    }
}
?>
```

## Diagramme des classes métier : gestion des droits

