

PWS - Repository

Le design pattern repository

Pierre Blarre - Florian Rodriguez

Introduction

- Quel est le rôle d'une classe comme Personnage ?
 - Un objet instanciant une classe comme Personnage a pour rôle de *représenter* une ligne présente en BDD.
- Il paraîtrait logique de placer le code du CRUD dans notre classe Personnage, c'est d'ailleurs comme cela que de nombreux programmes sont codés.
- Cependant, cette approche ne respecte pas un principe de conception très utilisé en programmation: La séparation des préoccupations (SoC en anglais pour Separation of Concerns).
 - https://fr.wikipedia.org/wiki/S%C3%A9paration_des_pr%C3%A9occupations
 - Un programme qui intègre les principes de séparation des préoccupations est appelé un programme modulaire, car il sépare son code en différents modules.
 - La SoC ne paraît pas forcément naturelle, car en apparence elle semble compliquer les choses; mais en réalité, c'est grâce aux techniques associées à la SoC qu'il est possible de développer et de maintenir des applications conséquentes.
- *Représenter* est très différent de *gérer*.

Cas d'un personnage

- Dans notre cas, séparer le CRUD de notre classe d'objet Personnage va permettre de respecter la SoC.
- Le fait de séparer le code qui interagit avec la base de données respecte le design pattern appelé Repository.
- Repository signifie dépôt ou entrepôt en anglais. On peut imaginer notre base de données comme un dépôt pour stocker nos personnages.
- Pour respecter le pattern Repository, il faut créer une seconde classe PersonnageRepository qui va contenir le code de son CRUD.
- Ceci évitera de “polluer” la classe Personnage avec du code qui n'est pas en relation avec l'application d'un jeu de combat.
- Le pattern Repository est un modèle de conception qui fournit une couche d'abstraction entre l'application et la couche de persistance des données. Il sépare la logique applicative de la couche d'accès aux données, ce qui permet aux développeurs de remplacer la couche d'accès aux données sans affecter les fonctionnalités de l'application.
- La persistance des données peut être effectuée en utilisant des fichiers, des bases de données, des services web, etc.

La classe PersonnageRepository

- Quelles seront les caractéristiques de mes objets ?
- Quelles seront les fonctionnalités de mes objets ?

« De quoi a besoin un repository pour fonctionner ? »

- une connexion à la base de données (Souvent appelé DAO pour Database Access Object) Quelles fonctionnalités:
 - Enregistrer une nouvelle entité
 - Mettre à jour une entité
 - Supprimer une entité
 - Récupérer une entité
- => CRUD

Exemple de code

Personnage.php

```
<?php
class Personnage
{
    private $nom;
    private $degats;

    public function __construct($nom, $degats)
    {
        $this->nom = $nom;
        $this->degats = $degats;
    }

    public function getNom()
    {
        return $this->nom;
    }

    public function getDegats()
    {
        return $this->degats;
    }

    public function setDegats($degats)
    {
        $this->degats = $degats;
    }
}
```



PersonnagesRepository.php

```
<?php
class PersonnagesRepository
{
    private $db;

    public function __construct($db)
    {
        $this->db = $db;
    }

    public function create(Personnage $personnage)
    {
        $query = $this->db->prepare('INSERT INTO personn
        $query->bindValue(':nom', $personnage->getNom())
        $query->bindValue(':degats', $personnage->getDeg
        $query->execute();
    }

    public function update(Personnage $personnage)
    {
        $query = $this->db->prepare('UPDATE personnage S
        $query->bindValue(':nom', $personnage->getNom())
        $query->bindValue(':degats', $personnage->getDeg
        $query->execute();
    }
}
```



Conclusion

- <https://github.com/MIASHS-UGA/php-poo-intro>
- Du côté du script PHP, chaque enregistrement de la base de données est représenté par un objet possédant une liste d'attributs identique à la liste des colonnes de la table.
- La communication avec la BDD se fait par le biais d'un objet différent de l'objet représentant l'enregistrement (une classe = un rôle). Un tel objet est un repository (dépôt).
- Un repository peut stocker les objets en BDD, mais peut tout-à-fait les stocker sur un autre support (fichier XML, fichier texte, etc.).