

Lara Terzi document

by Lara Terzi

FILE	3643_LARA_TERZI_LARA_TERZI_DOCUMENT_162370_2049255239.DOCX (21.71K)		
TIME SUBMITTED	02-FEB-2019 03:19AM (UTC+0000)	WORD COUNT	931
SUBMISSION ID	100155681	CHARACTER COUNT	8024

Lara Terzi

P148014

BSc Software Engineering

Advanced Web Development

Computing Portfolio

Mike Dacey

Table of Contents

- 1. Knowledge and Understanding Skills..... 3
 - 1.1. Design Pattern 1..... **Error! Bookmark not defined.**
 - 1.1.1. About design pattern 1 **Error! Bookmark not defined.**
 - 1.1.2. PHP examples of design pattern 1 **Error! Bookmark not defined.**
 - 1.2. Design pattern 2..... 5
 - 1.2.1. About design pattern 2 **Error! Bookmark not defined.**
 - 1.2.2. PHP examples of design pattern 2 **Error! Bookmark not defined.**
- 2. Practical Skills..... 6
- 3. References 7
- 4. Appendix 8



1. Knowledge and Understanding Skills

1.1. 'Adapter' Design Pattern

An Adapter design pattern is used when incompatible classes need to be used together. This design allows for classes to adapt to each other and therefore communicate by basically translating the different languages.



Below is an example of how PHP can use the adapter design pattern.

```
<?php
```

```
class Song {  
    private $artist;  
    private $name;  
    function __construct($artist_in, $name_in) {  
        $this->artist = $artist_in;  
        $this->name = $name_in;  
    }  
    function getArtist() {  
        return $this->artist;  
    }  
    function getName() {  
        return $this->name;  
    }  
}
```

```
class SongAdapter {  
    private $song;  
    function __construct(Song $song_in) {  
        $this->song = $song_in;  
    }  
    function getArtistAndName() {
```

```
return $this->song->getName().' by '.$this->song->getArtist();
```

```
}
```

```
}
```



```
writeln('Adapter design pattern.');
```

```
$song = new Song("Pink Floyd", "Hey You");
```

```
$songAdapter = new SongAdapter($song);
```

```
writeln('Artist & name : '.$songAdapter->getArtistAndName());
```

```
writeln("");
```

```
writeln('End.');
```

```
function writeln($line_in) {
```

```
    echo $line_in."<br/>";
```

```
}
```

```
?>
```

1.2. 'Singleton' Design Pattern

The 'Singleton' design pattern gives the assurance that a class will be used only once. This design is useful for key attributes and index attributes.

Below is an example of how PHP can be used with a singleton design pattern.

```
<?php
class ID
{
    private static $a;

    public static function singleton()
    {
        if (!isset(self::$a)) {
            self::$a = new __CLASS__;
        }
        return self::$a;
    }
}

$id1 = ID::singleton();
$id2 = ID::singleton();
?>
```

4

5

2. Practical Skills

To implement the knowledge learnt during the semester, a dog walking website was commenced named "Swansea's dog walking", however it was never finished. This website's aim is to build a community around not just dog lovers, but also around people who are going through a difficult chapter in their life. Animal companionship can offer such solace, and canine even more. Furthermore, exercise has been known to help depression, anxiety, and stress.

3. References

- [1] P. Salmon, *Effects of physical exercise on anxiety, depression, and sensitivity to stress: A unifying theory*. [URL] Available: <https://www.sciencedirect.com/science/article/pii/S027273589900032X>
- [2] L. Welling, L. Thomson, *PHP and MySQL Web Development*, 2nd ed. New Jersey: Hoboken, 2017.

4. Appendix

Database.php

```
<?php
namespace App\Classes\Model;

class Database
{
    public $databaseHandler;
    public $host = "localhost";
    public $dbname = "dog_walking";
    public $user = "root"
    public $password = "";

    public function __construct()
    {
        try {
            $this->databaseHandler = new PDO("mysql:host=$this->host;dbname=$this-
>dbname", $this->user, $this->password);
            echo "Database connected";
        }
        catch(PDOException $error) {
            echo $error->getMessage();
        }
    }
}
?>
```

DogWalking.php

```
<?php
namespacea ..\Classes\DogWalking;
use ..\Classes\Model\Database;
use PDO,PDOException;

class Post extends Database
{
    private $postId;
    private $userId;
    private $address;
    private $dogName;
    private $description;
    private $image;

    public function setData($arrangeData){
        if(array_key_exists('postId',$arrangeData))
        {
            $this->postId = $arrangeData('postId');
        }
        if(array_key_exists('userId',$arrangeData))
        {
            $this->userId = $arrangeData('userId');
        }
        if(array_key_exists('address',$arrangeData))
        {
            $this->address = $arrangeData('address');
```

```

    }
    if(array_key_exists('dogName',$arrangeData))
    {
        $this->dogName = $arrangeData('dogName');
    }
    if(array_key_exists('description',$arrangeData))
    {
        $this->postId = $arrangeData('description');
    }
    if(array_key_exists('image',$arrangeData))
    {
        $this->image = $arrangeData('image');
    }
}

public function index()
{
    $sql = "SELECT * from dog_walking";
    $statementHandler = $this->databaseHandler->query($sql);
    $statementHandler->setFetchMode(PDO::FETCH_OBJ);
    return $statementHandler->getch();
}

public function viewSingleData($postId){
    $sql = "SELECT * FROM dog_walking WHERE postId = ".$postId;
    $statementHandler = $this->databaseHandler->query($sql);
    $statementHandler->setFetchMode(PDO::FETCH_OBJ);
    return $statementHandler->fetch();
}

}

?>

```

Data-input.php

```
<?php
```

```
$servername = "localhost";
```

```
$username = "root";
```

```
$password = "";
```

```
$dbname = "dog_walking";
```

```
$conn = mysqli_connect($servername, $username, $password, $dbname);
```

```
if (!$conn) {
```

```
    die("Connection failed: " . mysqli_connect_error());
```

```
}
```

```
$sql = "INSERT INTO post (address, dogName, description, image)
```

```
VALUES ('12 Sarlou Court, SA2 0LW', 'Piki', 'Piki is a white male poodle who is very friendly and energetic! Please call 077000000000 if you want to take him for a walk!',  
'../DogWalking/Resources/Images/piki.jpg')";
```

```
$sql = "INSERT INTO post (address, dogName, description, image)
```

```
VALUES ('8 Uplands Terrace, SA2 5JZ', 'Sam', 'Sam is a 2 year old Husky. He is very energetic so be careful when taking him out! Please call 077000000001 if you want to take him for a walk!',  
'../DogWalking/Resources/Images/sam.jpg')";
```

```
$sql = "INSERT INTO post (address, dogName, description, image)
```

```
VALUES ('25 Derwen Fawr road, SA2 8DP', 'Kujo', 'Kujo is a very friendly black lab. Please call 077000000003 if you want to take him for a walk!', '../DogWalking/Resources/Images/kujo.jpg')";
```

```
$sql = "INSERT INTO post (address, dogName, description, image)
```

```
VALUES ('42 St Christopher's Ct, SA1 1UA', 'Princess', 'Princess is a small scared pomeranian! Please call 077000000005 to take her for a walk!', '../DogWalking/Resources/Images/princess.jpg')";
```

```
if (mysqli_query($conn, $sql)) {  
    echo "New post created successfully";  
} else {  
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);  
}  
  
mysqli_close($conn);  
?>
```

Post.php

```
<?php include'header.php';?>

<div class="inside-banner">
    <div class="container">
        <span class="pull-right"><a href="#">Home</a>Dogs</span>
        <h2>Dogs</h2>
    </div>
</div>

<div class="container">
<div class="post spacer">

<div class="row">

<div class="col-lg-10 col-sm-8">
    <?php
    require_once('../vendor/autoload.php');
    use ..\Classes\DogWalking\DogWalking;
    $dog_walking = new DogWalking();
    $singleData = $real_estate->viewSingleData($_GET['postId']);
    ?>

    <h2>Meet your new friend!</h2>
<div> class="row">
    <div class="col-lg-8">
    <div class="images">
```

Index.php

```
<?php include'header.php';?>
```

```
<div class="inside-banner">
```

```
    <div class="container">
```

```
        <span class=pull-right"><a href="index.php">Home</a>Swansea Dog Walking</span>
```

```
        <h2> Swansea Dog Walking!</h2>
```

```
    </div>
```

```
</div>
```

```
<div class="container">
```

```
<div class="post spacer">
```

```
<div class="row">
```

```
<div class=col-lg-3 col-sm-4 ">
```

```
    <div class="search-form">
```

```
        <h4><span class="glyphicon glyphicon-search"></span> Search for</h4>
```

```
        <form action="searchResult.php" method="get" class="form form-group">
```

```
            <input type="text" class="form-control" name="search"
```

```
placeholder="Search Dogs">
```

```
            <button class="btn btn-primary">Find Now</button>
```

```
        </form>
```

```
    </div>
```

```
</div>
```

```
<div class="col-lg-9 col-sm-8">
```

```
<div class="sortBy clearfix">
```

```
<div class="pull-left result">Showing: 4 of 4 </div>
```

```

</div>
<div class="row">
    <?php
    require_once ('../Vendor/autoload.php');
    use ..\Classes\DogWalking\DogWalking;
    $dog_walking = new DogWalking();
    $allData = $dog_walking->index();

    foreach ($allData as $oneData)
    {
        $image = explode(",", $oneData->images);

        ?>

        <div class="col-lg-4 col-sm-6">
            <div class="properties">
                <div class="image-holder"></div>
                <h4><a href="post.php?postId=<?php echo $oneData->postId;?>"><?php echo
                $oneData->name;?></a></h4>
                <a class="btn btn-primary" href="post.php?postId=<?php echo $oneData-
                >postId;?>">View Details</a>
            </div>
        </div>
    }
    ?>

```

FINAL GRADE

20/100

GENERAL COMMENTS

Instructor

Knowledge & Understanding (50) 8

Quality (20) 5

Login Observer (10) 0

Database Handling (15) 7

Software Testing (5) 0

The discussion of the Adaptor and Singleton patterns is very brief and the examples don't really explain the purpose of the patterns.

No working coded examples provided.

Add fixed dog walking data to database.

View all and view single item provided.

Inserts fixed values?

No delete or edit.

No login.

No observer pattern.

Limited database abstraction, some database handling code added to a Post class.

No evidence of testing.

Code is incomplete with a number of empty files.

PAGE 1

PAGE 2



Comment 1

Please check the table of contents before submitting a report.

PAGE 3



Comment 2

What problem does this pattern solve?

What are the benefits and drawbacks of using it?

How does it work?

PAGE 4



Comment 3

Why is this an Adapter class?

PAGE 5



Comment 4

What problem does this pattern solve?

What are the benefits and drawbacks of using it?

How does it work?



Comment 5

This is not a very good example.

Why would you use this class?

PAGE 6

PAGE 7

PAGE 8
