UNIVERSIDAD DE DAGUPAN

SCHOOL OF INFORMATION TECHNOLOGY EDUCATION

ITP03 | OBJECT-ORIENTED PROGRAMMING

WEEK2 | LABORATORY ACTIVITY 2

Name : Josh A. Barrientos
Year, Course & blk : 21 – ITE - 05
Subject : OOP
Date : 12/09/2024

SCORE:

WEEK2 | Laboratory Activity 2: Comprehensive Application of Access Modifiers, Constructors, Destructors, and Method Overloading

```
<?php
class Book {
  public $title;
  protected $author;
  private $price;

  public function __construct($title, $author, $price) {
    $this->title = $title;
    $this->author = $author;
    $this->price = $price;
  }

  public function getDetails() {

    return "Title: $this->title, Author: $this->author, Price: $" . number_format($this->price, 2);
  }

  protected function setPrice($price) {
    $this->price = $price;
  }

  public function __call($method, $arguments) {

    if ($method === 'updateStock') {
      echo "Stock updated for '$this->title' with arguments: " . implode(', ', $arguments) . "\n";
    } elseif ($method === 'setPrice') {
      $this->setPrice($arguments[0]);
    } else {
      throw new BadMethodCallException("Method '$method' does not exist.");
    }
  }
}

class Library {
  private $books = [];
  public $name;

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

  public function addBook(Book $book) {
    $this->books[] = $book;
```

```
   }

   public function removeBook($title) {

      foreach ($this->books as $index => $book) {
         if ($book->title === $title) {
            unset($this->books[$index]);
            echo "Book '$title' removed from the library.\n";
            return;
         }
      }
      echo "Book '$title' not found in the library.\n";
   }

   public function listBooks() {
      echo "Books in the library:\n";
      foreach ($this->books as $book) {
         echo $book->getDetails() . "\n";
      }
   }

   public function __destruct() {
      $this->books = [];
      echo "The Library '$this->name' is now closed.\n";
   }
}
$book1 = new Book('The Great Gatsby', 'F. Scott Fitzgerald', 12.99);
$book2 = new Book('1984', 'George Orwell', 8.99);
$library = new Library('City Library');

$library->addBook($book1);
$library->addBook($book2);

$book1->setPrice(12.99);
$book1->updateStock(50);

$library->listBooks();
$library->removeBook('1984');

echo "Books in the library after removal:\n";
$library->listBooks();

unset($library);
?>
```

EXPLANATION

Constructors:

The __construct method in both classes initializes the objects with specific values.

```
public function __construct($title, $author, $price) {
   $this->title = $title;
   $this->author = $author;
   $this->price = $price;
}
```

Methods:

The getDetails method in the Book class returns a formatted string containing the book's details. This method demonstrates how to expose certain functionalities while keeping the internal state hidden.

The setPrice method is protected, allowing price updates only within the class or its subclasses, maintaining control over how the price can be modified.

Magic Methods:

The __call magic method allows handling calls to methods that are not defined in the class. This is useful for dynamic method handling, such as updating stock or setting the price:

```
public function __call($method, $arguments) {
  if ($method === 'updateStock') {
    // Handle stock update
  } elseif ($method === 'setPrice') {
    $this->setPrice($arguments[0]);
  } else {
    throw new BadMethodCallException("Method '$method' does not exist.");
  }
}
```

Composition:

The Library class contains an array of Book objects, demonstrating composition. This allows the Library to manage multiple books, providing methods to add, remove, and list books:

```
public function addBook(Book $book) {
  $this->books[] = $book;
}
```

Destructors:

The __destruct method in the Library class is called when an object is destroyed. It cleans up the $books array and outputs a message indicating that the library is closed:

```
public function __destruct() {
  $this->books = [];
  echo "The Library '$this->name' is now closed.\n";
}
```