Домашняя работа

по предмету: Основы PHP (семинары)

Выполнил: Байбородин Александр

Группа: Программист | 17.04.2024

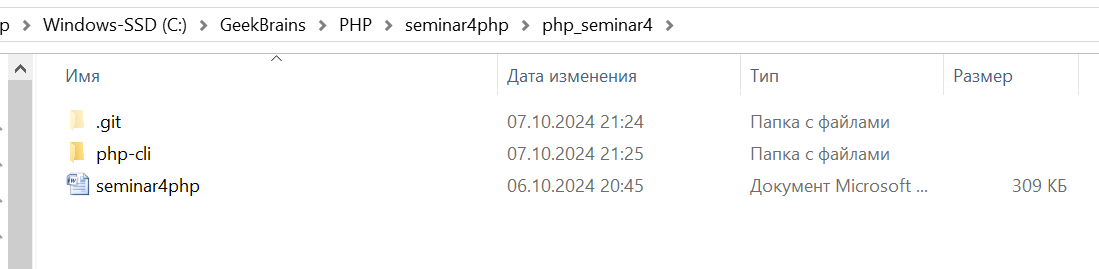
### Урок 4. ООП

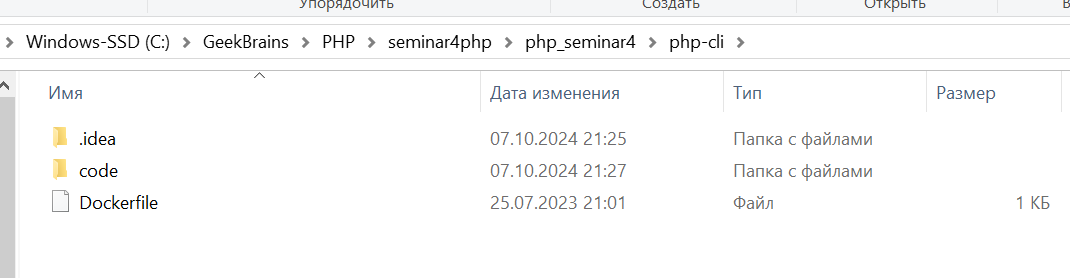
Задание

**1**. Придумайте класс, который описывает любую сущность из предметной области библиотеки: книга, шкаф, комната и т.п.  
  
**2**. Опишите свойства классов из п.1 (состояние).  
  
**3**. Опишите поведение классов из п.1 (методы).  
  
**4**. Придумайте наследников классов из п.1. Чем они будут отличаться?  
  
**5**. Создайте структуру классов ведения книжной номенклатуры.  
— Есть абстрактная книга.  
— Есть цифровая книга, бумажная книга.  
— У каждой книги есть метод получения на руки.  
  
У цифровой книги надо вернуть ссылку на скачивание, а у физической – адрес библиотеки, где ее можно получить. У всех книг формируется в конечном итоге статистика по кол-ву прочтений.  
Что можно вынести в абстрактный класс, а что надо унаследовать?  
  
**6**. Дан код:  
  
class A {  
public function foo() {  
static $x = 0;  
echo ++$x;  
}  
}  
$a1 = new A();  
$a2 = new A();  
$a1->foo();  
$a2->foo();  
$a1->foo();  
$a2->foo();  
  
Что он выведет на каждом шаге? Почему?  
  
Немного изменим п.5  
  
class A {  
public function foo() {  
static $x = 0;  
echo ++$x;  
}  
}  
class B extends A {  
}  
$a1 = new A();  
$b1 = new B();  
$a1->foo();  
$b1->foo();  
$a1->foo();  
$b1->foo();  
  
Что он выведет теперь?

Решение:

Сформируем структуру каталогов и файлов проекта:





Сформируем докер-файл, для сборки и запуска контейнера:

FROM php:8.2

VOLUME /code

RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer

ENV COMPOSER\_ALLOW\_SUPERUSER 1

WORKDIR /code



1. Реализуем класс, который описывает сущность из предметной области библиотеки: книга. Опишим свойства классов из п.1 (состояние). Опишим поведение классов из п.1 (методы). Придумаем наследников класса. Создадим структуру классов ведения книжной номенклатуры.  
   — Есть абстрактная книга.

— Есть цифровая книга, бумажная книга.

— У каждой книги есть метод получения на руки.

- У цифровой книги надо вернуть ссылку на скачивание, а у физической – адрес библиотеки, где ее можно получить. У всех книг формируется в конечном итоге статистика по кол-ву прочтений.

<?php

abstract class AbstractBook {

    protected $title;

    protected $author;

    protected $readCount = 0;

    public function \_\_construct($title, $author) {

        $this->title = $title;

        $this->author = $author;

    }

    public function getReadCount() {

        return $this->readCount;

    }

    abstract public function getDetails();

    abstract public function getAccess();

}

class EBook extends AbstractBook {

    private $downloadLink;

    public function \_\_construct($title, $author, $downloadLink) {

        parent::\_\_construct($title, $author);

        $this->downloadLink = $downloadLink;

    }

    public function getDetails() {

        echo "Название: {$this->title}, Автор: {$this->author}\n";

    }

    public function getAccess() {

        $this->readCount++;

        echo "Ссылка для скачивания: {$this->downloadLink}\n";

    }

}

class PhysicalBook extends AbstractBook {

    private $libraryAddress;

    public function \_\_construct($title, $author, $libraryAddress) {

        parent::\_\_construct($title, $author);

        $this->libraryAddress = $libraryAddress;

    }

    public function getDetails() {

        echo "Название: {$this->title}, Автор: {$this->author}\n";

    }

    public function getAccess() {

        $this->readCount++;

        echo "Адрес библиотеки: {$this->libraryAddress}\n";

    }

}

// Пример использования:

$ebook = new EBook('1984', 'Джордж Оруэлл', 'https://royallib.com/get/doc/oruell\_dgordg/1984.zip');

$ebook->getDetails();

$ebook->getAccess();

echo "Количество прочтений электронной книги: " . $ebook->getReadCount() . "\n";

echo "\n"; // Разделим вывод для удобства

$physicalBook = new PhysicalBook('Война и мир', 'Лев Толстой', 'г. Москва, ул. Пушкина, д. 10');

$physicalBook->getDetails();

$physicalBook->getAccess();

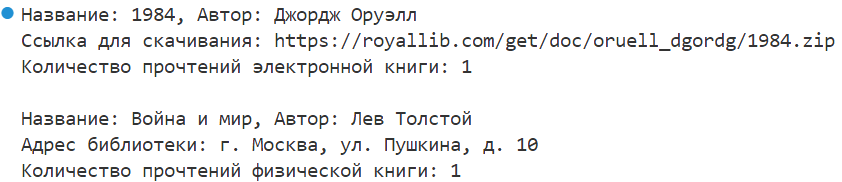
echo "Количество прочтений физической книги: " . $physicalBook->getReadCount() . "\n";

Соберем контейнер из докер-файла командой в каталоге php-cli:

docker build -t php-cli-app .

Запустим контейнер командой и проверим работу созданных классов:





Теперь реализуем в файле task4.php вывод кода подзадания 4:

<?php

class A {

    public function foo() {

        static $x = 0;

        echo ++$x;

    }

}

$a1 = new A();

$a2 = new A();

$a1->foo(); // выводит 1

$a2->foo(); // выводит 2

$a1->foo(); // выводит 3

$a2->foo(); // выводит 4

Проверяем работу файла через docker:





Теперь поменяем код как это требуется по условию и проверим результат через docker:

class A {

    public function foo() {

        static $x = 0;

        echo ++$x;

    }

}

class B extends A {

}

$a1 = new A();

$b1 = new B();

$a1->foo(); // выводит 1

$b1->foo(); // выводит 2

$a1->foo(); // выводит 3

$b1->foo(); // выводит 4





Видим, что вывод работает и выводит 1234.