1. Реалізувати та запустити код з лекції.

Скріншоти виконання коду в папках на GitHub (Image Action, Image ClassLoader, Image HomeController)

2. Вибрати предметну область для подальшої розробки.

Предметна область: магазин продовольчих товарів

3. Реалізувати модель MVC з необхідними системними конфігураціями.

Скріншоти виконання коду у папках на GitHub (Image\_MVC)

4. Під’єднати проєкт до системи контролю версій (GitHub, GitLab, Bitbucket тощо).

Посилання:

5. Описати поняття поліморфізму, різницю між агрегацією та композицією, а також інші принципи ООП у документі.

Парадигма поліморфізму – це взаємозамінність об'єктів з однаковим інтерфейсом. У PHP поліморфізм означає, що у коді програми можна використовувати замість екземплярів батьківського класу екземпляри підкласу. приклад.

Нехай є абстрактний клас

abstract class Product

{//визначаємо правило, що це товари повинні продаватися, тобто. мати метод do\_sales()

abstract public function do\_sales();

}

class Supermarket extends Product

{// перевизначаємо абстрактний метод продажу

public function do\_sales ()

{echo '<h4>B2C</h4>';}

}

class Kiosk extends Product

{// перевизначаємо абстрактний метод продажу

public function do\_sales()

{echo '<h4>покупець особисто контактує з продавцем</h4>';}

}

//Наповнюємо масив продажів об'єктами, похідними від Товар.

$Products[] = new Supermarket();

$Products[] = new Kiosk();

foreach ($Products as $Product)

{if ($Product instanceof Product) {

$Product->do\_sales(); //виводимо дані на друк

}

else { // обробка помилки }

}

Агрегація - зв'язок між класами, коли об'єкт одного класу містить інший об'єкт як свою частину. Наприклад, кожен замовлення містить набір товарів, тобто. об'єкт класу Product є частиною об'єкта класу Order.

Композиція — це зв'язок між класами, коли об'єкт одного класу є частиною об'єкта іншого класу і не може існувати окремо від нього.

Різниця між агрегацією та композицією в тому, що коли об'єкт-ціле композиції видаляється, всі його об'єкти-частини також видаляються.