МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

КАФЕДРА «ИИТ»

Лабораторная работа №7

По дисциплине «Проектирование интернет систем»

Выполнил: студент ФЭИС

группы ПО-2

Шелест П.Ю.

Проверила: Слинко Е.В.

Брест 2020

**Цель:** Познакомиться с практической реализацией принципа инверсии зависимостей, а также протоколами межсервисного взаимодействия.

**Задание:**

Реализуйте механизмы хранения (persistence) для ранее разработанного приложения – технологию, фреймворк, ORM выберите сами. Это могут быть реляционная БД, NoSQL, InMemory, файловая система, Redis и т.д. Реализуйте минимум два механизма доставки – способов вызова функций вашего приложения. Например, REST и командная строка. Отдельно реализуйте микросервис, который не будет выполнять никакой полезной работы, кроме вызова какой-либо функции вашего приложения по протоколу gRPC (третий способ доставки для вашего приложения), используйте Google Protobuf. Большим плюсом станет, если данный микросервис будет реализован не на PHP.

**Вариант**: *Продажа электрооборудования*

**1)Работа с электрооборудованием с помощью protobuf**

***ProtoProduct.proto***

syntax = "proto3";

message ProtoProduct

{

string id = 1;

string name = 2;

string price = 3;

string inform = 4;

string image = 5;

}

***ProtobufController***

Функция getProducts применяется для получения списка всех роботов и его вывод. Функция test применяется для создание объекта ProtoProduct, его отображения, изменения его полей(цена, изображение) и вывода обновленных данных.

<?php

include\_once './vendor/autoload.php';

include\_once 'GPBMetadata/ProtoProduct.php';

include\_once 'Entities/Products.php';

include\_once 'Services/ProductsUseCase.php';

include\_once 'Repositories/ClientMySQLRepository.php';

include\_once 'Repositories/ProductsMySqlRepository.php';

include\_once 'ProtoProduct.php';

class ProtobufController

{

function test()

{

echo "<br><br>";

$product = new ProtoProduct();

$product->setId("900");

$product->setName("тестовое имя");

$product->setPrice("900");

$product->setInform("тестовое описание");

$product->setImage("тестовое изображение");

$str=$product->serializeToString();

$product2 = new ProtoProduct();

try

{

$product2->mergeFromString($str);

}

catch (Exception $e)

{

echo "произошла ошибка merge";

}

echo $product2->serializeToJsonString();

$product2->setPrice('111');

$product2->setImage('новое тестовое изображение');

echo "<br><br><br>";

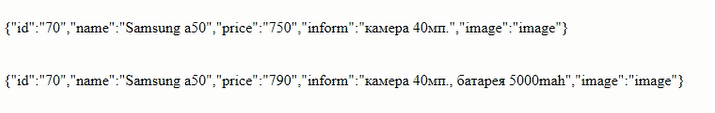
echo $product2->serializeToJsonString();

}

}

?>

**Результат:**



**2)Вывод в командную строку всего электрооборудования**

**Текст программы:**

***JsonController***

<?php

include\_once 'Repositories/ClientMySQLRepository.php';

include\_once 'Repositories/ProductsMySqlRepository.php';

include\_once 'Services/ProductsUseCase.php';

class JsonController

{

public function showProducts()

{

$client\_rep = new ClientMySQLRepository();

$product\_rep = new ProductsMySQLRepository();

$product\_usecase = new ProductsUseCase($client\_rep, $product\_rep);

$products = $product\_usecase->getProducts();

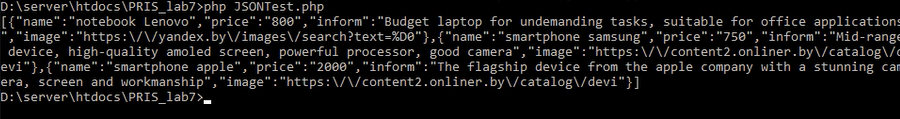
echo json\_encode($products);

}

}

?>

**Результат:**

****

**Выводы:**

В результате выполнения работы я ознакомился с практической реализацией принципа инверсии зависимостей и протоколами межсервисного взаимодействия.