Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №7

По дисциплине: «ПИС»

Тема: “Инфраструктура: хранение и

доставка. Межсервисное

взаимодействие”

Выполнил:

Студент 3 курса

Группы ПО-7

Комиссаров А.Е.

Проверил:

Михняев А.Л.

2023

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



**Ход работы.**

**1)Работа с товарами с помощью protobuf**

***ProtoItem.proto***

syntax = "proto3";

message ProtoItem

{

string id = 1;

string name = 2;

string price = 3;

string inform = 4;

string image = 5;

}

***ProtobufController***

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

<?php

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

include\_once 'GPBMetadata/ProtoItem.php';

include\_once 'Entities/Items.php';

include\_once 'Services/ItemsUseCase.php';

include\_once 'Repositories/UserMySQLRepository.php';

include\_once 'Repositories/ItemsMySqlRepository.php';

include\_once 'ProtoItem.php';

class ProtobufController

{

function test()

{

echo "<br><br>";

$Item = new ProtoItem();

$Item->setId("900");

$Item->setName("тестовое имя");

$Item->setPrice("900");

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

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

$str=$Item->serializeToString();

$Item2 = new ProtoItem();

try

{

$Item2->mergeFromString($str);

}

catch (Exception $e)

{

echo "произошла ошибка merge";

}

echo $Item2->serializeToJsonString();

$Item2->setPrice('111');

$Item2->setImage('новое тестовое изображение');

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

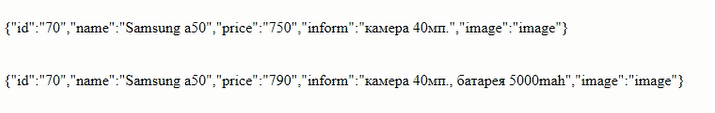
echo $Item2->serializeToJsonString();

}

}

?>

**Результат:**



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

**Текст программы:**

***JsonController***

<?php

include\_once 'Repositories/UserMySQLRepository.php';

include\_once 'Repositories/ItemsMySqlRepository.php';

include\_once 'Services/ItemsUseCase.php';

class JsonController

{

public function showItems()

{

$User\_rep = new UserMySQLRepository();

$Item\_rep = new ItemsMySQLRepository();

$Item\_usecase = new ItemsUseCase($User\_rep, $Item\_rep);

$Items = $Item\_usecase->getItems();

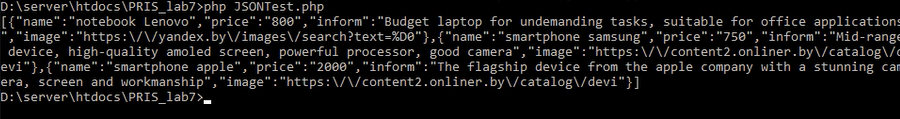
echo json\_encode($Items);

}

}

?>

**Результат:**

****

**Вывод:** я познакомился с практической реализацией принципа инверсии зависимостей, а также протоколами межсервисного взаимодействия.