

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Отчет по лабораторной работе №6
Дисциплина “ПИС”

Выполнила:
студентка, гр. ПО-3
Гаврилюк Р. И.

Проверил:
Лаврущик А.

Брест 2021

Лабораторная работа №6

Сервисы

Вариант 21

Цель работы: познакомиться с практической реализацией принципа инверсии зависимостей, принципа единственной ответственности и механизмом внедрения зависимостей.

Задание для выполнения: реализуйте сервисы на уровне приложения, реализующие с доменными моделями те же действия, что и сценарий транзакции из ЛР №1. Не забудьте о принципе единственной ответственности.

Хотя бы одно действие должно реализовываться доменным сервисом. Напишите модульные тесты для сервисов.

Поскольку инфраструктурный слой еще не реализован – используйте заглушки вместо репозиториев (ЛР №4).

Предметная область: продажа игрушек.

Программная реализация:

IndexController.php

```
<?php

namespace App\Controller;

use App\Entity\ShopCart;
use App\Entity\ShopItems;
use App\Entity\ShopOrder;
use App\Form\OrderFormType;
use App\Repository\ShopCartRepository;
use App\Repository\ShopItemsRepository;
use Doctrine\ORM\EntityManagerInterface;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\HttpFoundation\Session\SessionInterface;
use Symfony\Component\Routing\Annotation\Route;

class IndexController extends AbstractController
{
    private SessionInterface $session;

    /**
     * IndexController constructor.
     * @param SessionInterface $session
     */
    public function __construct(SessionInterface $session)
```

```
{  
    $this->session = $session;  
    $this->session->start();  
}  
  
/**  
 * @Route("/", name="index")  
 */  
public function index(): Response  
{  
    return $this->render(  
        'index/index.html.twig',  
        [  
            'title' => 'Главная страница',  
        ]  
    );  
}  
  
/**  
 * @Route("/shop/list", name="shopList")  
 * @param ShopItemsRepository $itemsRepository  
 * @return Response  
 */  
public function shopList(ShopItemsRepository $itemsRepository): Response  
{  
    $items = $itemsRepository->findAll();  
  
    return $this->render(  
        'index/shopList.html.twig',  
        [  
            'title' => 'SHOP LIST',  
            'items' => $items,  
        ]  
    );  
}  
  
/**  
 * @Route("/shop/cart/add/{id<\d+>}", name="shopCartAdd")  
 * @param ShopItems $shopItems  
 * @param EntityManagerInterface $em  
 * @return Response  
 */  
public function shopCartAdd(ShopItems $shopItems, EntityManagerInterface $em): Response  
{  
    $sessionId = $this->session->getId();  
  
    $shopCart = (new ShoppingCart())  
        ->setShopItem($shopItems)  
        ->setCount(1)  
        ->setSessionId($sessionId);
```

```

        $em->persist($shopCart);
        $em->flush();

        return $this->redirectToRoute('shopItem', ['id' => $shopItems->getId()]);
    }

    /**
     * @Route("/shop/item/{id<\d+>}", name="shopItem")
     *
     * @param ShopItems $shopItems
     * @return Response
     */
    public function shopItem(ShopItems $shopItems): Response
    {
        return $this->render(
            'index/shopItem.html.twig',
            [
                'title' => $shopItems->getTitle(),
                'description' => $shopItems->getDescription(),
                'price' => $shopItems->getPrice(),
                'id' => $shopItems->getId(),
            ]
        );
    }

    /**
     * @Route("/shop/cart", name="shopCart")
     *
     * @param ShoppingCartRepository $cartRepository
     * @return Response
     */
    public function shopCart(ShoppingCartRepository $cartRepository): Response
    {
        $session = $this->session->getId();
        $items = $cartRepository->findBy(['sessionId' => $session]);

        return $this->render(
            'index/shopCart.html.twig',
            [
                'title' => 'Корзина',
                'items' => $items,
            ]
        );
    }

    /**
     * @Route("/shop/order", name="shopOrder")
     *
     * @param Request $request
     * @return Response
     */
    public function shopOrder(Request $request, EntityManagerInterface $em): Response
    {

```

```

// just setup a fresh $task object (remove the example data)
$shopOrder = new ShopOrder();

$form = $this->createForm(OrderFormType::class, $shopOrder);

$form->handleRequest($request);
if ($form->isSubmitted() && $form->isValid()) {
    $shopOrder = $form->getData();

    if ($shopOrder instanceof ShopOrder) {
        $sessionId = $this->session->getId();
        $shopOrder->setStatus(ShopOrder::STATUS_NEW_ORDER);
        $shopOrder->setSessionId($sessionId);
        $em->persist($shopOrder);
        $em->flush();
        //session_regenerate_id
        $this->session->migrate();
    }

    return $this->redirectToRoute('index');
}

return $this->render(
    'index/shopOrder.html.twig',
    [
        'title' => 'Оформление заказа',
        'form' => $form->createView(),
    ]
);
}
}

```

ServiceEntityRepository.php

```

<?php

namespace Doctrine\Bundle\DoctrineBundle\Repository;

use Doctrine\ORM\EntityRepository;
use Doctrine\Persistence\ManagerRegistry;
use LogicException;

class ServiceEntityRepository extends EntityRepository implements ServiceEntityRepositoryInterface
{
    /**
     * @param string $entityClass The class name of the entity this repository manages
     */
    public function __construct(ManagerRegistry $registry, $entityClass)
    {
        $manager = $registry->getManagerForClass($entityClass);

```

```
if ($manager === null) {
    throw new LogicException(sprintf(
        'Could not find the entity manager for class "%s". Check your Doctrine configuration to make sure it is configured to load this entity\'s metadata.',
        $entityClass
    ));
}

parent::__construct($manager, $manager->getClassMetadata($entityClass));
}
```

Вывод: познакомилась с практической реализацией принципа инверсии зависимостей, принципа единственной ответственности и механизмом внедрения зависимостей.