

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

Отчет
По лабораторной работе №6
по дисциплине «ПриС»

Выполнила
студентка 3 курса
группы ПО-3
Пивчик В.Г.
Проверил
Лаврущик А.И.

Брест 2021

Лабораторная работа №5

Сервисы

ВАРИАНТ 9

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 No1. 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 репозитория (ЛР4).

Предметная область. Продажа электрооборудования.

Ход работы

Для того, чтобы продемонстрировать работу сервиса, будем использовать рабочий контроллер (ReservationController).

Программная реализация

ReservationService.php

```
<?php
namespace App\Service;
use App\Entity\Purchase;
use App\Repository\PurchaseRepository;

class ReservationService {
    private $purchaseRepository;
    public function __construct(PurchaseRepository $purchaseRepository) {
        $this->purchaseRepository=$purchaseRepository;
    }

    public function getAllPurchase() {
        return $this->purchaseRepository->findAll();
    }

    public function addPurchase(Purchase $purchase) {
        $this->purchaseRepository->add($purchase);
    }

    public function deletePurchaseById($id) {
        $this->purchaseRepository->delete($id);
    }
}
```

ReservationController.php

```
<?php
namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;
use App\Service\ReservationService;
use Symfony\Component\HttpFoundation\Request;
use App\Entity\Purchase;
use App\Form\PurchaseType;

class ReservationController extends AbstractController {
    private $reservationService;

    public function __construct(ReservationService $reservationService) {
        $this->reservationService=$reservationService;
    }

    /**
     * @Route("/purchase", name="purchase")
     */
    public function getAllPurchases() {
        return $this->render('reservation/purchase.html.twig', [
            'purchase' => $this->reservationService->getAllPurchase(),
        ]);
    }

    /**
     * @Route("/purchase/delete/{id}", name="deletePurchaseById")
     */
    public function deletePurchaseById($id) {
        $this->reservationService->deletePurchaseById($id);

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

    /**
     * @Route("/purchase/create", name="purchase")
     */
    public function createPurchase(Request $request) {
        $purchase = new Purchase();
        $form = $this->createForm(PurchaseType::class, $purchase);
        $form->handleRequest($request);

        if ($form->isSubmitted() && $form->isValid()) {
            $this->reservationService->addPurchase($purchase);
            return $this->redirectToRoute('purchase');
        }

        return $this->render('reservation/purchase_form.html.twig', [
            'form' => $form->createView(),
        ]);
    }
}
```

Результат работы:

Создание

Просмотр

Purchase name:

Cost:

Version:

ID	Name	Cost	Version
1	Phone	2000	1
4	iPad	50	1
5	MacBook	50000	16

Вывод

В данной лабораторной работе я познакомилась с практической реализацией принципа инверсии зависимостей, принципа единственной ответственности и механизмом внедрения зависимостей.