Министерство образования Республики Беларусь
Учреждение образования
“Брестский государственный технический университет”
Кафедра ИИТ

Отчёт
По лабораторной работе №6
По дисциплине ПрИС

Выполнил
Студент группы ПО-3
3-го курса
Кулинкович И. Т.

Проверил
Лаврущик А. И.

Брест 2021

Лабораторная работа №6

Сервисы

ВАРИАНТ 17

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 репозиториев (ЛР №4).

Предметная область. Управление фитнес-центром.

Ход работы

Для демонстрации работы сервиса нам потребуется рабочий контроллер.

Код программы

BookingController.php

```php
<?php
namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class BookingController extends AbstractController {
    private $bookingService;

    public function __construct(BookingService $bookingService) {
        $this->bookingService=$bookingService;
    }

    /**
     * @Route("/visitors", name="visitors")
     */
    public function getAllVisitors() {
        return $this->render('booking/visitors.html.twig', [
            'visitors' => $this->bookingService->getAllVisitors(),
        ]);
    }

    /**
     * @Route("/visitors/delete/{id}", name="deleteVisitorById")
     */
    public function deleteVisitorById($id) {
        $this->bookingService->deleteVisitorById($id);

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

    /**
     * @Route("/visitors/create", name="visitor")
     */
    public function createVisitor(Request $request) {
        $visitor = new Visitor();
```

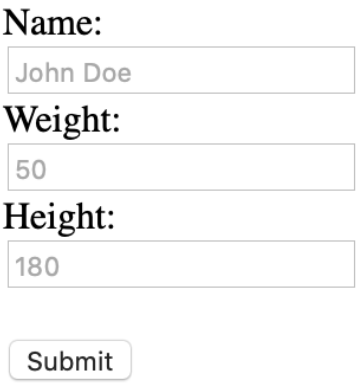
```
        $visitor = $this->createForm(VisitorType::class, $visitor);
        $form->handleRequest($request);

        if ($form->isSubmitted() && $form->isValid()) {
            $this->bookingService->addVisitor($visitor);
            return $this->redirectToRoute('visitors');
        }

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

BookingService.php

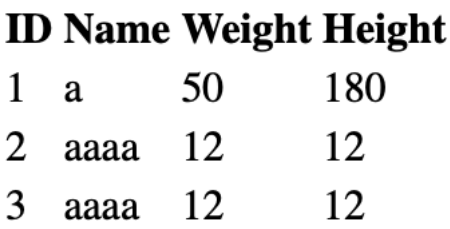
```
<?php
namespace App\Service;
use App\Entity\Visitor;
use App\Repository\VisitorRepository;

class BookingService {
    private $visitorRepository;
    public function __construct(VisitorRepository $visitorRepository) {
        $this->visitorRepository=$visitorRepository;
    }

    public function getAllVisitors() {
        return $this->visitorRepository->findAll();
    }

    public function addVisitor(Visitor $visitor) {
        $this->visitorRepository->add($visitor);
    }

    public function deleteVisitorById($id) {
        $this->visitorRepository->delete($id);
    }
}
```

Демонстрация работы

Форма создания



Просмотр



ID	Name	Weight	Height
1	a	50	180
2	aaaa	12	12
3	aaaa	12	12

Вывод

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