

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
Гаврилкович Е.В.
Проверил
Лаврущик А.И.

Брест 2021

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. Промышленная робототехника.

Ход работы

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

Программная реализация

ReservationService.php

```
<?php
namespace App\Service; use
App\Entity\Robot;
use App\Repository\RobotRepository;

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

    public function getAllRobot() {
        return $this->robotRepository->findAll();
    }

    public function addRobot(Robot $robot) {
        $this->robotRepository->add($robot);
    }

    public function deleteRobotById($id) {
        $this->robotRepository->delete($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\Robot;
use App\Form\RobotType;

class ReservationController extends AbstractController {private
    $reservationService;

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

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

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

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

    /**
     * @Route("/robot/create", name="robot")
     */
    public function createRobot(Request $request) {
        $robot = new Robot();
        $form = $this->createForm(RobotType::class, $robot);
        $form->handleRequest($request);

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

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

Результат работы:

Создание

Просмотр

Robot name:

Cost:

Version:

Submit

ID	Name	Cost	Version
1	ad	33	1
2	ad	33	1
3	Robot-php7	1000	4
4	RobotZX3000	5000	3

Вывод

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