Министерство образования Республики Беларусь Учреждение образования

«Брестский государственный университет» Кафедра ИИТ

# Лабораторная работа №7 По дисциплине ПИС

Выполнил:

студент группы ПО-4

Яковчик И.А

Проверил: Михняев А.Л.

Брест 2022

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

**Предметная область:** блог о фермерском хозяйстве.

**Задание для выполнения:** реализуйте сервисы на уровне приложения, производящие с доменными моделями те же действия, что и сценарий транзакции из ЛР №1. Хотя бы одно действие должно реализовываться доменным сервисом.

# Ход работы

**Агрегаты: Сервисы:**

* Note - NoteService
* User

# Репозитории:

* + NoteRepository
  + UserRepository
  + SpecializationRepository
  + CategoryRepository
  + NoteChangeEventRepository

# NoteController:

<?php

namespace App\Controller; use App\Entity\Note;

use App\Form\NoteType;

use App\Service\NoteService;

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

use Symfony\Component\HttpFoundation\Request;

class NoteController extends AbstractController

{

private $noteService;

public function construct(NoteService $noteService)

{

$this->noteService=$noteService;

}

/\*\*

* @Route("/notes", name="notes")

\*/

public function getAllNotes()

{

return $this->render('note/notes.html.twig', [ 'notes' => $this->noteService->getAllNotes(),

]);

}

/\*\*

* @Route("/notes/delete/{id}", name="deleteNoteById")

\*/

public function deleteNoteById($id)

{

$this->noteService->deleteNoteById($id); return $this->redirectToRoute('notes');

}

/\*\*

* @Route("/notes/create", name="note")

\*/

public function createNote(Request $request)

{

$note = new Note();

$form = $this->createForm(NoteType::class, $note);

$form->handleRequest($request);

if ($form->isSubmitted() && $form->isValid()) {

$this->noteService->addNote($note); return $this->redirectToRoute('notes');

}

return $this->render('note/note\_form.html.twig', [ 'form' => $form->createView(),

]);

}

}

**NoteService:**

<?php

namespace App\Service; use App\Entity\Note;

use App\Repository\NoteRepository;

class NoteService {

private $noteRepository;

public function construct(NoteRepository $noteRepository)

{

$this->noteRepository=$noteRepository;

}

public function getAllNotes() {

return $this->noteRepository->findAll();

}

public function addNote(Note $note) {

$this->noteRepository->add($note);

}

public function deleteNoteById($id) {

$this->noteRepository->delete($id);

}

}

**Вывод**: познакомился с практической реализацией принципа инверсии зависимостей, принципа единственной ответственности и механизмом

внедрения зависимостей.