

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический
университет»
Кафедра «ЭВМ и системы»

Лабораторная работа №6
Дисциплина «ПИС»

Выполнил:
студент группы:
ПО-3
Будяков В.В.
Проверил:
Лаврущик А.И.

Брест 2021 г.

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

Постановка задачи: реализовать сервисы на уровне приложения, реализующие с доменными моделями те же действия, что и сценарий транзакции из ЛР №1.

Ход работы:

В ParkPlaceService выполняется вся бизнес логика по бронированию и освобождению парковочного места. Для взаимодействия с базой данных, ParkPlaceService использует ParkPlaceRepository.

Сервисы:

- ParkPlaceService
- UserService

```
<?php
namespace App\Service;
use App\Repository\ParkPlaceRepository;
class ParkPlaceService
{
    private $parkPlaceRepository;

    public function __construct(ParkPlaceRepository $parkPlaceRepositoryInject)
    {
        $this->parkPlaceRepository = $parkPlaceRepositoryInject;
    }

    public function takeParkingPlace($parkPlaceName)
    {
        $parkPlaceId = substr($parkPlaceName, 9);
        $SparkPlace = $this->parkPlaceRepository->findOneBy(['id' => $parkPlaceId]);
        if ($this->parkPlaceRepository->setUser($SparkPlace)) {
            return array('isSuccess' => true);
        } else {
            return array('isSuccess' => false, 'errorMessage' => "Sorry, but the parking space \""
                . $parkPlaceName . "\" is occupied");
        }
    }

    public function vacateParkingPlace($parkPlaceName)
    {
        $parkPlaceId = substr($parkPlaceName, 9);
        $SparkPlace = $this->parkPlaceRepository->findOneBy(['id' => $parkPlaceId]);
        if ($this->parkPlaceRepository->deleteUser($SparkPlace)) {
            return array('isSuccess' => true);
        } else {
            return array('isSuccess' => false, 'errorMessage' => "Sorry, but the parking space \""
                . $parkPlaceName . "\" is not yours");
        }
    }

    public function findAll()
    {
        return $this->parkPlaceRepository->findAll();
```

```
    }  
}
```

В UserService выполняется регистрация нового пользователя. Пароль шифруется с помощью UserPasswordEncoderInterface. Для взаимодействия с базой данных, UserService использует UserRepository.

```
<?php  
  
namespace App\Service;  
  
use App\Entity\User;  
use App\Repository\UserRepository;  
use Symfony\Component\Security\Core\Encoder\UserPasswordEncoderInterface;  
  
class UserService  
{  
  
    private $userRepository;  
    private $passEncoder;  
  
    public function __construct(UserRepository $userRepositoryInject, UserPasswordEncoderInterface $passEncoderInject)  
    {  
        $this->userRepository = $userRepositoryInject;  
        $this->passEncoder = $passEncoderInject;  
    }  
  
    public function registerUser(User $user): bool  
    {  
        $user->setPassword($this->passEncoder->encodePassword($user, $user->getPassword()));  
        return $this->userRepository->addUser($user);  
    }  
}
```

Выводы: познакомился с тактическими шаблонами предметно ориентированного проектирования.