

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

**Отчет  
По лабораторной работе №4  
по дисциплине «ПрИС»**

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

Брест 2021

# Лабораторная работа №4

## Модульное тестирование

### ВАРИАНТ 9

**Цель работы.** Познакомиться с механизмами модульного тестирования веб-приложений, построенных на гексагональной архитектуре

#### Задание для выполнения.

Установите и настройте PHPUnit. Напишите модульные тесты для сценария транзакции из ЛР №1. Постарайтесь добиться 100% покрытия кода тестами. При написании постарайтесь учитывать, что в дальнейшем части этого кода вам могут пригодиться при тестировании доменной модели (ЛР5) и сервисов приложения (ЛР6).

**Предметная область.** Продажа электрооборудования.

## Ход работы

В ходе данной лабораторной работы логика была переписана с использованием Symfony, были получены 3 репозитория: ClientRepository, PurchaseRepository, ShoppingSessionRepository.

### ClientTest.php - тесты модели данных клиента

```
<?php namespace App\Tests;

use PHPUnit\Framework\TestCase;
use Symfony\Bundle\FrameworkBundle\Test\KernelTestCase;

class PurchaseTest extends KernelTestCase
{
    private $entityManager;

    protected function setUp(): void
    {
        $kernel = self::bootKernel();

        $this->entityManager = $kernel->getContainer()->get('doctrine')->getManager();
    }

    public function testInsert()
    {
        $repository = $this->entityManager->getRepository(\App\Entity\Purchase::class);
        $original_count = $repository->getPurchasesCount();
        $repository->addPurchase("iPhone", 1000, 10);
        $new_count = $repository->getPurchasesCount();
        $this->assertEquals($original_count + 1, $new_count);
    }

    public function testFindAll()
    {
        $repository = $this->entityManager->getRepository(\App\Entity\Purchase::class);
        $our_count = $repository->getPurchasesCount();
    }
}
```

```
        $get_all_count = count($repository->findAll());
        $this->assertEquals($our_count, $get_all_count);
    }
}
```

## PurchaseTest.php - тесты модели данных товара

```
<?php namespace App\Tests;

use PHPUnit\Framework\TestCase;
use Symfony\Bundle\FrameworkBundle\Test\KernelTestCase;

class ClientTest extends KernelTestCase
{
    private $entityManager;

    protected function setUp(): void
    {
        $kernel = self::bootKernel();

        $this->entityManager = $kernel->getContainer()->get('doctrine')->getManager();
    }

    public function testInsert()
    {
        $repository = $this->entityManager->getRepository(App\Entity\Client::class);
        $original_count = $repository->getClientsCount();
        $repository->addClient("Lera", 50505);
        $new_count = $repository->getClientsCount();
        $this->assertEquals($original_count + 1, $new_count);
    }

    public function testFindAll()
    {
        $repository = $this->entityManager->getRepository(App\Entity\Client::class);
        $our_count = $repository->getClientsCount();
        $get_all_count = count($repository->findAll());
        $this->assertEquals($our_count, $get_all_count);
    }
}
```

## ShoppingSessionTest.php - тесты модели данных сессии покупки

```
<?php namespace App\Tests;

use PHPUnit\Framework\TestCase;
use Symfony\Bundle\FrameworkBundle\Test\KernelTestCase;

class ShoppingSessionTest extends KernelTestCase
{
    private $entityManager;

    protected function setUp(): void
    {
        $kernel = self::bootKernel();
        $this->entityManager = $kernel->getContainer()->get('doctrine')->getManager();
    }
}
```

```

}

public function testInsert()
{
    $repository = $this->entityManager->getRepository(\App\Entity\ShoppingSession::class);
    $purchase_repository = $this->entityManager->getRepository(\App\Entity\Purchase::class);
    $client_repository = $this->entityManager->getRepository(\App\Entity\Client::class);
    $new_purchase = $purchase_repository->addPurchase("test purchase", 10, 10);
    $new_client = $client_repository->addClient("test client", 11111);
    $original_count = $repository->getShoppingSessionCount();
    $repository->addShoppingSession(new \DateTime(), $new_purchase, $new_client);
    $new_count = $repository->getShoppingSessionCount();
    $this->assertEquals($original_count + 1, $new_count);
}

public function testFindAll()
{
    $repository = $this->entityManager->getRepository(\App\Entity\ShoppingSession::class);
    $our_count = $repository->getShoppingSessionCount();
    $get_all_count = count($repository->findAll());
    $this->assertEquals($our_count, $get_all_count);
}
}

```

Результат работы тестов:

```

[valeriadmitruk@MacBook-Pro-Valeria ElectronicsShop % bin/phpunit tests
PHPUnit 8.5.15 by Sebastian Bergmann and contributors.

Testing tests
.....                                         6 / 6 (100%)

Time: 163 ms, Memory: 18.00 MB

OK (6 tests, 6 assertions)

```

## Вывод

В данной лабораторной работе я познакомилась с механизмами модульного тестирования веб-приложений, построенных на гексагональной архитектуре.