public function testRemove()

{

$this->cart->add(5, 3);

$this->cart->remove(5);

$this->assertEquals([], $this->cart->getItems());

}

public function testClear()

{

$this->cart->add(5, 3);

$this->cart->add(7, 14);

$this->cart->clear();

$this->assertEquals([], $this->cart->getItems());

}

public function testCount()

{

$this->cart->add(5, 3);

$this->assertEquals(1, $this->cart->getCount()); $this->cart->add(7, 14);

$this->assertEquals(2, $this->cart->getCount());

}

public function testAmount()

{

$this->cart->add(5, 3);

$this->assertEquals(3, $this->cart->getAmount()); $this->cart->add(7, 14);

$this->assertEquals(17, $this->cart->getAmount());

}

public function testEmptyStorage()

{

$cart = new Cart();

$this->setExpectedException('yii\base\InvalidConfigException'); $cart->getItems();

}

}

5. Add a separated test for checking the SessionStorage class:

<?php

namespace book\cart\tests\storage;

use book\cart\storage\SessionStorage; use book\cart\tests\TestCase;

class SessionStorageTest extends TestCase {

/\*\*

\* @var SessionStorage \*/

private $storage;

public function setUp()

{

parent::setUp();

$this->storage = new SessionStorage(['key' => 'test']);

}

public function testEmpty()

{

$this->assertEquals([], $this->storage->load());