\*/

private $cart;

public function beforeTestMethod($method)

{

parent::beforeTestMethod($method );

$this->cart = new TestedCart(['storage' => new FakeStorage()]);

}

public function testEmpty()

{

$this->array($this->cart->getItems())->isEqualTo([]); $this->integer($this->cart->getCount())->isEqualTo(0); $this->integer($this->cart->getAmount())->isEqualTo(0);

}

public function testAdd()

{

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

$this->array($this->cart->getItems())->isEqualTo([5 => 3]); $this->cart->add(7, 14);

$this->array($this->cart->getItems())->isEqualTo([5 => 3, 7 => 14]); $this->cart->add(5, 10);

$this->array($this->cart->getItems())->isEqualTo([5 => 13, 7 => 14]);

}

public function testSet()

{

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

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

$this->cart->set(5, 12);

$this->array($this->cart->getItems())->isEqualTo([5 => 12, 7 => 14]);

}

public function testRemove()

{

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

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

$this->array($this->cart->getItems())->isEqualTo([]);

}

public function testClear()

{

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

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

$this->cart->clear();

$this->array($this->cart->getItems())->isEqualTo([]);

}

public function testCount()

{

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

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

$this->integer($this->cart->getCount())->isEqualTo(2);

}

public function testAmount()

{

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

$this->integer($this->cart->getAmount())->isEqualTo(3);