{

|  |  |  |
| --- | --- | --- |
|  | $this | ->rese |
|  | $this | ->cart |
| } |  |  |
| /\* | \* |  |
| \* | @When | I add |
| \* | / |  |
| pu | blic fu | nction |
| { |  |  |
|  | $this | ->cart |
| } |  |  |
| /\* | \* |  |
| \* | @When | I set |
| \* | / |  |
| pu | blic fu | nction |
| { |  |  |
|  | $this | ->cart |
| } |  |  |
| /\* | \* |  |
| \* | @When | I clea |
| \* | / |  |
| pu | blic fu | nction |
| { |  |  |
|  | $this | ->cart |
| } |  |  |
| /\* | \* |  |
| \* | @Then | I shou' |
| \* | / |  |
| pu | blic fu | nction |

{

PHPUnit\_Framework\_Assert::assertEquals( 0,

$this->cart->getCount()

);

}

/\*\*

\* i

\*/

@Then I should have :count product(s)

public function iShouldHaveProductInTheCart($count) {

PHPUnit\_Framework\_Assert::assertEquals( intval($count),

$this->cart->getCount()

);

} /\*\*

\* @Then the overall cart amount should be :amount

\*/

public function theOverallCartPriceShouldBePs($amount) {

PHPUnit\_Framework\_Assert::assertSame( intval($amount),

$this->cart->getAmount()

);

} /\*\*

\* @Then I should have :pieces of :product \*/