LAPORAN PRAKTIKUM MATA KULIAH REKAYASA PERANGKAT LUNAK

TUGAS PRAKTIKUM RPL 04
UNIT TESTING DENGAN PHPUNIT

Adam Arthur Faizal

M3119001

TI A

PROGRAM STUDI TEKNIK INFORMATIKA

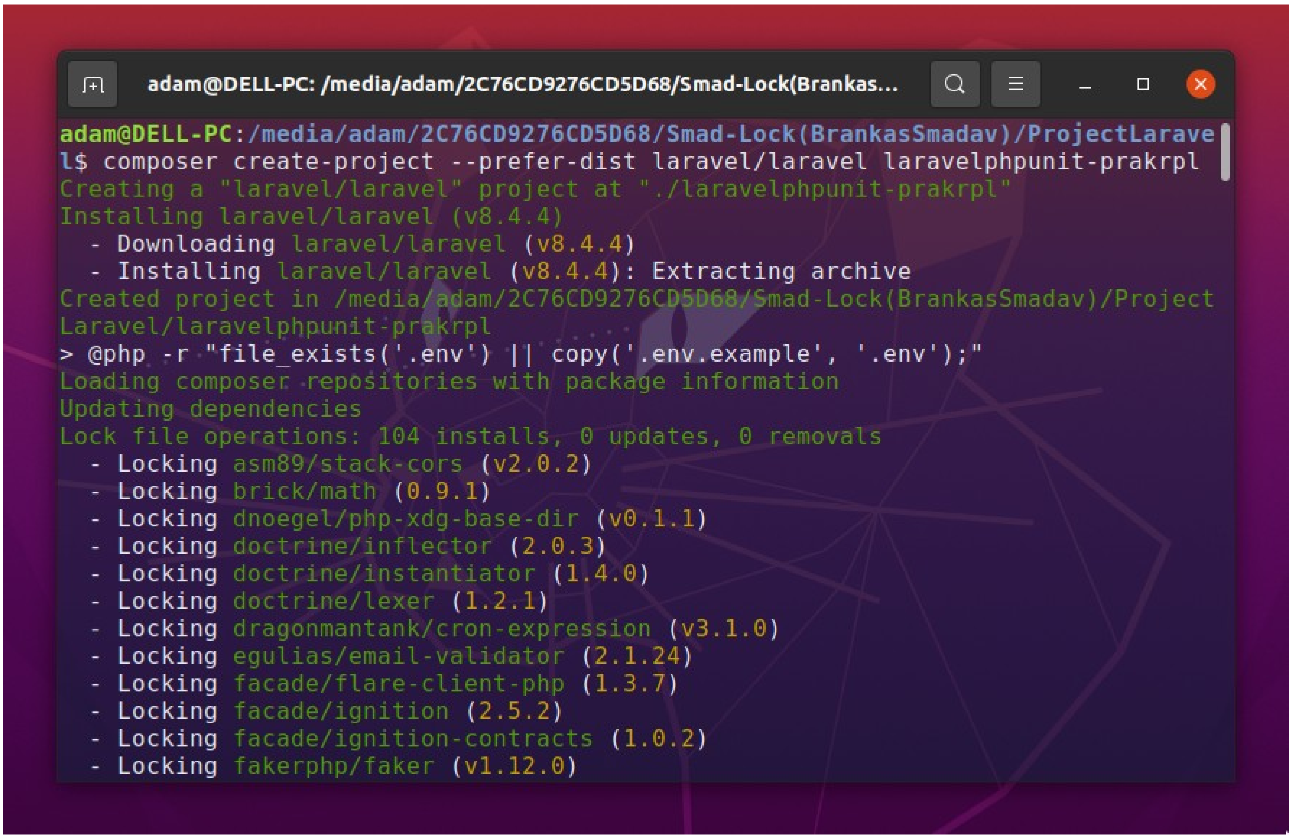
SEKOLAH VOKASI

UNIVERSITAS SEBELAS MARET

SURAKARTA

2020

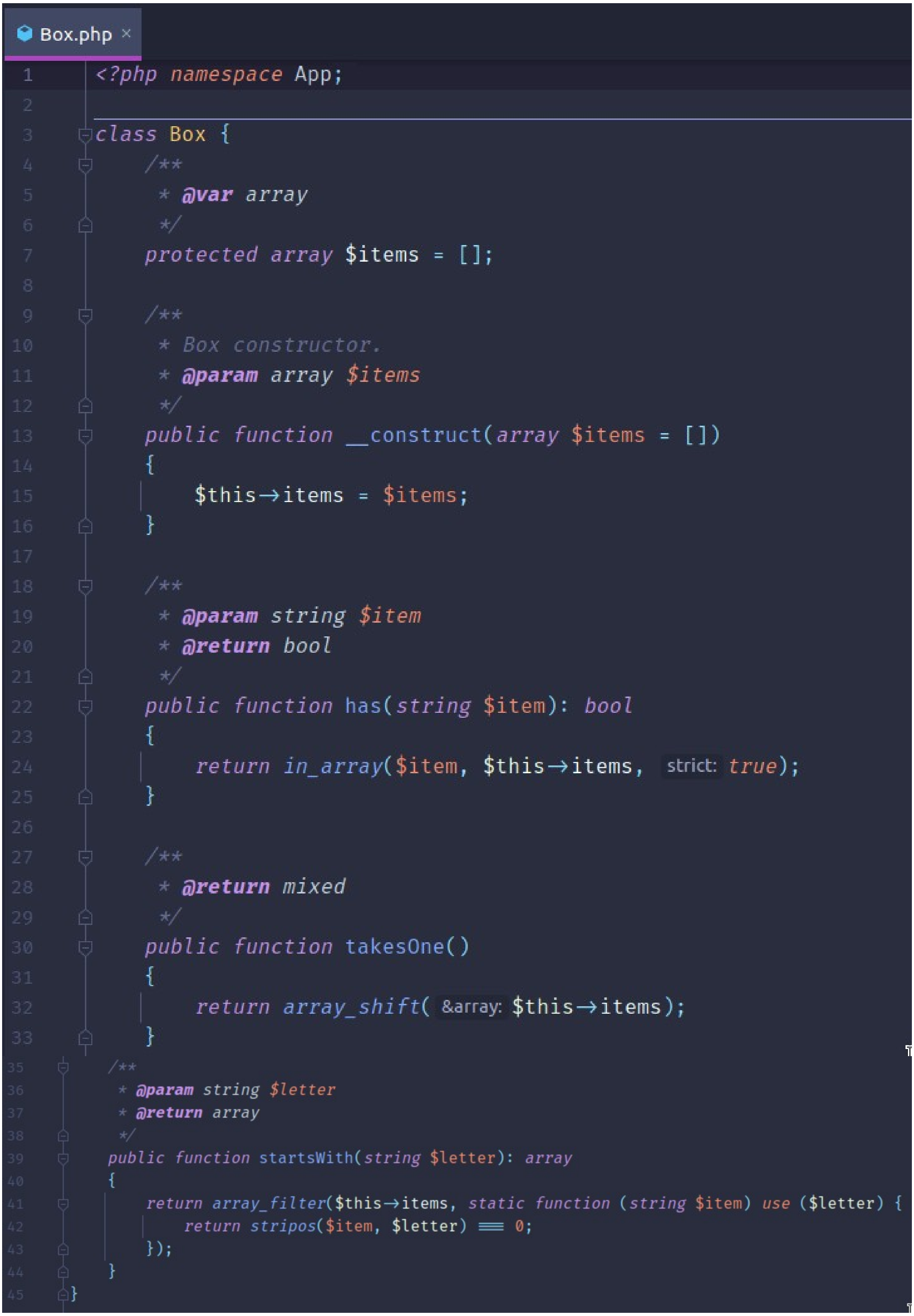
TUGAS



adam@DELL-PC: /media/adam/2C76CD9276CD5D68/Smad-Lock(Brankas...

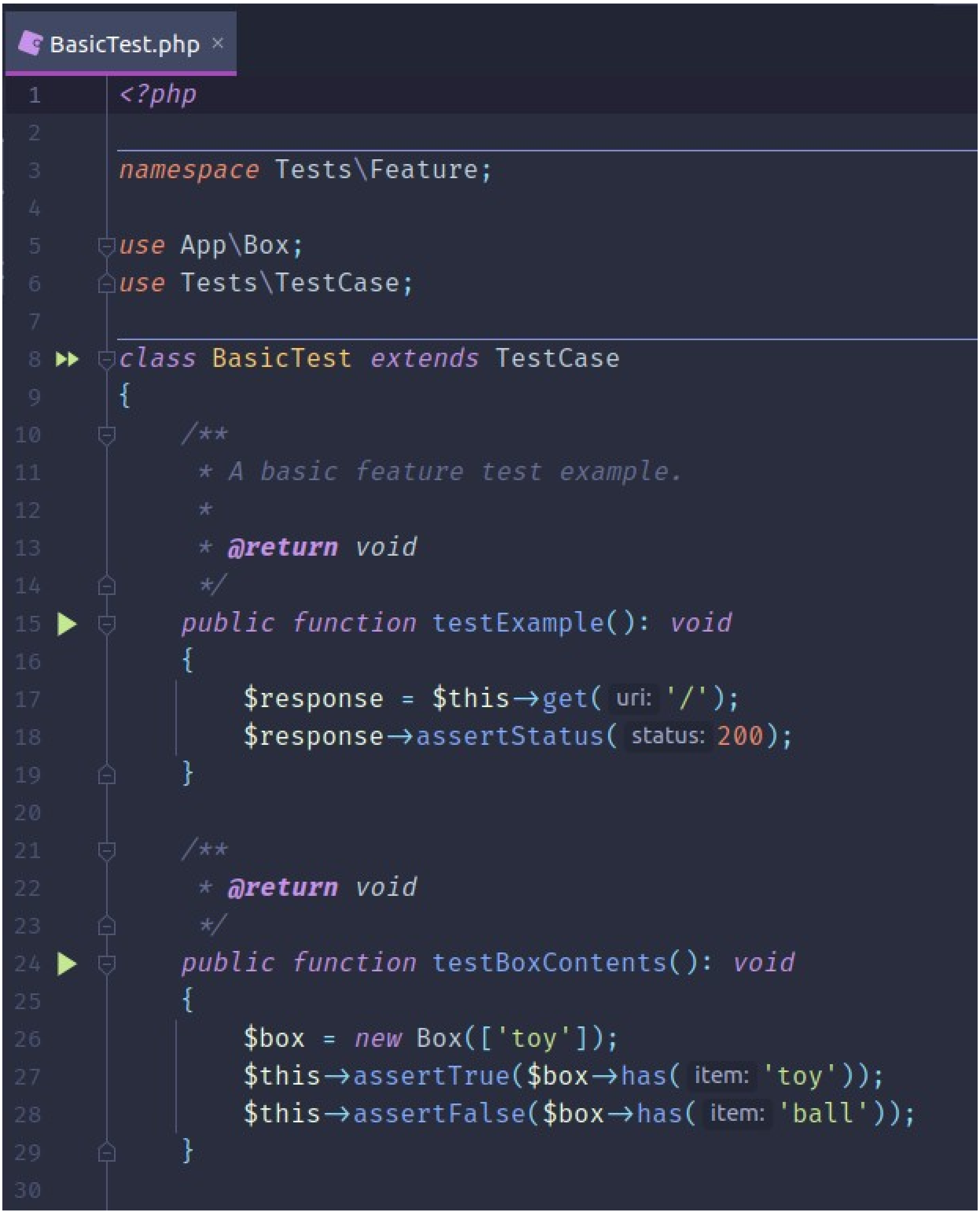
```
adam@DELL-PC:/media/adam/2C76CD9276CD5D68/Smad-Lock(BrankasSmadav)/ProjectLarave
l$ composer create-project --prefer-dist laravel/laravel laravelphpunit-prakrpl
Creating a "laravel/laravel" project at "./laravelphpunit-prakrpl"
Installing laravel/laravel (v8.4.4)
  - Downloading laravel/laravel (v8.4.4)
  - Installing laravel/laravel (v8.4.4): Extracting archive
Created project in /media/adam/2C76CD9276CD5D68/Smad-Lock(BrankasSmadav)/Project
Laravel/laravelphpunit-prakrpl
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 104 installs, 0 updates, 0 removals
  - Locking asm89/stack-cors (v2.0.2)
  - Locking brick/math (0.9.1)
  - Locking dnoegel/php-xdg-base-dir (v0.1.1)
  - Locking doctrine/inflector (2.0.3)
  - Locking doctrine/instantiator (1.4.0)
  - Locking doctrine/lexer (1.2.1)
  - Locking dragonmantank/cron-expression (v3.1.0)
  - Locking egulias/email-validator (2.1.24)
  - Locking facade/flare-client-php (1.3.7)
  - Locking facade/ignition (2.5.2)
  - Locking facade/ignition-contracts (1.0.2)
  - Locking fakerphp/faker (v1.12.0)
```

Proses instalasi Framework Laravel menggunakan composer

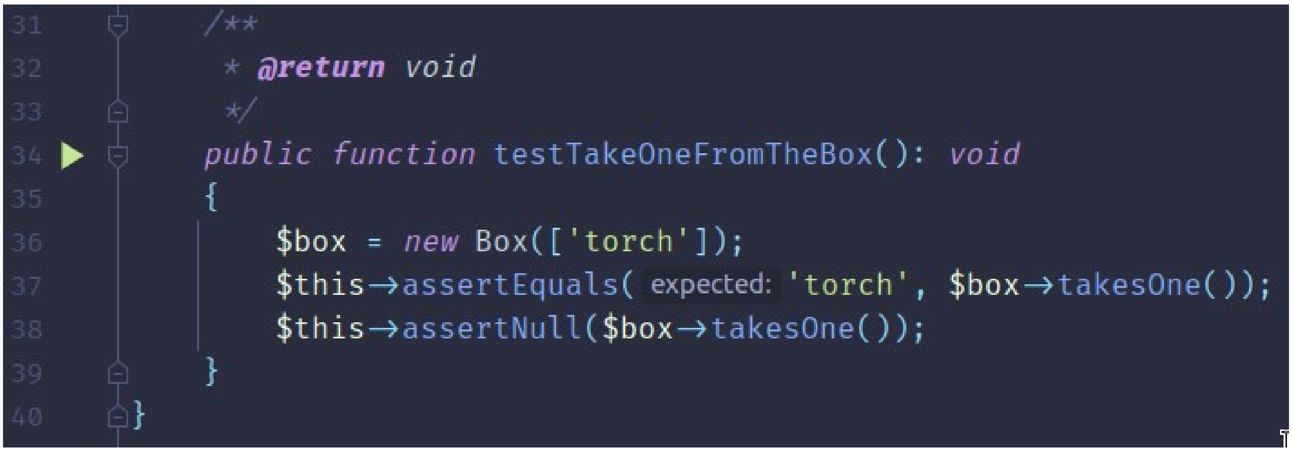


Box.php ×

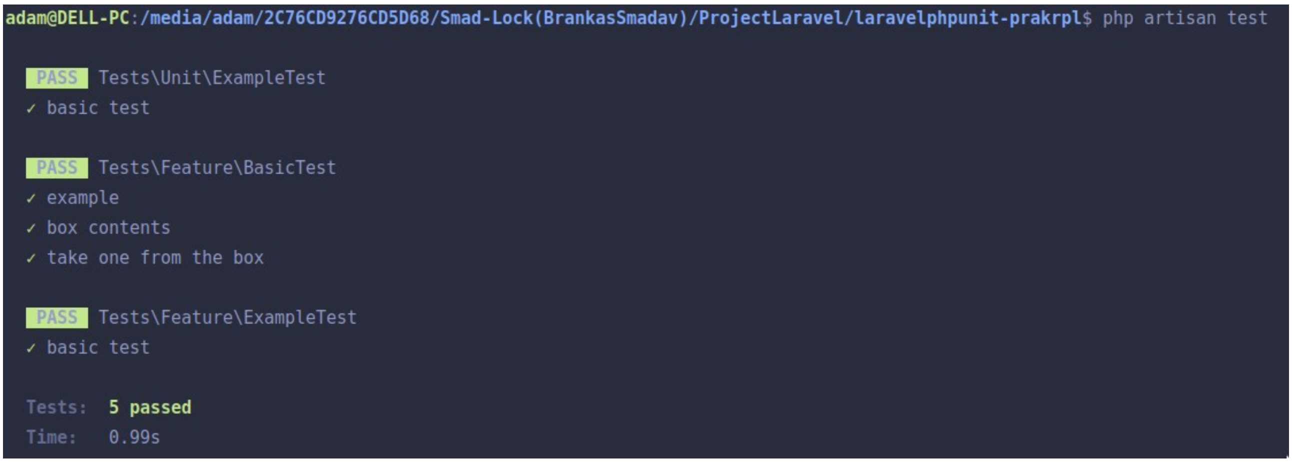
```php
<?php namespace App;

class Box {
    /**
     * @var array
     */
    protected array $items = [];

    /**
     * Box constructor.
     * @param array $items
     */
    public function __construct(array $items = [])
    {
        $this->items = $items;
    }

    /**
     * @param string $item
     * @return bool
     */
    public function has(string $item): bool
    {
        return in_array($item, $this->items,  strict: true);
    }

    /**
     * @return mixed
     */
    public function takesOne()
    {
        return array_shift( &array: $this->items);
    }
    /**
     * @param string $letter
     * @return array
     */
    public function startsWith(string $letter): array
    {
        return array_filter($this->items, static function (string $item) use ($letter) {
            return stripos($item, $letter) === 0;
        });
    }
}
```



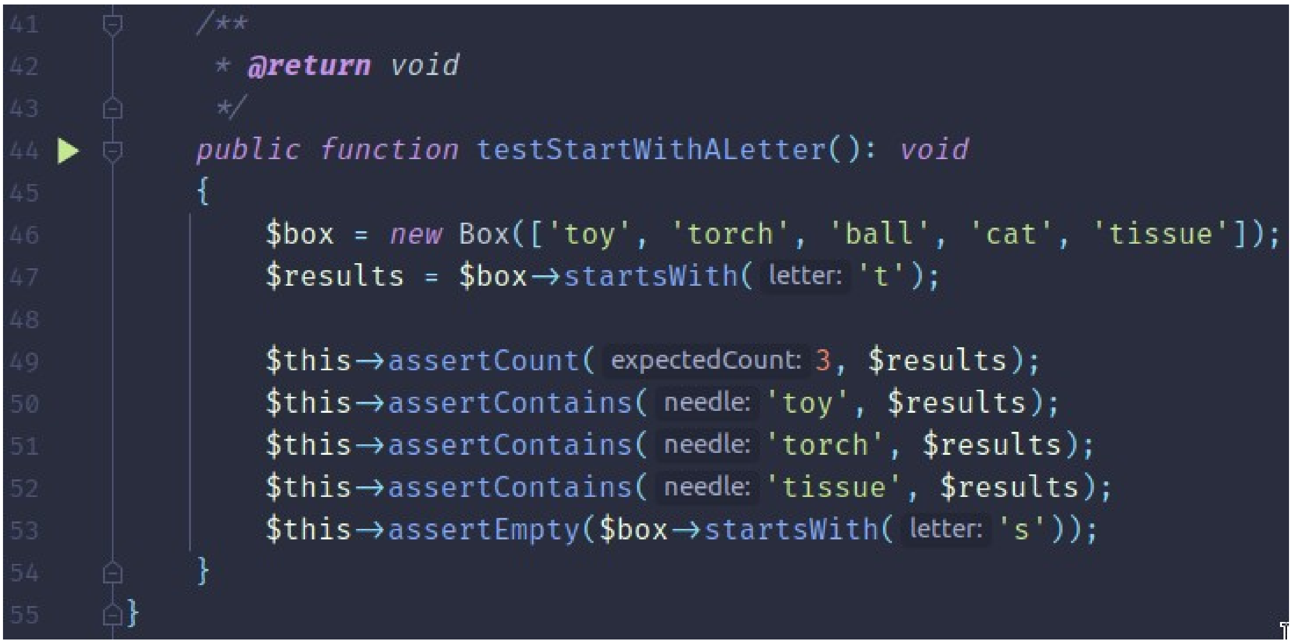
```php
<?php

namespace Tests\Feature;

use App\Box;
use Tests\TestCase;

class BasicTest extends TestCase
{
    /**
     * A basic feature test example.
     *
     * @return void
     */
    public function testExample(): void
    {
        $response = $this->get( uri: '/');
        $response->assertStatus( status: 200);
    }

    /**
     * @return void
     */
    public function testBoxContents(): void
    {
        $box = new Box(['toy']);
        $this->assertTrue($box->has( item: 'toy'));
        $this->assertFalse($box->has( item: 'ball'));
    }

```



```php
    /**
     * @return void
     */
    public function testTakeOneFromTheBox(): void
    {
        $box = new Box(['torch']);
        $this->assertEquals( expected: 'torch', $box->takesOne());
        $this->assertNull($box->takesOne());
    }
}
```



```
adam@DELL-PC:/media/adam/2C76CD9276CD5D68/Smad-Lock(BrankasSmadav)/ProjectLaravel/laravelphpunit-prakrpl$ php artisan test

   PASS  Tests\Unit\ExampleTest
  ✓ basic test

   PASS  Tests\Feature\BasicTest
  ✓ example
  ✓ box contents
  ✓ take one from the box

   PASS  Tests\Feature\ExampleTest
  ✓ basic test

  Tests:  5 passed
  Time:   0.99s
```



```php
    /**
     * @return void
     */
    public function testStartWithALetter(): void
    {
        $box = new Box(['toy', 'torch', 'ball', 'cat', 'tissue']);
        $results = $box->startsWith( letter: 't');

        $this->assertCount( expectedCount: 3, $results);
        $this->assertContains( needle: 'toy', $results);
        $this->assertContains( needle: 'torch', $results);
        $this->assertContains( needle: 'tissue', $results);
        $this->assertEmpty($box->startsWith( letter: 's'));
    }
}
```