

How it works...

In this case we have the main shoppingCart class with a low-level dependency, defined by an abstraction interface:

class ShoppingCart {

public function construct(StorageInterface $storage) { ... }

}

interface StorageInterface {

public function load();

public function save(array $items);

}

And we have some an implementation of the abstraction:

class SessionStorage implements StorageInterface