

```
class OrderProcessor
{
    public function checkout($order){/*...*/}
}
```

```
interface IOrderProcessor
{
    public function checkout($order);
}
```

```
class OrderProcessor implements IOrderProcessor
{
    public function checkout($order){/*...*/}
}
```

```
class Customer
{
    private $currentOrder = null;

    public function buyItems(IOrderProcessor $processor)
    {
        if(is_null($this->currentOrder)){
            return false;
        }

        return $processor->checkout($this->currentOrder);
    }

    public function addItem($item){
        if(is_null($this->currentOrder)){
            $this->currentOrder = new Order();
        }
        return $this->currentOrder->addItem($item);
    }

    public function deleteItem($item){
        if(is_null($this->currentOrder)){
            return false;
        }
        return $this->currentOrder ->deleteItem($item);
    }
}
```


