PHP-Design-Pattern: Strategy

Designprinzip

Algorithmen kapseln und austauschbar machen durch Komposition statt Vererbung.

PHP-Code

Strategieschnittstelle

```
interface Strategy {
    public abstract function algorithm();
}
```

Konkrete Strategien

```
class StrategyA implements Strategy{
    public function algorithm() { return 'AAAAAAAAAA'; }
}

class StrategyB implements Strategy{
    public function algorithm() { return 'BBBBBBBBBB'; }
}
```

Client

Muß bei Änderung der Strategie nicht verändert werden.

```
class Client {
    private $strategy = NULL;

    public function setStrategy(Strategy $s){
        $this->strategy = $s;
    }

    public function doSomething(){
        return $this->strategy->algorithm();
    }
}
```

Verwendung des Clients

Änderung der Strategie zur Laufzeit möglich

```
$client = new Client();

$client->setStrategy(new StrategyA());
echo $client->doSomething();

$client->setStrategy(new StrategyB());
echo $client->doSomething();
```

Literatur

[Wikipedia] http://de.wikipedia.org/wiki/Strategie_(Entwurfsmuster)