Design Pattern: Strategy

1 Designprinzip

Algorithmen kapseln und austauschbar machen durch Komposition statt Vererbung.

2 Java-Code

2.1 Strategieschnittstelle

```
public interface StrategieSchnittstelle {
        public abstract void algorithmus();
}
```

2.2 Konkrete Strategien

```
public class StrategieA implements StrategieSchnittstelle {
        @Override
        public void algorithmus() { System.out.println("Version A"); }
}
```

```
public class StrategieB implements StrategieSchnittstelle {
        @Override
        public void algorithmus() { System.out.println("Version B"); }
}
```

2.3 Client

Muß bei Änderung der Strategie nicht verändert werden.

```
public class Client {
        private StrategieSchnittstelle algorithmus;

        public void setStrategie(StrategieSchnittstelle al){ algorithmus = al; }

        public void tuwas(){ algorithmus.algorithmus(); }
}
```

2.4 Verwendung des Clients

Änderung der Strategie zur Laufzeit möglich

```
Client client = new Client();
client.setStrategie(new StrategieA());
client.tuwas();    // "Version A"
client.setStrategie(new StrategieB());
client.tuwas();    // "Version B"
```

Literatur

[Head First Design Patterns] http://www.oreilly.de/catalog/9780596007126/

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