



SEW S04 / ELEMENT-FACTORY

Implementierung des Factory-Method-Patterns



30. NOVEMBER 2014

STEFAN ERCEG
4AHITT

Inhalt

1. Github.....	1
2. Aufgabenstellung.....	2
3. Zeitabschätzung.....	3
4. Tatsächlicher Zeitaufwand	3
5. Designüberlegung.....	4
6. Arbeitsdurchführung	5
6.1 Anwendung des Factory-Method-Patterns.....	5
6.2 Einlesen der Eingaben des Benutzers.....	5
7. Testfälle	6
7.1 Ausgabe bestimmter Elemente in XML-Form	6
7.2 Ausgabe bestimmter Elemente in Yaml-Form	6
7.3 Ausgabe aller Elemente in XML- und Yaml-Form.....	7
7.4 Fehlermeldung bei falscher Eingabe von DB-Zugangsdaten	8
7.5 Fehlermeldung bei Eingabe von nicht existierendem Element.....	8
7.6 Fehlermeldung bei Eingabe von nicht existierender Factory.....	9
8. Lessons learned	10
9. Quellen	10

1. Github

Link: <https://github.com/serceg-tgm/4AHITT-SEW/tree/master/S04>

Tag: erceg_s04

2. Aufgabenstellung

Eine PHP-CLI-Anwendung soll Elemente der Propel-Konfiguration in verschiedenen "Dialekten" erzeugen z.B. YamlAdapter und XMLAdapter, YamlConnection und XMLConnection, ... es sollten zumindest 4 Parameter der Propel-Configuration umgesetzt werden.

Die einzelnen Objekte der Klassen nehmen im Konstruktor einen value entgegen, ergänzen ihn um das richtige Schlüsselwort und geben dieses über eine getString()-Methode wieder aus.

Beispiel (in Java-Notation):

```
Element y = new YamlAdapter('mysql');
System.out.print(y.getString()); // es wird "adapter: mysql" ausgegeben

Element x = new XMLAdapter('mysql');
System.out.print(x.getString()); // es wird "<adapter>mysql</adapter>" ausgegeben
```

Verwende das Factory-Method-Pattern um die Installation der mindestens 8 Klassen zu bewerkstelligen.

Welche Elemente verarbeitet werden, soll die Applikation aus einer SQL-Tabelle (Relationenmodell element(id, type, value)) lesen - die dafür erforderlichen Connection-Daten (Server, DB, Username, ...) sollen ebenfalls über Parameter auf die Kommandozeile angegeben werden können.

Die Factory soll über einen Kommandozeilen-Parameter angegeben werden können.

3. Zeitabschätzung

Teilaufgabe	Zeit
UML-Diagramm erstellen	20 Minuten
Implementierung der Anwendung	60 Minuten
Testen der Anwendung	30 Minuten
Protokoll schreiben	30 Minuten
<i>Gesamt</i>	140 Minuten (2 h 20 min)

4. Tatsächlicher Zeitaufwand

Teilaufgabe	Zeit
UML-Diagramm erstellen	10 Minuten
Implementierung der Anwendung	90 Minuten
Testen der Anwendung	20 Minuten
Protokoll schreiben	40 Minuten
<i>Gesamt</i>	160 Minuten (2h 40 min)

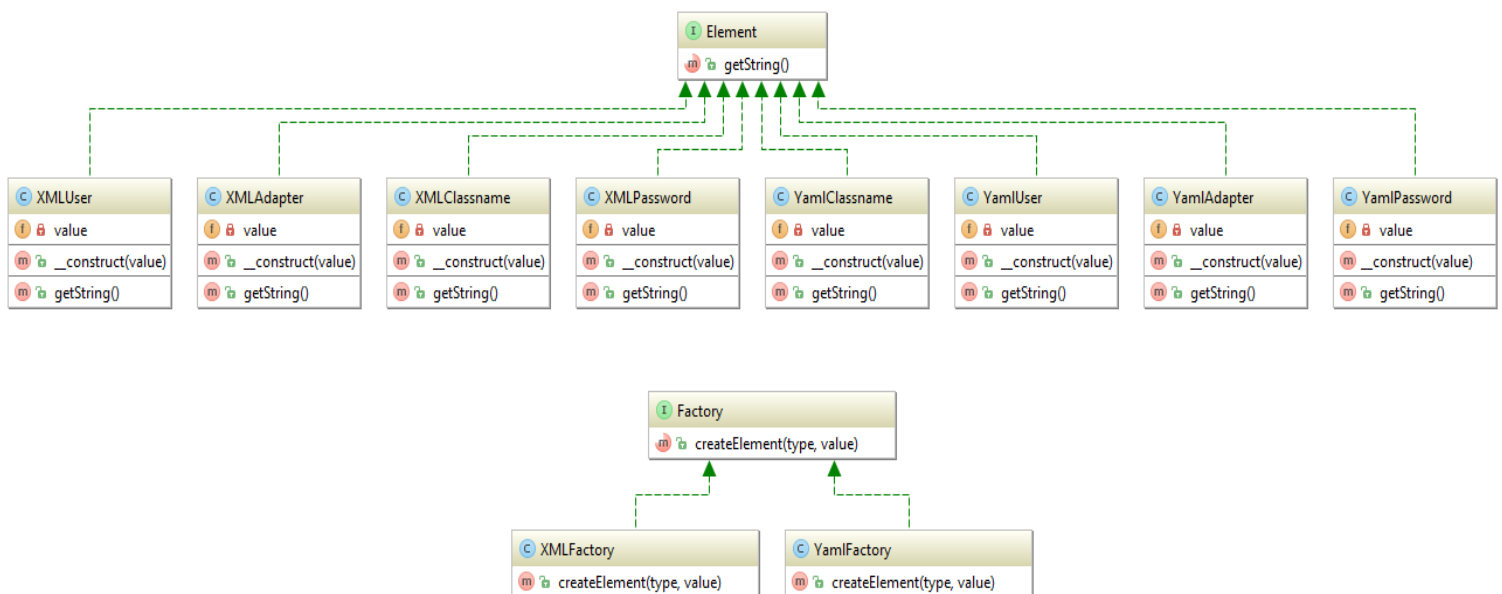
5. Designüberlegung

Da das Factory-Method-Pattern bei dieser Aufgabe sehr nützlich ist, wurde beim Entwurf nach diesem Pattern vorgegangen.

Ein Interface Element wurde erstellt, welches von den Klassen, die ein bestimmtes Element darstellen (z.B. YamlAdapter und XMLAdapter), implementiert wird. Dabei kamen 8 Klassen zustande, da 4 Element-Klassen zu je einer Factory (XML und Yaml) erstellt wurden.

Das Interface Factory besitzt die Factory-Method. Die Klassen XMLFactory und YamlFactory implementieren das Interface und somit auch die Factory-Method.

Folgendes Diagramm wurde auf Basis der vorher beschriebenen Überlegungen erstellt:



Powered by yFiles

6. Arbeitsdurchführung

Da die Anforderung lautete, eine PHP-CLI-Anwendung zu erstellen, wurde dieses Programm mittels PHP und nicht wie bei den meisten bisherigen SEW-Aufgaben mit Java umgesetzt.

6.1 Anwendung des Factory-Method-Patterns

Im Main-File des Programms, welches dann über die CLI mittels „php <Pfad zur Datei>“ aufgerufen wird, wurde die Factory je nach Benutzereingabe folgendermaßen zugewiesen:

```
$factory = new XMLFactory();
```

oder

```
$factory = new YamlFactory();
```

Das Element wurde dann dementsprechend mit dem Befehl

```
$element = $factory->createElement($type, $value);
```

initialisiert.

In den Klassen XMLFactory und YamlFactory, welche beide das Interface Factory implementieren, wurde folgendermaßen der Typ des Elements herausgefunden und ein neu erstelltes Objekt zurückgegeben:

```
if(strtolower($type) === "adapter")  
    return new YamlAdapter($value);
```

6.2 Einlesen der Eingaben des Benutzers

Das Einlesen der Eingaben des Benutzers erfolgte mit dem Befehl

```
trim(fgets(STDIN)); [1]
```

Diese Eingaben wurden jeweils in eine Variable gespeichert. Die ersten vier Eingaben, welche die DB-Zugangsdaten sind, wurden dann verwendet, um eine Verbindung zu der angegebenen Datenbank mittels

```
mysqli_connect($dbserver, $username, $password, $dbname);
```

aufzubauen. [2]

Konnte sich der Benutzer erfolgreich zur Datenbank verbinden, wird er gefragt, welche Elemente er ausgeben möchte. Gibt er dort ein in der Datenbank existierendes Element ein bzw. „alle“, falls er alle Elemente ausgeben möchte, wird die Frage nach der einzusetzenden Factory durchgeführt. Erfolgt bei dieser Eingabe ebenfalls kein Fehler, werden die gewünschten Elemente ausgegeben.

7. Testfälle

7.1 Ausgabe bestimmter Elemente in XML-Form

Bei diesem Testfall werden alle für den Datenbankverbindungs Aufbau notwendigen Eingaben korrekt eingegeben, als Element „adapter“ und als Factory „XML“ verwendet.

```
C:\Users\Acer>php C:\Users\Acer\Documents\PhpstormProjects\S04\Main.php
Bitte geben Sie die Adresse des Datenbankservers ein!
127.0.0.1
Bitte geben Sie den Namen des Users ein!
factory
Bitte geben Sie das Passwort des Users ein!
factory
Bitte geben Sie den Namen der Datenbank ein!
factory
Bitte geben Sie ein, welchen Elemente ausgegeben werden sollen
(zur Auswahl stehen 'adapter', 'classname', 'user' und 'password')
Wenn Sie alle Elemente ausgeben moechten, muessen Sie 'alle' eingeben.
adapter
Bitte geben Sie ein, welche Factory Sie verwenden moechten
(zur Auswahl stehen XML und Yaml)!
xml
<adapter> mysql </adapter>
<adapter> postgresql </adapter>
<adapter> db2 </adapter>
```

-> Testfall erfolgreich, da alle in der Datenbank enthaltenden Adapter-Elemente in XML-Form ausgegeben werden

7.2 Ausgabe bestimmter Elemente in Yaml-Form

Bei diesem Testfall werden alle für den Datenbankverbindungs Aufbau notwendigen Eingaben korrekt eingegeben, als Element „user“ und als Factory „Yaml“ verwendet.

```
C:\Users\Acer>php C:\Users\Acer\Documents\PhpstormProjects\S04\Main.php
Bitte geben Sie die Adresse des Datenbankservers ein!
127.0.0.1
Bitte geben Sie den Namen des Users ein!
factory
Bitte geben Sie das Passwort des Users ein!
factory
Bitte geben Sie den Namen der Datenbank ein!
factory
Bitte geben Sie ein, welchen Elemente ausgegeben werden sollen
(zur Auswahl stehen 'adapter', 'classname', 'user' und 'password')
Wenn Sie alle Elemente ausgeben moechten, muessen Sie 'alle' eingeben.
user
Bitte geben Sie ein, welche Factory Sie verwenden moechten
(zur Auswahl stehen XML und Yaml)!
yaml
user: Stefan Erceg
user: Michael Borko
user: Erhard List
```

-> Testfall erfolgreich, da alle in der Datenbank enthaltenden User-Elemente in Yaml-Form ausgegeben werden

7.3 Ausgabe aller Elemente in XML- und Yaml-Form

Bei diesem Testfall werden alle für den Datenbankverbindungs Aufbau notwendigen Eingaben korrekt eingegeben, als Element „alle“ eingegeben, damit alle verfügbaren Elemente angezeigt werden, und als Factory einmal „XML“ und einmal „Yaml“ verwendet.

XML-Form:

```
C:\Users\Acer>php C:\Users\Acer\Documents\PhpstormProjects\S04\Main.php
Bitte geben Sie die Adresse des Datenbankservers ein!
127.0.0.1
Bitte geben Sie den Namen des Users ein!
factory
Bitte geben Sie das Passwort des Users ein!
factory
Bitte geben Sie den Namen der Datenbank ein!
factory
Bitte geben Sie ein, welchen Elemente ausgegeben werden sollen
(zur Auswahl stehen 'adapter', 'classname', 'user' und 'password')
Wenn Sie alle Elemente ausgehen moechten, muessen Sie 'alle' eingeben.
alle
Bitte geben Sie ein, welche Factory Sie verwenden moechten
(zur Auswahl stehen XML und Yaml)?
xml
<adapter> mysql </adapter>
<adapter> postgresql </adapter>
<adapter> db2 </adapter>
<classname> PropelRuntimeConnectionConnectionWrapper1 </classname>
<classname> PropelRuntimeConnectionConnectionWrapper2 </classname>
<classname> PropelRuntimeConnectionConnectionWrapper3 </classname>
<user> Stefan Erceg </user>
<user> Michael Borko </user>
<user> Erhard List </user>
<password> bla1 </password>
<password> bla2 </password>
<password> bla3 </password>
```

Yaml-Form:

```
C:\Users\Acer>php C:\Users\Acer\Documents\PhpstormProjects\S04\Main.php
Bitte geben Sie die Adresse des Datenbankservers ein!
127.0.0.1
Bitte geben Sie den Namen des Users ein!
factory
Bitte geben Sie das Passwort des Users ein!
factory
Bitte geben Sie den Namen der Datenbank ein!
factory
Bitte geben Sie ein, welchen Elemente ausgegeben werden sollen
(zur Auswahl stehen 'adapter', 'classname', 'user' und 'password')
Wenn Sie alle Elemente ausgehen moechten, muessen Sie 'alle' eingeben.
alle
Bitte geben Sie ein, welche Factory Sie verwenden moechten
(zur Auswahl stehen XML und Yaml)?
yaml
adapter: mysql
adapter: postgresql
adapter: db2
classname: PropelRuntimeConnectionConnectionWrapper1
classname: PropelRuntimeConnectionConnectionWrapper2
classname: PropelRuntimeConnectionConnectionWrapper3
user: Stefan Erceg
user: Michael Borko
user: Erhard List
password: bla1
password: bla2
password: bla3
```

-> Testfall erfolgreich, da alle in der Datenbank enthaltenden Elemente sowohl in XML-Form, als auch in Yaml-Form korrekt ausgegeben werden

7.4 Fehlermeldung bei falscher Eingabe von DB-Zugangsdaten

Wird bei den DB-Zugangsdaten eine nicht zugängliche Datenbankserveradresse, ein nicht existierender Username und –password oder eine nicht existierende Datenbank angegeben, wird eine Fehlermeldung geworfen und das Programm wird beendet.

```
C:\Users\Acer>php C:\Users\Acer\Documents\PhpstormProjects\S04\Main.php
Bitte geben Sie die Adresse des Datenbankservers ein!
127.0.0.1
Bitte geben Sie den Namen des Users ein!
falsch
Bitte geben Sie das Passwort des Users ein!
existiert nicht!
Bitte geben Sie den Namen der Datenbank ein!
100 %-ig!
PHP Warning: mysqli_connect(): (28000/1045): Access denied for user 'falsch'@'localhost' (using password: YES) in C:\Users\Acer\Documents\PhpstormProjects\S04\Main.php on line 42
Es wurden falsche Daten fuer den Verbindungsaufbau zur Datenbank eingegeben!
C:\Users\Acer>
```

-> Testfall erfolgreich, da eine Fehlermeldung ausgegeben und das Programm beendet wird

7.5 Fehlermeldung bei Eingabe von nicht existierendem Element

Wird ein Element angegeben, welches nicht angeboten wird, wird eine Fehlermeldung geworfen und das Programm wird beendet. Elemente, die angeboten werden, sind „adapter“, „classname“, „user“ und „password“. Die Eingabe von „alle“ wird ebenfalls erlaubt, da dann alle in der Datenbank enthaltenden Elemente ausgegeben werden.

```
C:\Users\Acer>php C:\Users\Acer\Documents\PhpstormProjects\S04\Main.php
Bitte geben Sie die Adresse des Datenbankservers ein!
127.0.0.1
Bitte geben Sie den Namen des Users ein!
falsch
Bitte geben Sie das Passwort des Users ein!
existiert nicht!
Bitte geben Sie den Namen der Datenbank ein!
100 %-ig!
PHP Warning: mysqli_connect(): (28000/1045): Access denied for user 'falsch'@'localhost' (using password: YES) in C:\Users\Acer\Documents\PhpstormProjects\S04\Main.php on line 42
Es wurden falsche Daten fuer den Verbindungsaufbau zur Datenbank eingegeben!
C:\Users\Acer>php C:\Users\Acer\Documents\PhpstormProjects\S04\Main.php
Bitte geben Sie die Adresse des Datenbankservers ein!
127.0.0.1
Bitte geben Sie den Namen des Users ein!
factory
Bitte geben Sie das Passwort des Users ein!
factory
Bitte geben Sie den Namen der Datenbank ein!
factory
Bitte geben Sie ein, welchen Elemente ausgegeben werden sollen
(zur Auswahl stehen 'adapter', 'classname', 'user' und 'password')
Wenn Sie alle Elemente ausgeben moechten, muessen Sie 'alle' eingeben.
bla
Das eingegebene Element existiert nicht!
C:\Users\Acer>
```

-> Testfall erfolgreich, da eine Fehlermeldung ausgegeben und das Programm beendet wird

7.6 Fehlermeldung bei Eingabe von nicht existierender Factory

Wird eine Factory angegeben, welches nicht angeboten wird, wird eine Fehlermeldung geworfen und das Programm wird beendet. Bei diesem Programm können die Factories „XML“ und „Yaml“ genutzt werden.

```
C:\Users\Acer>php C:\Users\Acer\Documents\PhpstormProjects\S04\Main.php
Bitte geben Sie die Adresse des Datenbankservers ein!
127.0.0.1
Bitte geben Sie den Namen des Users ein!
factory
Bitte geben Sie das Passwort des Users ein!
factory
Bitte geben Sie den Namen der Datenbank ein!
factory
Bitte geben Sie ein, welchen Elemente ausgegeben werden sollen
    (zur Auswahl stehen 'adapter', 'classname', 'user' und 'password')
Wenn Sie alle Elemente ausgeben moechten, muessen Sie 'alle' eingeben.
password
Bitte geben Sie ein, welche Factory Sie verwenden moechten
    (zur Auswahl stehen XML und Yaml)?
bla
Die eingegebene Factory existiert nicht!
C:\Users\Acer>
```

-> Testfall erfolgreich, da eine Fehlermeldung ausgegeben und das Programm beendet wird

8. Lessons learned

- gelernt, wann es sinnvoll ist, ein Factory-Method-Pattern anzuwenden
- Skills in PHP erweitert, da ich bisher nicht oft Klassen und Interfaces in PHP erstellt, sondern PHP eher prozedural verwendet habe

9. Quellen

- [1] The PHP Group (2001, 2014). PHP – Ein- und Ausgabestreams [Online]. Available at: <http://php.net/manual/de/features.commandline.io-streams.php>
[abgerufen am 30.11.2014]
- [2] W3Schools (1999, 2014). PHP mysqli_connect() Function [Online]. Available at: http://www.w3schools.com/php/func_mysqli_connect.asp
[abgerufen am 30.11.2014]