Objektorientierte Programmierung WS 2013/2014 - Aufgabenblatt 5

Prof. Dr. U. Kastens
Institut für Informatik, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn

Ausgabe: 10.12.2013

Aufgabe 1 (Factory Method)

Das C++-Programm in `restaurant.cc` simuliert Abläufe in Restaurants. Es macht Gebrauch von Fabrikmethoden. Ordnen Sie die Rollen des Musters "Factory Method" ("Product", "Concrete Product", "Creator", "Concrete Creator") den Klassen der Restaurant-Simulation zu, indem Sie entsprechende Kommentare einfügen.

Aufgabe 2 (Abstract Factory)

Entwerfen Sie unter Verwendung des Entwurfsmusters "Abstract Factory" ein Java-Programm, das je nach Jahreszeit eine Weihnachts- bzw. Oster-Schokoladenfabrik erstellt. In der Fabrik werden zwei Arten von Festtagssüßigkeiten hergestellt: Hohlfiguren und gefüllte Kugeln. An Weihnachten gibt es Schoko-Weihnachtsmänner und gefüllte Schoko-Kugeln. An Ostern gibt es Schoko-Osterhasen und gefüllte Schoko-Ostereier. Zeichnen Sie zunächst das Abstract Factory-Schema für dieses Beispiel. Implementieren Sie dann die Klassen in Java.

Hier ist das Hauptprogramm in der Klasse `Client`, das zur Jahreszeit passende Süssigkeiten produziert:

```
import java.util.Calendar;

class Client {
  public static void main (String [] args) {
    int month = Calendar.getInstance().get(Calendar.MONTH);
    FestFactory f = null;

    switch (month) {
      case 2: case 3: case 4: f = new OsterFactory();
          break;
      case 10: case 11: case 12:  f = new WeihnachtsFactory();
          break;
      default: System.out.println("Momentan ist weder Weihnachten noch Ostern! :(");
    }
    Hohlfigur h = f.CreateHohlfigur();
    Kugel k = f.CreateKugel();

    f.print(); h.print(); k.print();
   }
}
```

seine Ausgabe ist momentan (Anfang Dezember):

```
Es ist Weihnachten.
Es gibt Schokoladen-Weihnachtsmaenner.
Es gibt Schokokugeln.
```

Aufgabe 3 (Bridge)

Die Java-Anwendung im Verzeichnis `blatt5/bridge` verwendet des Entwurfsmusters "Bridge". Ordnen Sie die Rollen im Brücken-Muster den Klassen dieser Anwendung zu. Erweitern Sie die Anwendung, um die folgenden charakteristischen Eigenschaften des Musters zu demonstrieren:

- Die benutzte Implementierung kann dynamisch gewechselt werden.
- Unabhängig von der Abstraktion kann eine neue Implementierungsart ergänzt werden, zum Beispiel als Ausgabeformat eine kommaseparierte, geklammerte Liste.
- Unabhängig von den Implementierungen kann eine verfeinerte Abstraktion hinzugefügt werden, z.B. "drucke eine einfache Folge von Strings".