Composite Design Pattern

Übersicht

Name: Composite Pattern

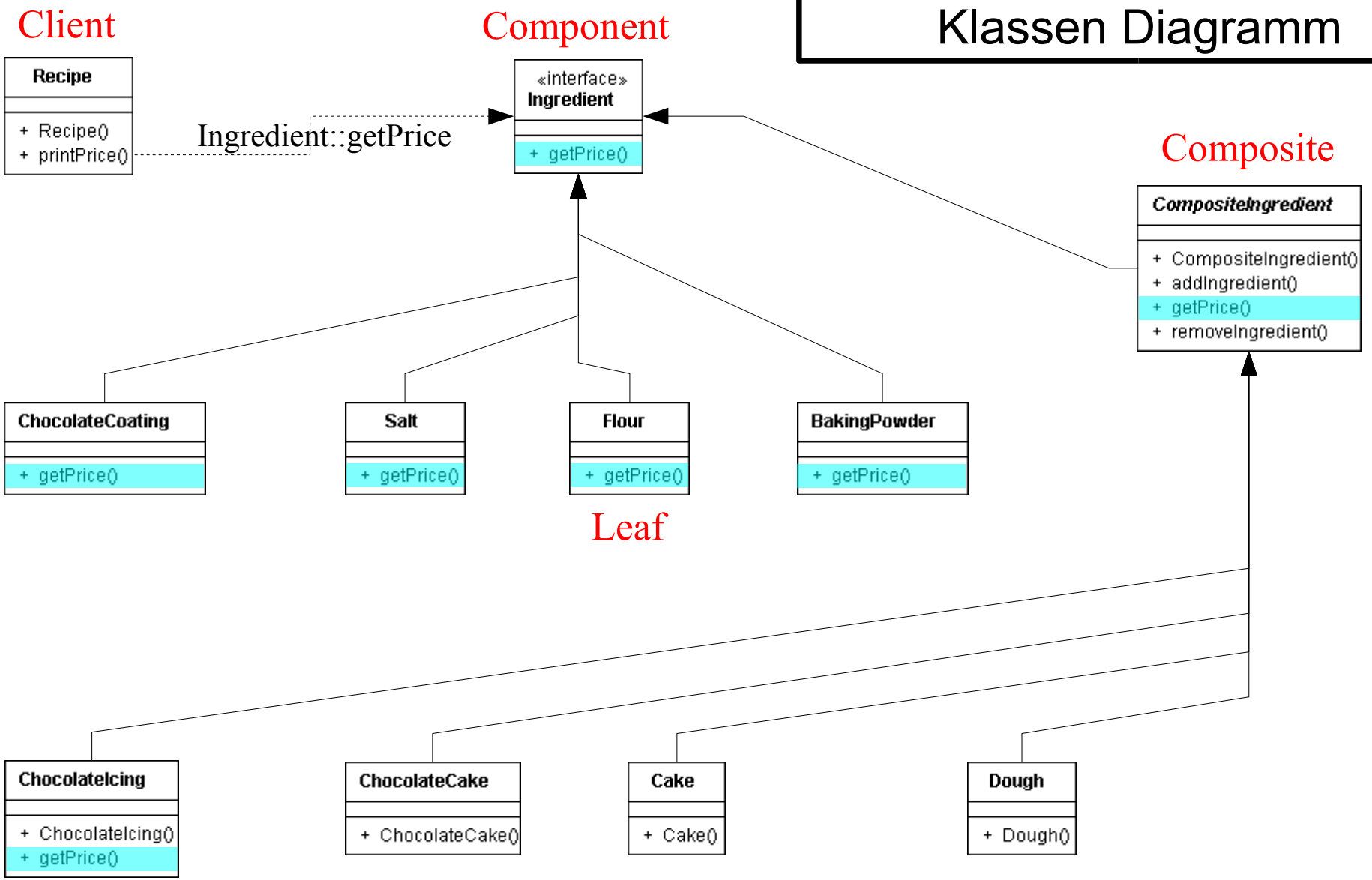
Typ: Structural Pattern

Problem: Einzelelemente und zusammengesetzte Elemente sollen vom Client gleich behandelt werden können, ohne daß er den Unterschied “merkt”.

Lösung: Sowohl die Einzelelemente als auch die zusammengesetzten Elemente implementieren das gleiche Interface (*Component*). Zusammengesetzte Elementie implementieren ein zusätzliches Interface (*Composite*), daß Operationen auf der Elementzusammenstellung ermöglicht. Client operiert nicht mit den konkreten Typen sondern mit dem gemeinsamen Interface (*Component*).

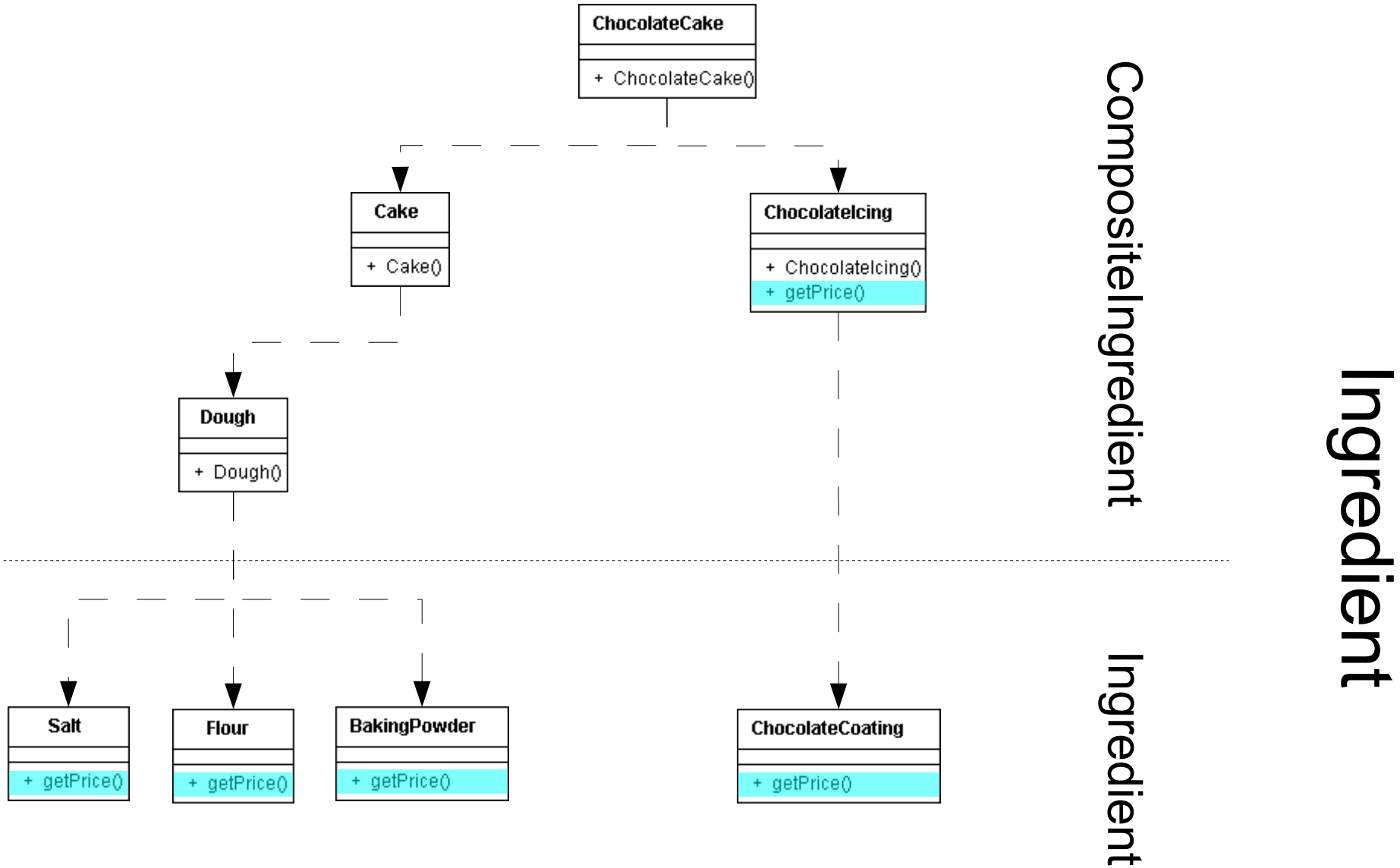
Composite Design Pattern

Klassen Diagramm



Composite Design Pattern

Objekt Diagramm



Composite Design Pattern

Source Code - Component

```
public interface Ingredient {

    float getPrice();

}
```

Composite Design Pattern

Source Code - Composite

```
import java.util.*;

public abstract class CompositeIngredient implements Ingredient {
    private Vector children = null;

    public CompositeIngredient(){
        children = new Vector();
    }

    public float getPrice() {
        float sum = 0;
        Iterator iterator = null;

        for(iterator = children.iterator();iter.hasNext();){
            sum += ((Ingredient)(iter.next())).getPrice();
        }

        return(sum);
    }

    public void addIngredient(Ingredient i){
        children.add(i);
    }
    public void removeIngredient(Ingredient i){
        children.remove(i);
    }
}
```

Composite Design Pattern

Source Code - Flour

```
public class Flour implements Ingredient{
    public float getPrice() {
        return(0.59f);
    }
}
```

Composite Design Pattern

Source Code - Cake

```
public class Cake extends CompositeIngredient {
    public Cake(){
        this.addIngredient(new Dough());
    }
}
```

Composite Design Pattern

Bemerkungen

Indikatoren für den Einsatz des Patterns:

- hirarchisch organisierte Objekte
- Eltern-Objekte verhalten Sich wie Kind-Objekte
- Hirarchie soll einfach erweiterbar sein

Wichtige Merkmale des Patterns:

- Composite Interface ist abgeleitet von Component Interface
- Leafs implementieren Component Interface
- Composites implementieren Composite Interface

Probleme des Composite Patterns:

- Composite und Component Interface muß für alle Eventualitäten ausgelegt sein => vielleicht zu generell
- Es kann nicht sichergestellt werden, daß ein Composite nur bestimmte Implementierung von Component oder Composite enthält

Composite Design Pattern

Quellen

Informationen über Patterns

- Design Patterns
  E. Gamma, R. Helm, R. Johnson, J. Vlissides
  ISBN: 0-201-63361-2
- http://www.dofactory.com/patterns/Patterns.aspx

Beispiel Implementierung

- http://www.hdm-stuttgart.de/~ns019/downloads/patterns/CompositePattern.jar

Vortrag zum Downloaden

- http://www.hdm-stuttgart.de/~ns019/downloads/patterns/CompositePattern.pdf