

---

Aufgaben zur Klausur **Softwaredesign** im WS 03/04 (WI h252, WI h253, II h752, MI h403, MI h404, MI h405)

Zeit: 60 Minuten

erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen ausschließlich an den gekennzeichneten Stellen in das Aufgabenblatt ein. Ist ihre Lösung wesentlich umfangreicher, so überprüfen Sie bitte nochmals Ihren Lösungsweg.

Sollten Unklarheiten oder Mehrdeutigkeiten bei der Aufgabenstellung auftreten, so notieren Sie bitte, wie Sie die Aufgabe interpretiert haben.

Viel Erfolg !

Diese Klausur besteht einschließlich dieses Deckblattes aus 9 Seiten

---

### Aufgabe 1:

Gegeben seien der folgenden Java-Code

---

```
abstract class A {  
    abstract void f();  
}
```

```
class B extends A {  
    void f() { }  
}
```

```
class C extends A {  
    A d;  
  
    void f() { d.f(); }  
}
```

---

Geben Sie das zugehörige OMT-Diagramm an:

Ist in dieser Struktur ein Entwurfsmuster enthalten? Wenn ja, welches?

ja  nein

Begründung:

.....

Gegeben seien der folgenden Java-Code

---

```
abstract class L {
    abstract void f();
}
class M extends L {
    void f() { }
}
class N extends L {
    L d;
    void f() { d.f(); }
}
class O extends N {
    void f() { d.f(); }
}
class P extends N {
    void f() { d.f(); }
}
```

---

Geben Sie das zugehörige OMT-Diagramm an:

Ist in dieser Struktur ein Entwurfsmuster enthalten? Wenn ja, welches?

ja  nein

Begründung:

.....

Gegeben seien der folgenden Java-Code

---

```
class X {  
    Object node;  
    X [] children;  
}
```

---

Geben Sie das zugehörige OMT-Diagramm an:

Ist in dieser Struktur ein Entwurfsmuster enthalten? Wenn ja, welches?

ja  nein

Begründung:

.....

Dieses ist eine rekursive Struktur. Damit eine rekursive Struktur brauchbar wird, ist immer ein Basisfall nötig, für den die Rekursion abbricht. Ist dieses eine brauchbare Struktur?

ja  nein

Begründung:

.....

---

## Aufgabe 2:

Entwickeln Sie ein Datenmodell in Form einer Abstrakten Syntax für die Verarbeitung einer einfachen Programmiersprache. In dieser Sprache soll ein Programm (*Program*) aus einer Anweisung (*Statement*) bestehen. Als Ausprägungen für Anweisungen gibt es Anweisungsfolgen (*Sequence*), Verzweigungen (*IfStatement*), Schleifen (*WhileStatement*) und Zuweisungen (*Assignment*). Eine Zuweisung besteht aus einer Variablen (*Variable*) als linker Seite und einem Ausdruck (*Expr*) als rechter Seite. Ein Ausdruck ist entweder eine Variable (*Variable*), eine Konstante (*Constant*) oder ein zweistelliger Ausdruck (*BinaryExpr*). Eine Variable besteht aus einer Zeichenreihe für den Variablennamen, eine Konstante aus einer ganzen Zahl. Als Operatoren (*Operator*) in einem zweistelligen Ausdruck sollen  $+$ ,  $-$ ,  $*$ , *div*,  $=$  und  $\neq$  erlaubt sein.

Verwenden Sie in dem Datenmodell pro Typdefinition nur einen Typkonstruktor. *String* sei ein vordefinierter Datentyp für Zeichenreihen. Verwenden Sie weiterhin die *kursiv* geschriebenen Namen als Datentypnamen.

Das Datenmodell als Abstrakte Syntax:

- 1) .....
- 2) .....
- 3) .....
- 4) .....
- 5) .....
- 6) .....
- 7) .....
- 8) .....
- 9) .....
- 10) .....
- 11) .....
- 12) .....
- 13) .....
- 14) .....
- 15) .....

Die Abstrakte Syntax transformiert in ein gleichwertiges OMT–Diagramm:

Welche Strukturmuster werden in diesem Datenmodell verwendet? Nennen Sie den jeweiligen Musternamen, die beteiligten Klassen und die beteiligten Referenzen.

1. Mustername, beteiligte Klassen und Referenzen

.....  
.....  
.....

2. Mustername, beteiligte Klassen und Referenzen

.....  
.....  
.....

3. Mustername, beteiligte Klassen und Referenzen

.....  
.....  
.....

4. Mustername, beteiligte Klassen und Referenzen

.....  
.....  
.....



**Aufgabe 3:**

Klassifizieren Sie das Fabrikmethoden-Muster.

.....  
.....

Welches ist der Zweck des Musters?

.....  
.....

Wo ist das Muster typischerweise anwendbar?

- 1) .....
- 2) .....
- 3) .....

Geben Sie das Klassendiagramm für das Fabrikmethoden-Muster an: