


Fachwerte in JWAM




kg


¢ % £

\$ m² €

¥ 31.12.99



Anforderungen



Workplace Solutions

- Konstruktion von Werten mit Klassen
- Garantie der Wertsemantik
- Geringer Speicherplatzverbrauch
- Elegante Anbindung an Benutzungsoberfläche

Übersicht

Definition:

- Fachwerte sind benutzerdefinierte Werte, die Werte im Anwendungsbereich modellieren.
- Sie verfügen über einen definierten Wertebereich und festgelegte Operationen. Die innere Repräsentation ist verborgen.

Wert vs. Objekt	
Werte <ul style="list-style-type: none">• haben keinen Lebenszyklus• abstrakt, ohne Identität, nur Gleichheit• definiert oder undefiniert, aber unveränderlich• keine gemeinsame Nutzung• referentiell transparent	Objekte <ul style="list-style-type: none">• haben einen Lebenszyklus• Repräsentationen von Objekten in Einsatzkontext• veränderbar bei Wahrung der Identität• gemeinsame Nutzung

Einschub: Flyweight

Fliegengewicht (für Fachwerte)

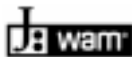


Problemfall 1: Speicherplatzverbrauch



Workplace Solutions

Wird jedesmal ein neuer Fachwert erzeugt, kann dies zu Speicherplatzproblemen führen.



Einschub: Flyweight-Muster



Workplace Solutions

Flyweights sind kleine, leichte, feinkörnige Objekte, die durch Object Sharing auch in großen Stückzahlen (>> 100.000) noch effizient eingesetzt werden können.

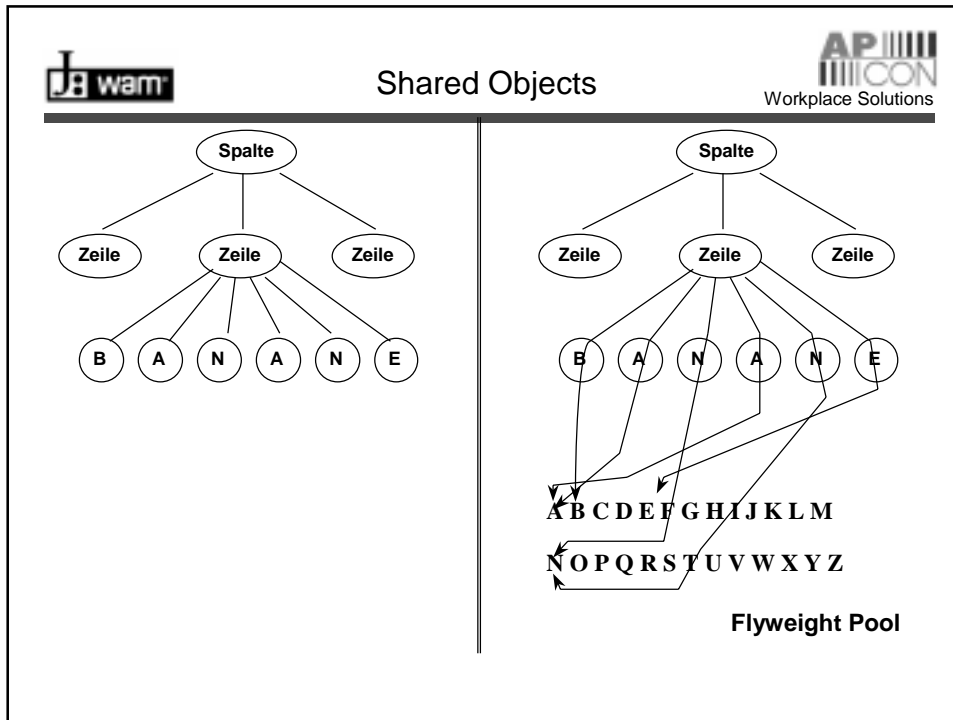
Buchstaben-Objekte

Beispiel:

Auch in objekt-orientierten Texteditoren werden die Buchstaben der dargestellten Texte häufig gar nicht als Objekte implementiert, da einfache Implementierungen die große Zahl von Buchstaben-Objekten nicht mehr effektiv verwalten könnten.

Zeilen-Objekte

Spalten-Objekte



DA wam **APCON**
Workplace Solutions

Innere und äußere Zustände

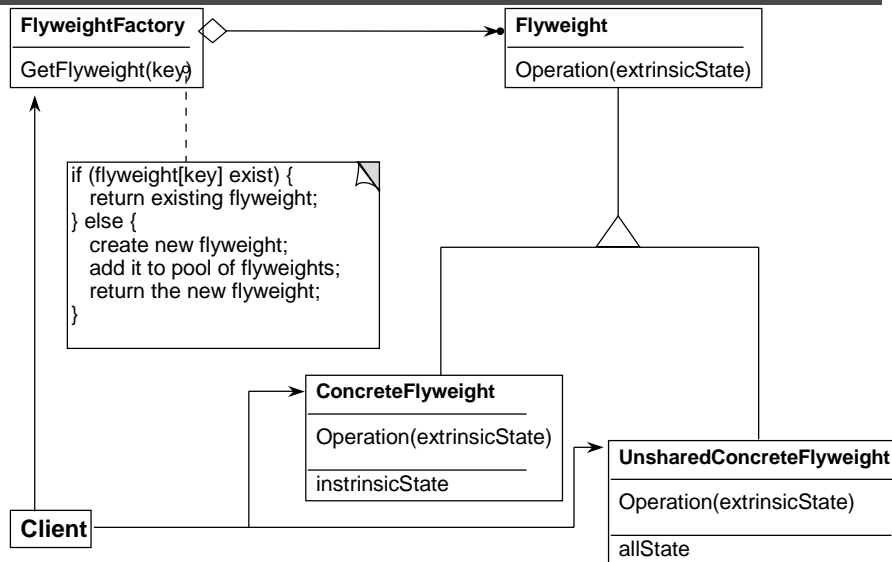
Damit ein Flyweight als Shared Object verwendet werden kann, darf es keine Annahmen über seinen Einsatzkontext machen.

<p>intrinsic state:</p> <p>Kontextfreie Informationen</p> <ul style="list-style-type: none">• wird vom Flyweight-Objekt gespeichert und verwaltet	<p>extrinsic state:</p> <p>Kontextabhängige Informationen</p> <ul style="list-style-type: none">• wird vom Klienten verwaltet• wird dem Flyweight-Objekt als Parameter übergeben
--	--

Shared Objects sollten immer über eine abstrakte Fabrik erzeugt werden, welche auch den Flyweight Pool verwaltet.

Das Flyweight-Muster findet eine sinnvolle Anwendung, wenn **alle** der folgenden Punkte zutreffen:

- die Anzahl der verwendeten Objekte ist groß
- hohe "Speicherkosten" aufgrund der großen Objektanzahl
- ein Großteil des Objektzustand kann nach außen verlagert werden (extrinsic)
- viele Gruppen von Objekten können durch wenige Shared Objects ersetzt werden
- die Klienten hängen nicht von der Objektidentität der Flyweight-Objekte ab



Prinzipiell sind Objekte veränderlich. Das bricht mit der Wertsemantik.

Realisierung:

- Fachwerte werden über Klassen realisiert.
- Es muß sichergestellt werden, daß sich diese Klassen gemäß Wertsemantik verhalten.
- Dies wird dadurch realisiert, daß Fachwertklassen keine verändernden Operationen haben.
- Unterscheidung zwischen Operationen an dem Wert (nur Funktionen) und Operationen an der Wertklasse.

Vorteile durch die Verwendung von Fachwerten in der IAK:

Da Fachwerte der Wertsemantik gehorchen und somit keine Seiteneffekte haben, können sie in der Interaktionskomponente beliebig verwendet werden, ohne die Funktionskomponente beim Zugriff auf das Material zu umgehen.

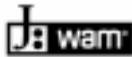
Integration von Fachwerten in der IAK:

1. Fachwerte werden bis zu den jeweiligen Interaktionsformen weitergegeben.
2. Dort werden sie über die Präsentation angezeigt.
3. Eingaben werden von der Präsentationsform bzw. der Interaktionsform durch Fabrikmethoden in die Erzeugung von Fachwerten umgesetzt.

- In `de.jwam.lang.domainvalue`
- Fachwert-Objekte werden durch Fachwertfabrik erzeugt.
- Fachwert und Fachwertfabrik werden jeweils durch eine Klasse realisiert. Fachwertklasse hat Prefix „dv“ (=DomainValue).
- Fachwertfabrik-Klasse ist Inner-Class der Fachwertklasse.
- Fachwert-Objekt weiß, von welcher Fabrik es erzeugt wurde.
- Erzeugte Fachwert-Objekte werden in Fachwert-Universum gehalten (DomainValueUniverse). Dieses wird als Teil der Arbeitsumgebung persistent gemacht.
- Flyweight-Pool wird in WeakHashMap verwaltet.

- `toString()` - liefert String-Repräsentation (menschenslesbar)
- `toStrings()` - liefert lange String-Repräsentation (menschenslesbar)
- `equals()` - sind zwei Fachwerte gleich? (Standardimplementation basierend auf „`toString()`“.)
- `hashCode()` - wertsemantische Standard-Implementation basierend auf „`toString()`“
- `isDefined()` - handelt es sich um einen definierten oder undefinierten Wert?
- `factory()` - liefert die erzeugende Fabrik

- `canCreateFromString()` - kann der Wert aus einer String-Repräsentation erzeugt werden?
- `value(String)` - liefert Fachwert aus String-Repräsentation
- `isValid(String)` - handelt es sich um eine gültige String-Repräsentation eines Fachwertes?
- `defaultValue()` - liefert einen Default-Fachwert
- `type()` - welchen Fachwerttyp erzeugt diese Fabrik?
- `unregisterValue(DomainValue)` - entferne Fachwert aus Fachwert-Universum

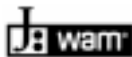


Standard-Fachwerte in JWAM



- dvString
- dvText
- dvEnumeration
- dvInteger
- dvBoolean
- dvIdentifier
- dvThingDescription
- dvTableOfContents

- ifFillInDomainValue
- ifFillInString
- ifFillInText
- ifFillInEnumeration
- ifFillInInteger
- ifFillInBoolean
- ifFillInThingDescription
- ifTableOfContentsSelection



Fachwert-Interaktionsformen



- Liegen unter
 - de.jwamx.technology.iafpf.interaction
 - de.jwamx.technology.iafpf.swingpf
- Zuordnung Fachwert zu Fachwert-Interaktionsform wird festgelegt in PFFactory (de.jwamx.technology.iafpf).