

27.04.2012	Advanced Software Engineering University of Vienna Forschungsgruppe Software Architecture	
<i>Kennzahl</i>	<i>Matrikelnummer</i>	<i>FAMILIENNAME</i> <i>Vorname</i>

12	10	6	15	7	4	6	Σ 60
1	2	3	4	5	6	7	

Bitte lesen Sie diesen Abschnitt. Sie haben **60** Minuten Zeit die Fragen zu beantworten.

Hinweise:

- Füllen Sie Kennzahl, Matrikelnummer, Familienname und Vorname zuerst aus.
 - Während der Prüfung sind keinerlei Unterlagen erlaubt!
 - Technische Hilfsmittel wie Übersetzungscomputer, Taschenrechner, Mobiltelefone, etc. sind nicht erlaubt!
 - Wenn Sie Probleme beim Verstehen einer Frage haben, fragen Sie.
 - Sie können die Fragen sowohl auf Deutsch als auch auf Englisch beantworten.
 - Schreiben Sie mit dokumentenechten Schreibutensilien (kein Bleistift).
 - Sie können den Appendix (letzte Seite) von der Prüfung abtrennen.
 - Viel Erfolg!
-

Task 1 / Aufgabe 1 (12 Points / Punkte)

Two important concepts in domain-driven development are Entities and Value Objects. Identify all Entities and Value Objects in the following domain model and explain why you identified a model element as Entity or Value Object.

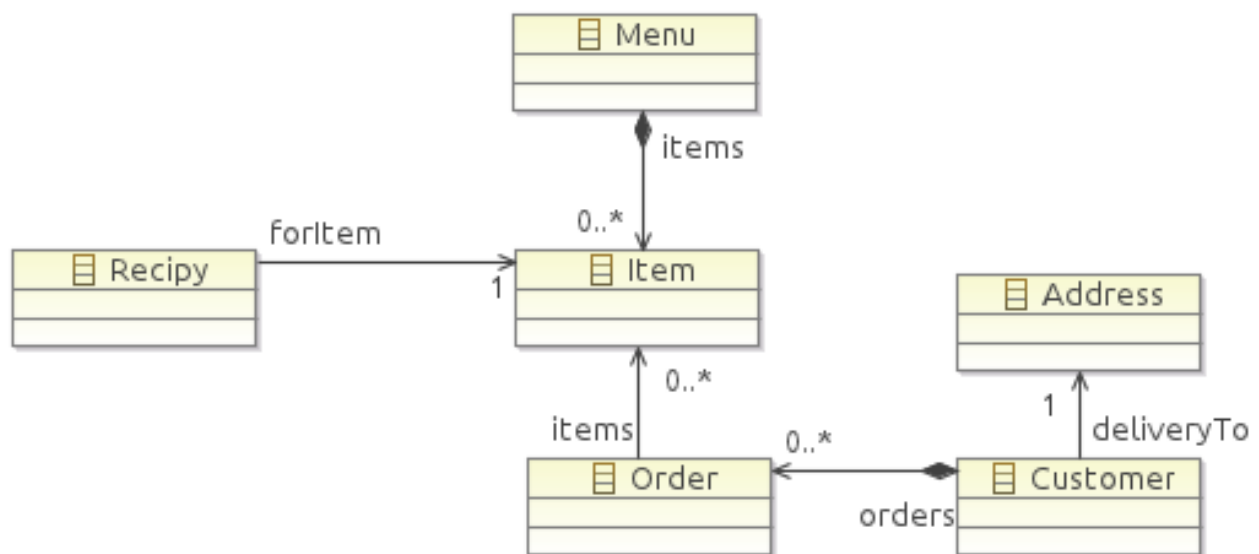
Mark each Entity with an “E” and each Value Object with a “V”

Zwei wichtige Konzepte im Domain-driven Development sind Entitäten und Value Objects.

Identifizieren Sie im folgenden Domänenmodell alle Entitäten und Value Objects und begründen Sie Ihre Entscheidungen.

Markieren Sie jede Entity mit einem „E“ und jedes Value Object mit einem „V“

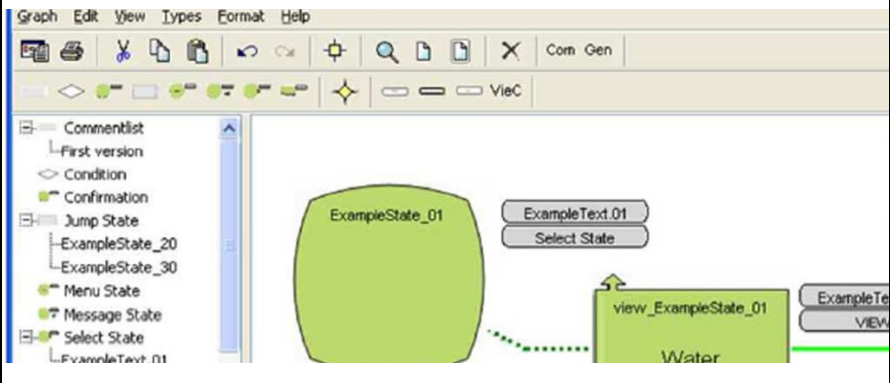
Pizza Delivery:



Task 2 / Aufgabe 2 (10 Points / Punkte)

Are the following code fragments examples for domain specific languages (DSLs)? Mark the appropriate cell with an X.

Sind die folgenden Codefragmente Beispiele für domain specific languages (DSLs)? Markieren Sie die zutreffende Zelle mit einem X.

Language	Example	DSL	NO DSL
C	<pre>printf ("Characters: %c %c \n", 'a', 65);</pre>		
C++	<pre>cout << "Output sentence";</pre>		
SQL	<pre>SELECT * FROM "users";</pre>		
RegExp	<pre>\b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}\b</pre>		
Groovy	<pre>def messageDigest = MessageDigest.getInstance("SHA1")</pre>		
Xtext	<pre>Type: DataType Entity;</pre>		
GraphViz	<pre>digraph G { subgraph cluster_0 { style=filled; color=lightgrey; node [style=filled, color=white]; a0 -> a1 -> a2 -> a3; label = "process #1"; } ... }</pre>		
UI application modeling			
Embedded System Control	<pre>Stateswitch linefollower State running Int32 light = 0; Light = ecrobot_get_light_sensor(SENSOR_PORT_T::NXT_PORTS1); ...</pre>		
Tcl	<pre>button .b -text "Push Me" -command {tk_messageBox -message "hello, world"} pack .b</pre>		

Task 3 / Aufgabe 3 (6 Points / Punkte)

What does "Ousterhout's Dichotomy" say about the relation of scripting and system programming languages? Explain Ousterhout's claim in detail and answer the following questions:

- What does "gluing" mean?
- Why is "typing" un-/important?

Was besagt "Ousterhouts Dichotomie" über das Verhältnis von Skript- und Systemprogrammiersprachen? Erläutern Sie Ousterhouts Behauptung im Detail und beantworten Sie die folgenden Fragen:

- Was bedeutet „gluing“?
- Warum ist „typing“ un-/wichtig?

Task 4 / Aufgabe4 (15 Points / Punkte)

Fill the gaps 2-6 in the code on the next page by inserting the correct letters (as you can see for gap number1) that are given to you in the table below. Be careful, some lines of code may not be used. The expected output of the script is printed below.

For your convenience, you will find the code fragments at the Appendix as well, which can be separated from this exam.

Füllen Sie die leeren Felder 2-6 im Code auf der nächsten Seite mit den richtigen Buchstaben (wie im Feld Nr. 1 zu sehen ist), welche in der Tabelle unten zu finden sind. Beachten Sie: einige Codezeilen werden nicht genutzt. Die erwartete Ausgabe des Scriptes finden Sie unten.

Zur leichteren Handhabung finden Sie die Code-Fragment im Anhang, den Sie von dieser Prüfung abtrennen können.

Expected output / Erwartete Ausgabe:

26.99 USD = 20.5048428 EUR = 16.750611131748 GBP = 26.98389448435548816 USD

Code fragments

A	<code>new Currency(26.99, CurrencyAbbrv. USD)</code>
B	<code>[value: 26.99, currency: CurrencyAbbrv. USD]</code>
C	<code>26.99.USD</code>
D	<code>currencyMap[CurrencyAbbrv.indexOf(delegate)][it]</code>
E	<code>currencyMap["\${delegate.currency}"]["\$it"]</code>
F	<code>currencyMap[delegate][it]</code>
G	<code>currencyMap[it][delegate]</code>
H	<code>{ [value: delegate, currency: it] }</code>
I	<code>new Currency(delegate, it)</code>
J	<code>{ -> new Currency(value: delegate, currency: it) }</code>
K	<code>CurrencyAbbrv. "\$delegate"</code>
L	<code>{ CurrencyAbbrv.it }</code>
M	<code>// create a new price</code>
N	<code>def Number value</code>
O	<code>return "\$value \$currency";</code>
P	<code>new Currency(delegate, it)</code>
Q	<code>newValue. "\$it"</code>
R	<code>[newValue, it]</code>
S	<code>this.value = newValue</code>
T	<code>EUR, USD, GBP</code>
U	<code>GRAM, KILOGRAM, TON</code>
V	<code>"get\$it"</code>
W	<code>it.metaClass. EUR</code>
X	<code>new Method(name: "getUSD")</code>
Y	<code>"get" + delegate</code>

```

enum CurrencyAbbrv {
  1 T
}

def currencyMap = [
  EUR : [EUR:1.00000, USD:1.31613, GBP:0.81691],
  USD : [EUR:0.75972, USD:1.00000, GBP:0.62068],
  GBP : [EUR:1.22387, USD:1.61092, GBP:1.00000]
]

class Currency {
  def Number value
  def CurrencyAbbrv currency

  public String toString() {
    return "$value $currency";
  }
}

CurrencyAbbrv.each {
  methodToAdd = 2
  createCurrency = 3
  Number.metaClass."$methodToAdd" = createCurrency

  Currency.metaClass."$methodToAdd" = { ->
    factor = 4
    newValue = delegate.value * factor
    return 5
  }
}

price = 6

inEUR = price.EUR
inGBP = inEUR.GBP
inUSD = inGBP.USD
print "$price = $inEUR = $inGBP = $inUSD"

```

Task 5 / Aufgabe 5 (7 Points / Punkte)

Describe a model-driven tool chain in software development and explain each of its items in a short sentence.

Beschreiben Sie eine „Model-driven Tool Chain“ in der Softwareentwicklung. Beschreiben Sie jedes Element der Tool Chain mit einem kurzen Satz.

Task 6 / Aufgabe6 (4 Points / Punkte)

When do you use a simple constructor (or simply "new") and when do you use a Factory to create instances?

In welchen Fällen wird ein Konstruktor zum Erstellen von Instanzen benutzt und in welchen Fällen eine Factory?

Task 7 / Aufgabe 7 (6 Points / Punkte)

Explain the difference between external and embedded DSLs! Which alternatives or variants are possible?

Erklären Sie den Unterschied zwischen externen und eingebetteten DSLs! Welche Alternativen oder Varianten sind möglich?

Appendix

Task 4

Code fragments

A	<code>new Currency(26.99, CurrencyAbbrev. USD)</code>
B	<code>[value: 26.99, currency: CurrencyAbbrev. USD]</code>
C	<code>26.99.USD</code>
D	<code>currencyMap[CurrencyAbbrev.indexOf(delegate)][it]</code>
E	<code>currencyMap["\${delegate.currency}"]["\$it"]</code>
F	<code>currencyMap[delegate][it]</code>
G	<code>currencyMap[it][delegate]</code>
H	<code>{ [value: delegate, currency: it] }</code>
I	<code>new Currency(delegate, it)</code>
J	<code>{ -> new Currency(value: delegate, currency: it) }</code>
K	<code>CurrencyAbbrev."\$delegate"</code>
L	<code>{ CurrencyAbbrev.it }</code>
M	<code>// create a new price</code>
N	<code>def Number value</code>
O	<code>return "\$value \$currency";</code>
P	<code>new Currency(delegate, it)</code>
Q	<code>newValue."\$it"</code>
R	<code>[newValue, it]</code>
S	<code>this.value = newValue</code>
T	<code>EUR, USD, GBP</code>
U	<code>GRAM, KILOGRAM, TON</code>
V	<code>"get\$it"</code>
W	<code>it.metaClass.EUR</code>
X	<code>new Method(name: "getUSD")</code>
Y	<code>"get" + delegate</code>

Expected output:

26.99 USD = 20.5048428 EUR = 16.750611131748 GBP = 26.98389448435548816 USD