

26.04.2013	<b>Advanced Software Engineering</b> <b>University of Vienna</b> <b>Forschungsgruppe Software Architecture</b>		
<i>Kennzahl</i>	<i>Matrikelnummer</i>	<i>FAMILIENNAME</i>	<i>Vorname</i>

						Σ 60
9	12	8	12	10	9	
1	2	3	4	5	6	

---

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 (9 Points / Punkte)**

Explain the term “Domain Specific Language”. What are DSLs used for? What are the characteristics of DSLs?

Erklären Sie den Begriff „Domain Specific Language“. Wofür werden DSLs verwendet? Welche Eigenschaften sind charakteristisch für DSLs?

## Task 2 / Aufgabe 2 (12 Points / Punkte)

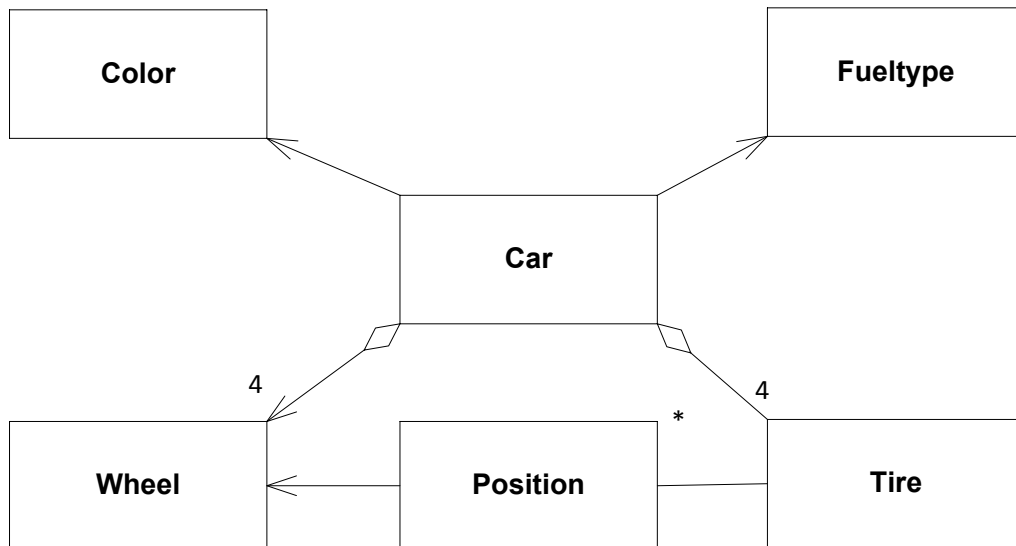
Two important concepts in domain-driven development are Entities and Value Objects. Identify all Entities and Value Objects in the following domain model and explain for **each element** why you identified it as Entity or Value Object.

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 für **jedes einzelne** Ihre Entscheidungen.

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



### Task 3 / Aufgabe 3 (8 Points / Punkte)

- a) Explain the term “model” in general!
  - b) Which models are used in Software Development? Give at least 3 examples for models used in Software Development and describe how they are used.
- 
- a) Erklären Sie den Begriff „Modell“ im Allgemeinen.
  - b) Welche Modelle werden in der Softwareentwicklung eingesetzt? Nennen und erklären Sie 3 Beispiele für Modelle, die in der Softwareentwicklung verwendet werden.

## Task 4 / Aufgabe 4 (12 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 number 1) 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-Fragmente im Anhang, den Sie von dieser Prüfung abtrennen können.

### Expected output / Erwartete Ausgabe:

```
failed: property 'age' not in range '18..65'.  
failed: property 'name' is null.  
Validation complete.
```

### Code fragments

A	<code>o = \$("age"   "name").checkConstraints()</code>
B	<code>o.metaClass.constraints = checkConstraints(Closure</code>
C	<code>o.constraints.setDelegate(this)</code>
D	<code>o.constraints = new Closure()</code>
E	<code>void checkConstraints(String name, Object args) {</code>
F	<code>Object invokeMethod(String name, Object args) {</code>
G	<code>Object checkConstraints(Closure constraints) { args</code>
H	<code>for (subject.constraints) { args[0] = it</code>
I	<code>it, delegate -&gt;</code>
J	<code>// validate each constraint</code>
K	<code>switch(val?.class) {</code>
L	<code>if (val != null) {</code>
M	<code>switch( it.value ) {</code>
N	<code>for ( args[0] ) {</code>
O	<code>switch( name ) {</code>
P	<code>if (it.key == "size" &amp;&amp;</code>
Q	<code>if (it.key == "size" &amp;&amp; it.value[0] &lt; val &lt;</code>
R	<code>if (it.key == "blank" &amp;&amp; it.value != null)</code>
S	<code>'\${it}'.\$"</code>
T	<code>int age</code>
U	<code>'\${it.value[0]}..\${it.value[1]}'.\$"</code>
V	<code>'\${it.value.inspect()}'.\$"</code>
W	<code>'\${it.value[size]}'.\$"</code>
X	<code>'\${it.value.call()}'.\$"</code>
Y	<code>format "C:"</code>

**class** Customer {

1 T

String name

**static constraints** = {

age(size:18..65)

name(size:3..20, blank:false)

}

}

**class** Validator {

**def** subject

**public void** validate(**def** o) {

subject = o

2

o.constraints.call()

println "Validation complete."

}

3

**def** val = subject.getProperty(name)

args[0].each {

4

**case** null: **if** (it.key == "blank" && !val)

println "failed: property '\${name}' is null."

**break**

**case** Integer :

**if** (it.key == "size" && !(it.value.contains(val)))

println "failed: property '\${name}' not in range

5

**break**

**case** String:

6

println "failed: property '\${name}' is too long or too short."

**break**

**default:** break;

}

}

}

}

customer = **new** Customer(age: 2, name:null)

validator = **new** Validator()

validator.validate(customer)

## Task 5 / Aufgabe 5 (10 Points / Punkte)

Create a valid instance of the XText grammar for a domain specific language that is described below. Use each grammar rule at least once. Use each arithmetic operator (+,-,\*,/) at least once. For your convenience, you will find the XText grammar at the Appendix as well, which can be separated from this exam.

Erstellen Sie eine Instanz der folgenden Grammatik einer domänenspezifischen Sprache. Verwenden Sie jede Grammatikregel mindestens einmal. Verwenden Sie jeden arithmetischen Operator (+,-,\*,/) mindestens einmal. Zur leichteren Handhabung finden Sie die Code-Fragment im Anhang, den Sie von dieser Prüfung abtrennen können.

```
grammar org.eclipse.xtext.example.arithmetics.Arithmetics with org.eclipse.xtext.common.Terminals
import "http://www.eclipse.org/emf/2002/Ecore" as ecore
generate arithmetics "http://www.eclipse.org/Xtext/example/Arithmetics"

Module:
    'module' name=ID
    (imports+=Import)*
    (statements+=Statement)*;

Import:
    'import' importedNamespace=ImportName;

ImportName:
    ID ('.' '*')?;

Statement:
    Definition | Evaluation;

Definition:
    'def' name=ID (('(' args+=DeclaredParameter (',' args+=DeclaredParameter)* ')')?
    ':' expr=Expression ';');

DeclaredParameter:
    name=ID;

AbstractDefinition:
    Definition | DeclaredParameter;

Evaluation:
    expression=Expression ';';

Expression:
    Addition;

Addition returns Expression:
    Multiplication (({Plus.left=current} '+' | {Minus.left=current} '-') right=Multiplication)*;

Multiplication returns Expression:
    PrimaryExpression (({Multi.left=current} '*' | {Div.left=current} '/') right=PrimaryExpression)*;

PrimaryExpression returns Expression:
    '(' Expression ')' |
    {NumberLiteral} value=NUMBER |
    {FunctionCall} func=[AbstractDefinition] (('(' args+=Expression (',' args+=Expression)* ')')?);

terminal NUMBER returns ecore::EBigDecimal:
    ('0'..'9')* ('.' ('0'..'9')+)?;

terminal INT returns ecore::EInt:
    'this one has been deactivated';
```





## Task 6 / Aufgabe 6 (9 Points / Punkte)

- a) Explain the differences between dynamically and statically typed languages!
  - b) Are languages of one of these type more reliable than the other? Explain your answer!
- 
- a) Erklären Sie die Unterschiede zwischen Sprachen mit dynamischer Typisierung und Sprachen mit statischer Typisierung!
  - b) Sind Sprachen eines der beiden Typen robuster als Sprachen des anderen Typs? Erklären Sie Ihre Entscheidung!

## Appendix

### Task 4

#### Code fragments

A	o = \$("age"   "name").checkConstraints()
B	o.metaClass.constraints = checkConstraints(Closure
C	o.constraints.setDelegate(this)
D	o.constraints = new Closure()
E	void checkConstraints(String name, Object args) {
F	Object invokeMethod(String name, Object args) {
G	Object checkConstraints(Closure constraints) { args
H	for (subject.constraints) { args[0] = it
I	it, delegate ->
J	// validate each constraint
K	switch(val?.class) {
L	if (val != null) {
M	switch( it.value ) {
N	for ( args[0] ) {
O	switch( name ) {
P	if (it.key == "size" &&
Q	if (it.key == "size" && it.value[0] < val <
R	if (it.key == "blank" && it.value != null)
S	'\${it}':.
T	int age
U	'\${it.value[0]}..\${it.value[1]}':.
V	'\${it.value.inspect()}':.
W	'\${it.value[size]}':.
X	'\${it.value.call()}':.
Y	format "C:"

#### Expected output / Erwartete Ausgabe:

failed: property 'age' not in range '18..65'.  
failed: property 'name' is null.  
Validation complete.

## Task 5

### XText Grammar

```
grammar org.eclipse.xtext.example.arithmetics.Arithmetics with org.eclipse.xtext.common.Terminals
import "http://www.eclipse.org/emf/2002/Ecore" as ecore
generate arithmetics "http://www.eclipse.org/Xtext/example/Arithmetics"

Module:
  'module' name=ID
  (imports+=Import)*
  (statements+=Statement)*;

Import:
  'import' importedNamespace=ImportName;

ImportName:
  ID ('.' '*')?;

Statement:
  Definition | Evaluation;

Definition:
  'def' name=ID (('(' args+=DeclaredParameter (',' args+=DeclaredParameter)* ')')?
  ':' expr=Expression ';');

DeclaredParameter:
  name=ID;

AbstractDefinition:
  Definition | DeclaredParameter;

Evaluation:
  expression=Expression ';';

Expression:
  Addition;

Addition returns Expression:
  Multiplication (({Plus.left=current} '+' | {Minus.left=current} '-') right=Multiplication)*;

Multiplication returns Expression:
  PrimaryExpression (({Multi.left=current} '*' | {Div.left=current} '/') right=PrimaryExpression)*;

PrimaryExpression returns Expression:
  '(' Expression ')' |
  {NumberLiteral} value=NUMBER |
  {FunctionCall} func=[AbstractDefinition] (('(' args+=Expression (',' args+=Expression)* ')')?);

terminal NUMBER returns ecore::EBigDecimal:
  ('0'..'9')* ('.' ('0'..'9')+)?;

terminal INT returns ecore::EInt:
  'this one has been deactivated';
```