

30.01.2011	Advanced Software Engineering University of Vienna Forschungsgruppe Software Architecture	
<i>Kennzahl</i>	<i>Matrikelnummer</i>	<i>FAMILIENNAME</i> <i>Vorname</i>

12	14	6	15	7	6
1	2	3	4	5	6

Σ 60

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!
-

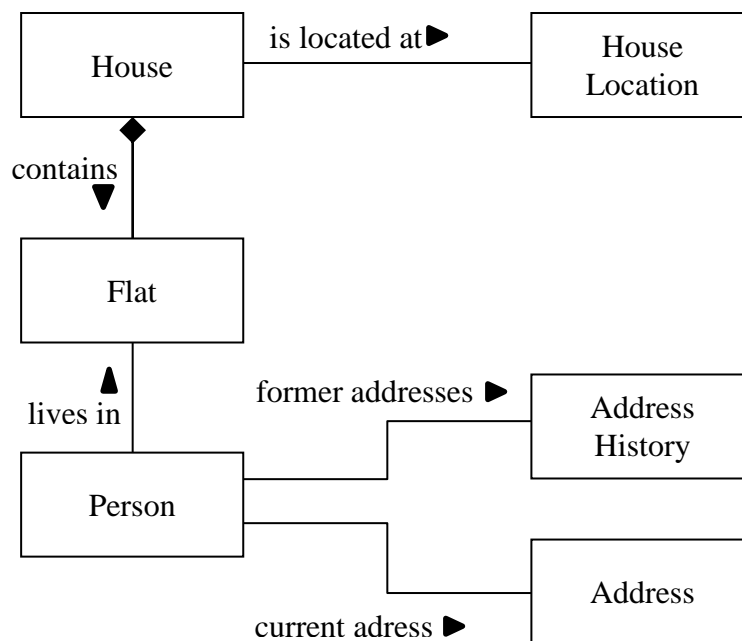
Task 1 / Aufgabe 1 (12 Points / Punkte)

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“



Task 2 / Aufgabe 2 (14 Points / Punkte)

Create a valid instance of the grammar for a domain specific language that is described below. Use each grammar rule at least once.

Erstellen Sie eine Instanz der folgenden Grammatik einer domänenspezifischen Sprache. Verwenden Sie jede Grammatikregel mindestens einmal.

```
DinkyCarModel:
    room=Room
    (commands+=Command)+ ;

Command:
    (DirectionCommand | AccelerationCommand)
    ('.'|'!') ;

Room:
    'The room '
    name=EString
    'has a measure of'
    x=EInt
    ' mm to '
    y=EInt
    ' mm.'
    'In the middle of the room there you can see a'
    vehicles+=Vehicle ((' , a ' vehicles+=Vehicle)* (' , and a ' vehicles+=Vehicle))?
    '.' ;

DirectionCommand :
    movement=Movement
    'the'
    vehicle=Vehicle
    distance=EInt
    'mm to the'
    direction=Direction ;

AccelerationCommand:
    PositiveAcceleration | NegativeAcceleration ;

PositiveAcceleration:
    'speed up to'
    speed = EDouble
    'miles per hour' ;

NegativeAcceleration:
    'break speed to '
    speed=EDouble
    'miles per hour' ;

enum Vehicle: Car = 'Car' | Truck = 'Truck' | Moped = 'Moped';
enum Movement: drive = 'drive' | move = 'move' | drift = 'drift';
enum Direction: left = 'left' | right = 'right';

EDouble returns ecore::EDouble: '-'? INT? '.' INT (('E'|'e') '-'? INT)?;
EString returns ecore::EString: STRING | ID;
EInt returns ecore::EInt: '-'? INT;
terminal INT returns ecore::EInt: ('0'..'9')+ ;
terminal STRING :
    '"' ( '\\' ('b'|'t'|'n'|'f'|'r'|'"'|"'"|'\\') | !('\\'|'"') ) * '"' |
    "'" ( '\\' ('b'|'t'|'n'|'f'|'r'|'"'|"'"|'\\') | !('\\'|'"') ) * "'";
```


Task 3 / Aufgabe 3 (6 Points / Punkte)

What are the characteristics of dynamically typed languages? Name at least 3 characteristics and give a short code example for each.

Wie lauten die Charakteristiken (Kennzeichen) von dynamisch getypten Sprachen? Nennen Sie mindestens 3 Charakteristiken und geben Sie jeweils ein kurzes Code-Beispiel.

Task 4 / Aufgabe 4 (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 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-Fragment im Anhang, den Sie von dieser Prüfung abtrennen können.

Expected output / Erwartete Ausgabe:

Kevin has to walk 252 METERS

Hi Kevin!

Code fragments

A	<code>Number.metaClass.getMeters</code>
B	<code>NumberCategory</code>
C	<code>DistanceFactory.create</code>
D	<code>def distanceClosure</code>
E	<code>new Kevin()</code>
F	<code>"Kevin"</code>
G	<code>Person: "Kevin"</code>
H	<code>[name: "Kevin"]</code>
I	<code>new Person(name: "Maria")</code>
J	<code>new Person(*maria)</code>
K	<code>[new Person("Maria")]</code>
L	<code>"Maria".new</code>
M	<code>// a new scope</code>
N	<code>use(NumberCategory)</code>
O	<code>NumberCategory.each</code>
P	<code>maria.doesSomethingTo kevin</code>
Q	<code>maria."\$doesSomethingTo" kevin</code>
R	<code>maria.doesSomethingTo { kevin }</code>
S	<code>"\$doesSomethingTo"(maria, kevin)</code>
T	<code>MILLIMETERS, CENTIMETERS, METERS, DECIMETERS, KILOMETERS</code>
U	<code>GRAM, KILOGRAM, TON</code>

```

final class Distance {
  def Number number
  def Unit unit
  String toString() { "${number} ${unit}" }
}

enum Unit {
  1 T
}

2 = { -> new Distance(number: delegate, unit: Unit.METERS) }

class NumberCategory {
  static Distance and(Number lhs, Distance rhs) {
    new Distance(number: lhs+rhs.number, unit: rhs.unit)
  }
}

class Person {
  def name
  def goes(distance) {
    println "${this} has to walk ${distance}"
  }
  def greets(someone) {
    println "Hi ${someone}!"
  }
  String toString() { "${name}" }
}

def doesSomethingTo = "greets"
Person kevin, maria

3 {
  kevin = 4
  maria = 5
  shopping = 200.and(52.meters)
}
kevin.goes shopping
6

```

Task 5 / Aufgabe 5 (7 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 6 / Aufgabe 6 (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 Dichtonomie" 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?



Appendix

Task 4

Code fragments

A	<code>Number.metaClass.getMeters</code>
B	<code>NumberCategory</code>
C	<code>DistanceFactory.create</code>
D	<code>def distanceClosure</code>
E	<code>new Kevin()</code>
F	<code>"Kevin"</code>
G	<code>Person: "Kevin"</code>
H	<code>[name: "Kevin"]</code>
I	<code>new Person(name: "Maria")</code>
J	<code>new Person(*maria)</code>
K	<code>[new Person("Maria")]</code>
L	<code>"Maria".new</code>
M	<code>// a new scope</code>
N	<code>use(NumberCategory)</code>
O	<code>NumberCategory.each</code>
P	<code>maria.doesSomethingTo kevin</code>
Q	<code>maria."\$doesSomethingTo" kevin</code>
R	<code>maria.doesSomethingTo { kevin }</code>
S	<code>"\$doesSomethingTo"(maria, kevin)</code>
T	<code>MILLIMETERS, CENTIMETERS, METERS, DECIMETERS, KILOMETERS</code>
U	<code>GRAM, KILOGRAM, TON</code>

Expected output:

Kevin has to walk 252 METERS

Hi Kevin!