15.03.2013	**Advanced Software Engineering** **University of Vienna** **Forschungsgruppe Software Architecture**	
Kennzahl	*Matrikelnummer*	*FAMILIENNAME* *Vorname*

6	6	9	15	10	6	8	∑ 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 (6 Points / Punkte)

Explain the differences between an Entity Factory and a Value Object Factory!

Erklären Sie die Unterschiede zwischen einer Entity Factory und Value Object Factory!

Task 2 / Aufgabe 2 (6 Points / Punkte)

Define and explain the terms **Service**, **Repository**, and **Aggregate** in the context of Model Driven Development!

Definieren und erklären Sie die Begriffe **Service**, **Repository** und **Aggregate** im Kontext von Modell-getriebener Entwicklung!

Task 3 / Aufgabe 3 (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 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.

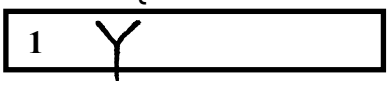
Expected output / Erwartete Ausgabe:

```
Kevin has to walk 252 METERS
Maria: Hi Kevin!
```

Code fragments

A	`Number.metaClass.getMeters`
B	`NumberCategory`
C	`DistanceFactory.create`
D	`def distanceClosure`
E	`new Kevin()`
F	`"Kevin"`
G	`Person: "Kevin"`
H	`[name: "Kevin"] as Person`
I	`new Person(name: "Maria")`
J	`new Person(*maria)`
K	`[new Person("Maria")]`
L	`"Maria".new`
M	`Person.createInstance("Maria")`
N	`// a new scope`
O	`def "goes shopping" =`
P	`def and = {number -> delegate + number }`
Q	`use(NumberCategory)`
R	`NumberCategory.each`
S	`Number.expando.setMethod(and, ["left","right"], { return left + right }`
T	`class ShoppingPerson`
U	`maria.doesSomethingTo kevin`
V	`maria."$doesSomethingTo" kevin`
W	`maria.doesSomethingTo { kevin }`
X	`"$doesSomethingTo"(maria, kevin)`
Y	`MILLIMETERS, CENTIMETERS, METERS, DECIMETERS, KILOMETERS`
Z	`GRAM, KILOGRAM, TON`

```
class Person {}

enum Unit {
```



```
}

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

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

}
```


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

Person.metaClass.name = new String()
Person.metaClass.goes =
            { distance -> println "${name} has to walk ${distance}" }
Person.metaClass.greets =
                { someone -> println "${name}: Hi ${someone.name}!" }

def doesSomethingTo = "greets"
```


```
{
      shopping = 200.and 52.meters
}

kevin =
```



```
maria =
```



```
kevin.goes shopping
```

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. Verwendne 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 (6 Points / Punkte)

What are the criteria that have to be considered when choosing between a scripting language and a system programming language for a task at hand? Explain, why each criteria will cause the use of a scripting language or a system programming language.

Welche Kriterien sind zu beachten, wenn Sie sich zur Entwicklung eines Programmes zwischen einer Skriptsprache und einer Systemprogrammiersprache entscheiden müssen? Erklären Sie für jedes Kriterium, warum dies zum Einsatz einer Skriptsprache oder einer Systemprogrammiersprache führt.

Task 7 / Aufgabe 7 (8 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 Unterschwiede 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	`Number.metaClass.getMeters`
B	`NumberCategory`
C	`DistanceFactory.create`
D	`def distanceClosure`
E	`new Kevin()`
F	`"Kevin"`
G	`Person: "Kevin"`
H	`[name: "Kevin"] as Person`
I	`new Person(name: "Maria")`
J	`new Person(*maria)`
K	`[new Person("Maria")]`
L	`"Maria".new`
M	`Person.createInstance("Maria")`
N	`// a new scope`
O	`def "goes shopping" =`
P	`def and = {number -> delegate + number }`
Q	`use(NumberCategory)`
R	`NumberCategory.each`
S	`Number.expando.setMethod(and, ["left","right"], { return left + right }`
T	`class ShoppingPerson`
U	`maria.doesSomethingTo kevin`
V	`maria."$doesSomethingTo" kevin`
W	`maria.doesSomethingTo { kevin }`
X	`"$doesSomethingTo"(maria, kevin)`
Y	`MILLIMETERS, CENTIMETERS, METERS, DECIMETERS, KILOMETERS`
Z	`GRAM, KILOGRAM, TON`

Expected output / ErwarteteAusgabe:

```
Kevin has to walk 252 METERS
Maria: Hi Kevin!
```

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';
```