

xx.xx.201x	Advanced Software Engineering University of Vienna Forschungsgruppe Software Architecture	
<i>Kennzahl</i>	<i>Matrikelnummer</i>	<i>FAMILIENNAME</i> <i>Vorname</i>

10	15	6	10	10	9
1	2	3	4	5	6

Σ 60

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

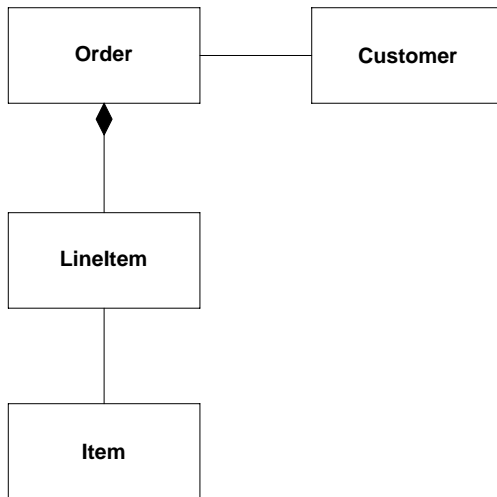
Hinweise:

- 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).
 - Viel Erfolg!
-

Task 1 / Aufgabe 1 (10 Points / Punkte)

Two very 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.

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.



Task 2 / Aufgabe 2 (15 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.

```
FastFoodRestaurant:
    'Restaurant'
    '{'
        name = STRING
        address = (RestaurantAddress)?
        'sells' '(' sellsProducts += (Product) ("," sellsProducts += (Product))* ')'
        storage = StorageFacility
    '}'

Product:
SimpleProduct | PackageProduct;

SimpleProduct:
    'Product'
    '{'
        'id' id = ID
        'name' name = STRING
        'category' category = STRING
        'foodValue' foodValue = INT
        'price' price = DOUBLE

        'isMadeOf' '(' isMadeOf += ([Ingredient]) ("," isMadeOf += ([Ingredient]))* ')'
    '}'

Ingredient:
    'Ingredient'
    '{'
        'name' name = STRING
        'price' price = DOUBLE
    '}'

PackageProduct:
    'Package'
    '{'
        'id' id = ID
        'name' name = STRING
        'price' price = DOUBLE
        //'components' components = STRING
        'consists of' '('
            consistsOf += ([Product]) ("," consistsOf += ([Product]))+
        ')'
    '}'

StorageFacility:
    'StorageFacility'
    '{'
        'maximumCapacity' capacity = INT 'cubic m.'
        'contains' '{'
            stores += (Ingredient)*
        '}'
    '}'

RestaurantAddress:
    'country' country = STRING
    'postalCode' zip = INT
    'city' city = STRING
    'street' street = STRING
    'number' number = INT;

DOUBLE:
    '-'? INT '.'? INT?
;
terminal ID :
    '^'?('a'..'z'|'A'..'Z'|'_') ('a'..'z'|'A'..'Z'|'_'|'0'..'9')* ;
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)

Name at least 3 advantages and 3 disadvantages of Dynamic Typing.

Nennen Sie 3 Vorteile und 3 Nachteile von dynamischer Typisierung.

Task 4 / Aufgabe 4 (10 Points / Punkte)

Fill these gaps (next page) by inserting the correct letters (as you can see for number 5) 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.

Füllen Sie die leeren Felder (auf der nächsten Seite) mit den richtigen Buchstaben (wie im Feld Nr. 5 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.

Expected output / Erwartete Ausgabe:

```
Order[ products=[hamburger, salad]; specialGift=coke; totalSum=12]
Order[ products=[cheeseburger, fries, coke]; specialGift=; totalSum=0]
[cheeseburger, fries]
```

Code fragments

A	<code>println products.findAll(filter)</code>
B	<code>println filter</code>
C	<code>filter.println</code>
D	<code>products.join(filter)</code>
E	<code>iWant = new Order("hamburger", "salad", "coke", 12)</code>
F	<code>iWant = ["hamburger", "salad"]</code>
G	<code>iWant = "hamburger", "salad"</code>
H	<code>firstOrder = new Order(*firstOrder)</code>
I	<code>firstOrder = new Order(firstOrder)</code>
J	<code>firstOrder = [new Order()]</code>
K	<code>for(OrderCategory)</code>
L	<code>// a new scope</code>
M	<code>use(OrderCategory)</code>
N	<code>OrderCategory.each</code>
O	<code>burgerFilter = { it != myGift }</code>
P	<code>burgerFilter(it = new Order())</code>
Q	<code>burgerFilter = "coke"</code>
R	<code>secondOrder burgerFilter</code>
S	<code>secondOrder.command burgerFilter</code>
T	<code>secondOrder.command { burgerFilter }</code>
U	<code>secondOrder."\$command" burgerFilter</code>


```

class Order {
    def products
    def specialGift
    def totalSum

    Order(products, specialGift, totalSum) {
        this.products = products
        this.specialGift = specialGift
        this.totalSum = totalSum
    }

    def list(Closure filter) {

    }

    @Override
    public String toString() {
        return "Order[ "+
            "products="+products+"; "+
            "specialGift="+specialGift+"; " +
            "totalSum="+totalSum+"]";
    }
}
class OrderCategory {
    static Order plus(String s1, String s2) {
        return new Order([s1,s2], "", 0)
    }
    static Order plus(Order order, String s2) {
        order.products.add(s2)
        return order
    }
}

```

1

2

```

myGift = "coke"
iPay = 12
firstOrder = [iWant, myGift, iPay]

```

3

```

println firstOrder
def secondOrder = "salad"

```

4

```

{
    secondOrder = "cheeseburger" + "fries" + "coke"
}
println secondOrder

```

5

0

```

command = "list"

```

6

Task 5 / Aufgabe 5 (10 Points / Punkte)

Explain the term “model” in general! What is a model in Software Development? Give at least 3 examples for models used in Software Development and describe how they are used.

Erklären Sie den Begriff „Modell“ im Allgemeinen. Was sind Modelle im Kontext der Softwareentwicklung? Nennen und erklären Sie 3 Beispiele für Modelle, die in der Softwareentwicklung verwendet werden.

Task 6 / Aufgabe 6 (9 Points / Punkte)

Give at least 3 application areas for scripting languages and name at least 2 characteristics that make scripting languages a good choice for each of the application areas.

Nennen Sie mindestens 3 typische Anwendungsgebiete für Scriptsprachen und nennen Sie je Anwendungsgebiet mindestens zwei Eigenschaften, die Skriptsprachen dafür prädestinieren!