Exercises Lesson DP-04: Midterm Homework

Name:

\_\_\_\_Registration-Nb.:\_\_\_\_\_

# Task 1 Software Development Processes

1.a Describe the Waterfall Model and discuss for what kind of projects it is suitable and for what kind of projects problems with this model are expected [3 points]

Suitable for small projects with very clear requirements
 Example: New version of an already established program
 Problems expected with complicated and big projects. Additionally projects covering a complete new and challenging idea, where requirements are not very clear and changes are expected

#### 1.b What is the important characteristic of object oriented process models? [2 points]

- They use the incremental and iterative approach
- Additionally they use the same modeling language for both object oriented analysis and object oriented design, typically UML.

#### 1.c Describe the tasks of the customer and developers concerning the "User stories" in Agile Methods during the "Planning Game"

[2 points]

- Customer
  - Writing story cards
  - Overworking, or splitting up story cards depending on feedback of Developer
  - Choosing which story cards to be implemented, depending on duration of time box, implementation time developers could offer, and guessed duration of implementation of a story card
- Developer
  - Understanding story cards, clarification of issues with Customer
  - Guessing needed time for realization of a story card



# Task 2 Requirements Analysis

# 2.a Check, if the requirements are functional or non functional (required constraints)

#### [3 points]

Requirement	functional	Non functional
The user enters his account number	Х	
The administrator edits customer data	Х	
The system should process 10,000 queries per hour		Х
The customer would like to know, if a car in his wished category is available	X	
The customer adapts his User Interface		X <sup>1)</sup>
The system should run on Windows XP and on Linux		Х

<sup>1)</sup> Depending on the specification it could be a functional requirement as well, if the GUI is a critical success factor, e. g. in a webshop, or a GUI focused application

#### 2.b Are <<include>>- and <<extend>>-Relationships in Use-Case-Diagrams necessary? Explain your answer. [2 points]

The <<include>>- Relationship helps to save redundancy, the <<extend>>-Relationships shows optional possibilities.

So, both relationships help to clearly arrange a Use-Case-Diagram, but they are not necessary.



## Proposal Version 2.0

# Task 3 UML Diagrams

### 3.a Add the missing associations



# 3.b Create an object diagram representing the following facts [2 points]

An object **b** of the class **M** is associated with an object **f** of the class **W** and is associated with an object **r** of the class **W**.





## [3 points]

# Task 4 Design Pattern – General

4.a Which of the following statements describe the characteristics of Design Pattern (more than one answer could be selected)?		
1.	Design Pattern try "to separate the things that change from the things that remain the same."	$\checkmark$
2.	Design Pattern represent a complete idea within a program.	
3.	A Design Pattern is an effective implementation of a problem.	0
4.	A Design Pattern can encapsulate actions or processes.	

#### 4.b Choose four Design Pattern and assign them to the appropriate category: [2 points]

Creational	Structural	Behavioural
<ul> <li>Abstract Factory</li> <li>Builder</li> <li>Factory Method</li> <li>Prototype</li> <li>Singleton</li> </ul>	<ul> <li>Adapter</li> <li>Bridge</li> <li>Composite</li> <li>Decorator</li> <li>Façade</li> <li>Flyweight</li> <li>Proxy</li> </ul>	<ul> <li>Chain of Responsibility</li> <li>Command</li> <li>Interpreter</li> <li>Iterator</li> <li>Mediator</li> <li>Memento</li> <li>Observer</li> <li>State</li> <li>Strategy</li> <li>Template Method</li> <li>Visitor</li> </ul>

# 4.c Describe the intent and the motivation for the Design Pattern "Abstract Factory"

[2 points]

Provide an interface for creating families of related or dependent objects without specifying their concrete class.

It is helpful e. g. to handle different operation systems for GUI widgets or different factories of specified objects.



#### Task 5 **Design Pattern – Concrete**

Following class diagram is given:



#### 5.a Which Design Pattern is used?

# [1 Point]

## Composite

What is the main difference between the implementations of getSize() in 5.b classes Room and Apartment? Write corresponding pseudo code. [4 points]

```
for Apartment:
```

getsize() calculates size of Apartment plus all the sizes of the components - recursively

```
getsize() {
           size = size + getsize(children);
           return size:
}
```

for Leaf: getsize() returns the size of the leaf

```
getsize() { return size;}
```

```
Extend the diagram above with a class "Kitchen".
                                                                           [1 point]
5.c
```

In the class diagram, indicate which component takes which role in the design 5.d pattern. Use the generic pattern names for roles.

[2 points]

