Frank Hoffmann
your name:_____ your person number:_____-_____

Exam in
2D1359 & 2D1360 Objektorienterad modellering programmering och analys
Thursday, 22nd August 2002, 14:00-19:00

No study aids allowed.
Points and grading:
Grade 3, 4, and 5 require 20, 24 and 28 out of 36 points

Theoretical part:

Question 1: (3) Explain the difference between composition and generalization. Give two examples of objects/classes that are related through composition on the one hand and generalization on the other hand. Draw the corresponding UML class diagrams.

Composition is a stronger form of aggregation in which the whole is composed of parts and the parts are necessary to the whole (example processor is composed of CPU, memory, I/O port etc). Composition denotes a “has-a” or “is built of” kind of relationship. Generalisation is a relationship defined at the class level which means that all objects must obey this relationship. Generalization denotes a “is a kind of” relationship. (example car, bus, motor cycle is a kind of vehicle). See also lecture slides for lecture 1 and pages 396ff and 414ff in Larman

Question 2: (3) What is the difference between “call by reference” and “call by value”. Assume, a function **f()** obtains an integer parameter **x**, which value it increments by one. Specify the C++ code of this function and the function call for the case when the parameter **x** is
a.) passed by value
b.) passed by reference

In a call by value the function receives a copy of the parameter and the original variable is not affected by changes made to the local copy. In a call by reference the program passes the address of the original variable which could then be altered within the function. Call by reference therefore has a side effect on the calling routine whereas call by value does not not.

```
void f1(int x)   //call by value
{ x++; }

void f2(int& x) //call by reference
 { x++;}

int x=3;
```

your name:_____ your person number:_____-_____
f1(x); // x still has value 3
f2(x); //x has new value 4

see also lecture slides to lecture 15, slides 33 & 34

Question 3: (2) What is (usually) the multiplicity of the following associations?

a. King is ruler of country
b. Wheel is part of vehicle
c. Nephew is relative of uncle
d. Tourist visits city

Answer:
a: 0-1 but 0-1-* is acceptable (see queen Elisabeth),
b:*-1 or 0,4-1 or 2,4-1,
c: *-*, someone can have an aunt but not uncle, an uncle can have a niece but no nephew
,d: *-*

Question 4: (3) What is the difference between static and dynamic binding? When does dynamic binding occur in OO programming? Give a code example of dynamic binding in JAVA.

In static binding the method that is called is linked to the variable/object at compile time, in dynamic binding it is determined during run time, when the concrete object class is known. Dynamic binding is related to polymorphism, namely that subclasses can override or implement (abstract super class) a method declared in their superclass. For code examples see lecture slides lecture 3 slides 40ff.

Question 5: (2) What is the objective of object oriented design? What artifacts belong to the design model? What is the single most crucial task in object design?

Object oriented design is concerned with defining software objects and how they collaborate to fulfill the requirements (do the thing right). The single most crucial task in OOD is assigning responsibilities to objects. Artifacts in the design model are use-case realizations in form of interaction diagrams, sequence diagrams and design class diagrams see also lecture 1 slides 15ff and Larman chapter 1 + 17,18 and 19.

Question 6: (2) What happens during requirement analysis and what are its objectives? Name different types of requirements.

Types of requirements: FURPS+ , functional, usability, reliability, performance, supportability,

Requirement analysis is concerned with identifying the functional requirements of the problem and to describe them as **use cases**. Requirement analysis includes an

investigation of the problem domain. The goal is to find out what needs to be done (do the right thing). See also Larman chapters 1, 5,6,7

Question 7: (2) What are the key concepts of Extreme Programming (XP)?

-Refactoring
-Coding in pairs of two
-Extensive, automatic testing (write tests first)
-iterative development
-continuous integration, short iteration cycles, small releases
-simple design
-coding standards
see also lecture notes lecture 2

Question 8: (5) Describe the Composite pattern. What is the intent of the Composite pattern and what problem does it solve, and how does it solve the problem? Give an example of a design in which the Composite pattern is used. For this example draw the UML design class diagram that contains the classes, associations and main methods.

The intent of the composite pattern is to compose objects into tree structures to represent part-whole hierarchies. Composite lets clients treat individuals objects and compositions of objects uniformly. The composite pattern describes how to use recursive composition so that clients do not have to distinguish between primitive and container objects. An example would be a graphics system in which primitive graphic objects such as lines, rectangles, triangles and text can be grouped into composed objects. There is a common base class Graphic from which subclasses for primitives and composites are derived. The abstract base class defines basic graphical operations for drawing, moving, removing etc. The Composite class aggregates graphic objects and delegates the methods for drawing to its constituents. For more details see lecture slides lecture 9 slides 10 ff. and Larman pages 358 ff.

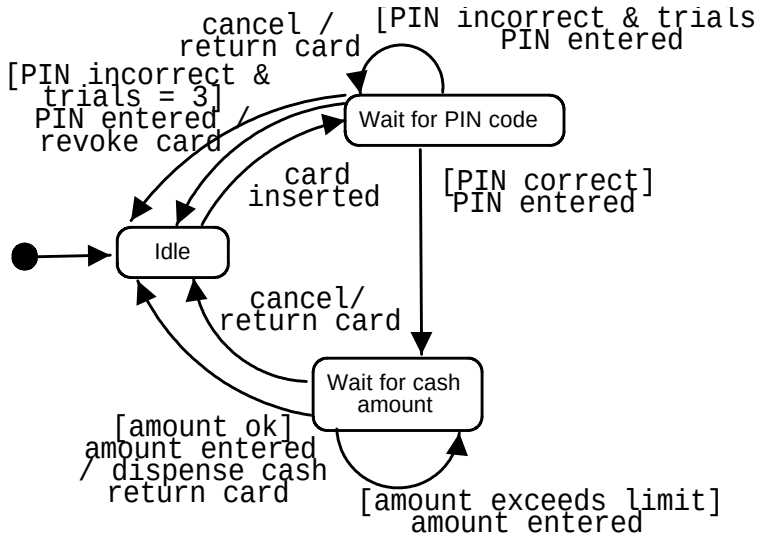
Practical Part:

Question 9: (8) A cash machine (bankautomat) dispenses cash in form of bank notes to an authorized customer who identifies herself with a banking card (bankkort) and an authorization pin number. The customer starts the transaction by inserting her card. The customer is asked to enter her pin number for identification. If an incorrect pin number was entered the customer can enter the pin code again. If the customer enters a wrong pin code three times, the transaction is cancelled and the banking card is withdrawn. If the correct pin number was entered the customer can specify the amount of cash she would like to withdraw. The system verifies that the customer is authorized to withdraw the requested amount of money (requested amount does not exceed her account balance). If authorization of the transaction was successful the machine returns the card and dispenses the cash. If authorization failed, the user can

enter a new amount. At any point the user can cancel the transaction in which case her banking card is returned.

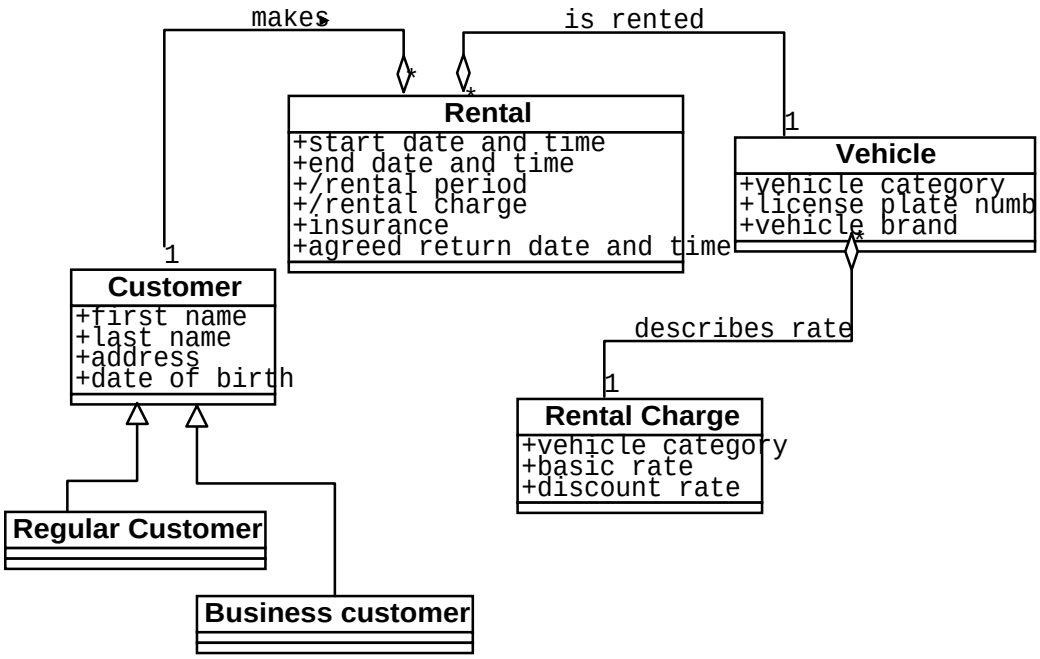
Draw a UML state diagram that shows the **states**, **events, conditions** and **actions** in the system. In addition to the state diagram, specify three lists for **states**, **events** and **actions** separately, clearly showing if you consider an element a state, an event or an action.
The purpose of the state diagram is to show how the bank machine handles the transaction, in particular it should clearly illustrate how the machine responds to customer inputs on the terminal and how the machine manages cash dispense, customer identification, transaction authorization and card handling.

States: Idle (start state), wait for PIN code, wait for cash amount
Events: cancel, card inserted, PIN entered, amount entered
Conditions: [PIN correct], [PIN incorrect], [Amount ok], [Amount exceeds limit] [trials < 3] [trials=3]
Actions: return card, revoke card, dispense cash



Question 10: (6) Draw a UML class diagram, containing concepts, generalizations, named associations with multiplicity (composition and aggregation where applicable) and attributes for the following domain. The task is to design a software system for a car rental that manages the rental and return of vehicles. There are two types of customers, regular customers and preferred business customers of companies that obtain a lower cooperate rate. A driver rents a vehicle for a certain period of time, specified through the agreed pick-up and return dates and times. The vehicle must be returned to the rental location by the agreed return date otherwise surcharges are applied. Vehicles are classified into different categories (economy, compact, standard, intermediate, luxury) for which different rates apply. Rental charges are calculated from the daily rate for that vehicle multiplied by the number of rental days. The customer has the ability to purchase additional insurance for collision damage, liability and theft. Drivers under 25 have to pay an additional insurance premium.

Frank Hoffmann
your name:_____ your person number:_____-_____



Good-luck!!!