**UTCN**

**Computer Science Department**

**Software Design 2012/2013**

**LABORATORY 2 – OCL**

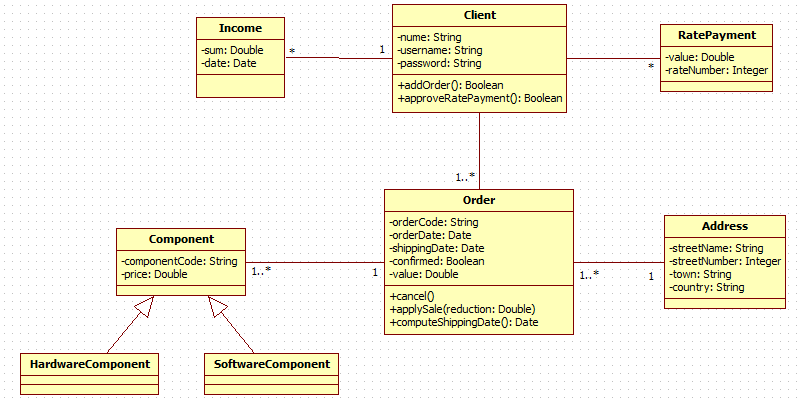
**==================================================================**

1. **Theoretical Background**

See course 2 (link).

1. **Problems**

**P1.** Consider the UML class diagram in Figure 1.

****

**Figure 1.** UML class diagram

1. Write constraints as OCL invariants for the following:

a.1) the *value* attribute of the *Order* class should be > 0.

a.2) the *value* attribute of the *Order* class will be 0 if no component has been ordered.

a.3) the *value* attribute of the *Order* class is equal to the sum of the ordered components’ prices.

a.4) different instances of the *Component* class have distinct codes.

1. Write a precondition in OCL to specify that the price reduction which is applied to the value of an order must be positive.
2. Write post-conditions in OCL to specify the following constraints:

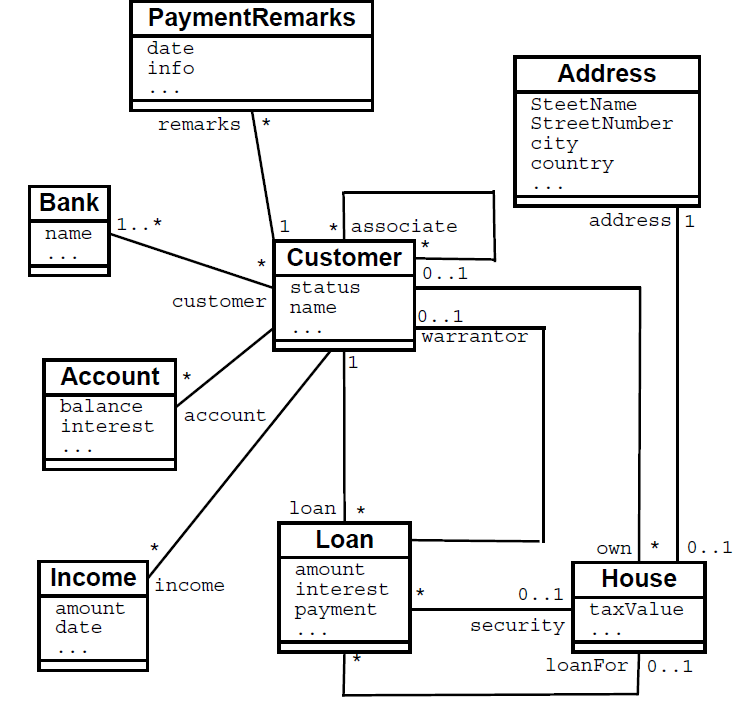
c.1) cancelling an order means setting its confirmation on false.

c.2) the result of the *applySale* operation from the *Order* class consists of applying a reduction to the total value of the order.

c.3) the result of the *approveRatePayment* from the *Client* class specifies that a client can pay an order in more rates. Consider that a client can pay the ordered components in rates if the following constraints are fulfilled: the client has received a salary in the last 12 months and within each month the salary was higher than 2000 lei. Write a post-condition in OCL for the operation *approveRatePayment* which expresses the conditions mentioned previously.

**P2.** Consider the UML class diagram presented in Figure 2. Write constraints as OCL invariants for the concept Loan:

1. If a loan has a warrantor, then the warrantor is an associate of the customer who has the loan.
2. If a Loan has a House as security, then the Customer having the Loan must own that House.
3. If a Loan has a House as security, then that House must have a taxValue (assessed value) of at least 20% of the amount of the Loan.



**Figure 2.** UML class diagram

**P3.** Consider the UML class diagram presented in Figure 2 and assume that the number of instances of concept Income that are related with a customer shows how many times during the last 12 months the person got a salary. If there are, for example, 9 instances of Income related to a customer, then the customer got a salary in 9 out of 12 months. Suppose that an operation getLoan():Boolean is added to class Customer. This operation is supposed to return whether a customer can be given a loan (true) or not (false). As a business decision, a customer can be given a loan if and only if the following conditions hold (unless houses as securities or warrantors are involved, in this case the whole situation gets more complicated):

* The customer got a salary in each of the last 12 months.
* In each of the last 12 months, the salary was more than 1000 euro

Give an OCL specification of Customer::getLoan() that expresses these conditions..

**P4.** Consider the class model in Figure 3 for managing credit card accounts. An institution may issue many credit card accounts, each identified by an account number.Each account has a maximum credit limit, a current balance and a mailing address. The account serves one or more customers who reside at the mailing address. The institution periodically issues a statement for each account. The statement lists a payment due date, finance charge and minimum payment. The statement itemizes various transactions that have occurred throughout the billing interval: cash advances, interest charges, purchases, fees and adjustments to the account. The name of the merchanmt is printed for each purchase. Write OCL expressions to answer the following credit card questions:

1. What transactions occurred for a credit card account within a time interval?
2. What volume of transactions were handled by an institution in the last year?
3. How many credit card accounts does a customer currently have?
4. What is the total maximum credit for a customer, for all accounts?

****

**Figure 3.** Class model for managing credit card accounts

1. **Supplementary Bibliography**

[1]<http://www.csci.csusb.edu/dick/samples/ocl.html>