# Lab assignment 4: Domain Driven Design

## Exercise 1: Entities, value objects and aggregate roots

a. In the domain model of the webshop we designed in Lab 2, define for each domain class if it is a value object or an entity.
b. In the same class diagram, find the aggregates, and the corresponding aggregate roots. Separate the aggregates from each other.

## Exercise 2: Implementing DDD

Apply DDD to the code for our webshop that we wrote in Lab 3.
Extend this code with the following functionality:

In the ShoppingService add the method:

***void checkOut(String cartId)***

Write the OrderService with the methods:

***void createOrder(ShoppingCart shoppingCart)***
***Order getOrder(String orderId)***

You can use the cartId as the unique id for the order.

The createOrder method only transforms a ShoppingCart into an Order and saves this Order in the database. You don't need to add Address, Shipping, Payment, Customer, etc.

## Exercise 3: Rest Client

Write one Rest client application that makes the following rest calls to our webshop:

1. Add product1
2. Add product2
3. Add product1 to the shoppingcart (with cartId=1) with quantity 3
4. Add product2 to the shoppingcart (with cartId=1) with quantity 2
5. Checkout the shoppingcart to create an order with orderId=1
6. Retrieve the order with orderId=1 and print it to the console.

In Sakai you find the projects RestClient and RestServer. Unzip both files somewhere on your computer, and then import them in Eclipse. First run the RestServer. Then run the RestClient. You can use the RestClient project as example to write the Rest Client for the webshop.