

Coding practices

We used the coding practices to improve the quality of the code. Readability was the most important goal for our team. When team members have problem with their implementation other members don't need too much time to understand the code and can easily provide help. To keep the coding conventions in our project clear we used the Google java style Guide.

We also tried to keep methods and function as small as possible. For example the method `subscribeToPlan` has to check if the customer is a new customer in our system. We splitted the check to an extra method `isNewCustomer` which get called by `subscribeToPlan`.

Location: `task3\src\core\Controller.java`

```
public void subscribeToPlan(int customerLocalID, int productID, int planType) {  
    Customer customer = getCustomerByLocalID(customerLocalID);  
    Product productPlan = getProductPlanByID(productID, planType);  
  
    //if (productPlan == null) System.out.println("is null");  
    //System.out.println("is subscribed?: " + isSubscribedToProduct(customerLocalID, productID));  
  
    boolean isNewCustomer = isNewCustomer(customerLocalID, productID);  
  
    if (customer != null  
        && productPlan != null  
        && !isSubscribedToProduct(customerLocalID, productID)) {  
  
        Subscription subscription = new Subscription(customerLocalID, productID, planType,  
            isNewCustomer);  
        subscriptions.add(subscription);  
  
        double chargedAmount = productPlan.getPlanRate();  
        if (isNewCustomer) {  
            chargedAmount = 0;  
        }  
  
        LocalDateTime date = subscription.getValidUntil();  
        YearMonth paidMonth = YearMonth.from(date);  
        //if chargedAmount is 0, the first month is free for the customer and the invoice is "paid"  
        customer.addInvoice(subscription.getSubscriptionID(), chargedAmount, paidMonth, null);  
    }  
}
```

```
private boolean isNewCustomer(int customerLocalID, int productID) {  
    for (Subscription subscription : subscriptions) {  
        if (subscription.getCustomerID() == customerLocalID  
            && subscription.getProductID() == productID) {  
            return false;  
        }  
    }  
  
    return true;  
}
```

We also focused that the naming of functions and methods are pronounceable, searchable and avoid disinformation. Our functions and methods have verb phrase names, e.g. createProduct, createCustomer or paymentReceived. Furthermore our classes and objects have noun phrase names, e.g. PaymentMehod, ProxyCient or Product. Moreover we avoided comments for obvius code and we tried to comment as we wrote the code, which saved us a lot of time.