Key design principles

Abstraction Principle
The abstraction principle means that each significant piece of functionality in a program should be implemented in just one place in the source code. We used abstraction to focus on the essential features of a design element.

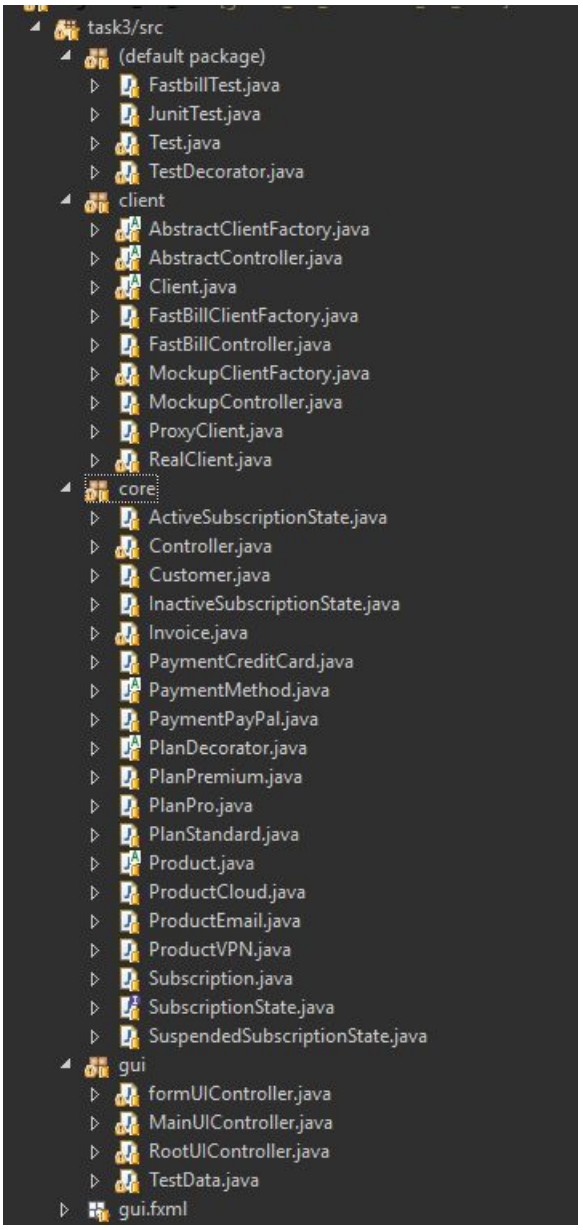
For example we have the method subcribeToPlan:
Location: task3\src\core\Controller.java

Properties like the customerLocalD, productID and the planType are essential, but other knowledge about the subscription like the cost or the time of the subscription or where the subscription is going to be store are not essential for the task the class design is used for.

```
public void subscribeToPlan(int customerLocalID, int productID, int planType) {

  Customer customer = getCustomerByLocalID(customerLocalID);
  Product productPlan = getProductPlanByID(productID, planType);


  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);

    customer.addInvoice(subscription.getSubscriptionID(), chargedAmount, paidMonth, null);

  }


}
```

Modularity

We used packages as a modular solution. Through the packages we can add a level of abstraction to our project landscape. We have three main packages the package client which handle all client specific code, the package core which has all classes to create objects and to manage and edit the objects and the last package is responsible for the graphical interface.


task3/src
(default package)
FastbillTest.java
JunitTest.java
Test.java
TestDecorator.java
client
AbstractClientFactory.java
AbstractController.java
Client.java
FastBillClientFactory.java
FastBillController.java
MockupClientFactory.java
MockupController.java
ProxyClient.java
RealClient.java
core
ActiveSubscriptionState.java
Controller.java
Customer.java
InactiveSubscriptionState.java
Invoice.java
PaymentCreditCard.java
PaymentMethod.java
PaymentPayPal.java
PlanDecorator.java
PlanPremium.java
PlanPro.java
PlanStandard.java
Product.java
ProductCloud.java
ProductEmail.java
ProductVPN.java
Subscription.java
SubscriptionState.java
SuspendedSubscriptionState.java
gui
formUIController.java
MainUIController.java
RootUIController.java
TestData.java
gui.fxml