Application start from MarketStore class which is entry point for this task. After start application execution of program continue in Engine class which implements Runnable interface. In run method in Engine class I create three instance of Card: Bronze Card, Silver Card and Gold Card. I use PayDeck class four static methods for printing output. In my structure I use abstract class BaseCard which contains common information for other type of cards. In every different type I override methods for calculate depending on condition. So when I create first object bronzeCard and I call with static method PayDeck.printPurchaseValue(bronzeCard) I use methotds in BronzeCard class. It is analogous with other methods in Paydeck class PayDeck.printDiscountRate(bronzeCard), PayDeck.printDiscount(bronzeCard) and PayDeck.printTotalPurchaseValue(bronzeCard). When I call Paydeck with other object silverCard I use methods in SilverCard class. Similarly is with goldCard object. I use hard-code values in initial value of object, because I understand condition in this way, but if this is mistake I can create new variable and put in initial state of objects.