

//state pattern //Context  
class Account{​​​​​  
Balance=0;  
const MinBalnce=500;  
const OverDue=1000;  
IAccountStateHandler stateHandler;  
public void setStateHandler(IAccountStateHanlder handler) {​​​​​ this.stateHandler=handler;}​​​​​  
public void Deposit(int amt){​​​​​  
  this.stateHandler.Deposit(amt,this);  
}​​​​​  
public void WithDraw(int amt){​​​​​   this.stateHandler.Withdraw(amt,this);  
}​​​​​  
}​​​​​ interface IAccountStateHandler{​​​​​  
 void Deposit(int amt,Account context);  
 void Withdraw(int amt,Account context);  
}​​​​​ class AccountOpensStateHandler:IAccountStateHandler{​​​​​ public  void Deposit(int amt,Account context){​​​​​  
        conetxt.balance+=amt;   
        if(balance > minBalance){​​​​​  
        context.setStateHandler(new ActiveStateHandler())  
        }​​​​​  
}​​​​​  
public  void Withdraw(int amt,Account context);  
}​​​​​ class AccountActiveStateHandler:IAccountStateHandler{​​​​​  
public  void Deposit(int amt,Account context){​​​​​  
}​​​​​  
public  void Withdraw(int amt,Account context);  
}​​​​​



