using System;

using System.Collections.Generic;

public class Customer

{

private string name;

private string address;

private List<Order> orders;

public Customer(string name, string address)

{

this.name = name;

this.address = address;

this.orders = new List<Order>();

}

public void addOrder(Order order)

{

this.orders.Add(order);

}

public void removeOrder(Order order)

{

this.orders.Remove(order);

}

public string getName()

{

return this.name;

}

public string getAddress()

{

return this.address;

}

public List<Order> getOrders()

{

return this.orders;

}

}

public class Order

{

private DateTime date;

private string status;

private List<OrderDetail> orderDetails;

private Payment payment;

public Order(DateTime date, string status)

{

this.date = date;

this.status = status;

this.orderDetails = new List<OrderDetail>();

this.payment = null;

}

// Methods

public void addOrderDetail(OrderDetail orderDetail)

{

this.orderDetails.Add(orderDetail);

}

public void removeOrderDetail(OrderDetail orderDetail)

{

this.orderDetails.Remove(orderDetail);

}

public void setPayment(Payment payment)

{

this.payment = payment;

}

public DateTime getDate()

{

return this.date;

}

public string getStatus()

{

return this.status;

}

public List<OrderDetail> getOrderDetails()

{

return this.orderDetails;

}

public Payment getPayment()

{

return this.payment;

}

public double calcSubTotal()

{

double subTotal = 0;

foreach (OrderDetail orderDetail in this.orderDetails)

{

subTotal += orderDetail.calcSubTotal();

}

return subTotal;

}

public double calcTax()

{

double tax = 0;

foreach (OrderDetail orderDetail in this.orderDetails)

{

tax += orderDetail.calcTax();

}

return tax;

}

public double calcTotal()

{

return this.calcSubTotal() + this.calcTax();

}

public double calcTotalWeight()

{

double totalWeight = 0;

foreach (OrderDetail orderDetail in this.orderDetails)

{

totalWeight += orderDetail.calcWeight();

}

return totalWeight;

}

}

public abstract class Payment

{

private double amount;

public Payment(double amount)

{

this.amount = amount;

}

public double getAmount()

{

return this.amount;

}

public abstract bool authorized();

}

public class Cash : Payment

{

private double cashTendered;

public Cash(double amount, double cashTendered) : base(amount)

{

this.cashTendered = cashTendered;

}

public double getCashTendered()

{

return this.cashTendered;

}

public override bool authorized()

{

return true;

}

}

public class Check : Payment

{

private string name;

private string bankID;

public Check(double amount, string name, string bankID) : base(amount)

{

this.name = name;

this.bankID = bankID;

}

public string getName()

{

return this.name;

}

public string getBankID()

{

return this.bankID;

}

public override bool authorized()

{

return true;

}

}

public class Credit : Payment

{

private string name;

private string type;

private DateTime expDate;

public Credit(double amount, string name, string type, DateTime expDate) : base(amount)

{

this.name = name;

this.type = type;

this.expDate = expDate;

}

// Methods

public string getName()

{

return this.name;

}

public string getType()

{

return this.type;

}

public DateTime getExpDate()

{

return this.expDate;

}

public override bool authorized()

{

return true;

}

}

public class OrderDetail

{

private int quantity;

private string tax;

}