

Main:

```
import PaymentServices.PhoneOrderService;
import PaymentServices.OrderService;

public class Main {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        OrderService orderService = null;
        String customerName;
        Order order;
        int customerAnswerForOrder = 0;
        int customerAnswerForPaymentMethod = 0;

        System.out.println("Enter Customer Name : ");
        customerName = scanner.nextLine();
        order = new Order(customerName);

        while (customerAnswerForOrder != 3) {
            System.out.println("For Ordering Sandwich enter 1.");
            System.out.println("For Ordering Pizza enter 2.");
            System.out.println("For submit your order enter 3");
            customerAnswerForOrder = scanner.nextInt();

            if (customerAnswerForOrder == 1) {
                order.addItem(new Food("sandwich", 1000));
            } else if (customerAnswerForOrder == 2) {
                order.addItem(new Food("pizza", 2000));
            }
        }

        System.out.println("Enter Your Payment Method (1 for online, 2 for on-site, 3 for phone):");
        customerAnswerForPaymentMethod = scanner.nextInt();

        if (customerAnswerForPaymentMethod == 1) {
            orderService = new OnlineOrderService();
            orderService.onlineOrderRegister(customerName);

            orderService = new PhoneOrderService();
            orderService.onlineOrderRegister(customerName);
        } else if (customerAnswerForPaymentMethod == 2) {
            orderService = new OnSiteOrderService();
            orderService.onSiteOrderRegister(customerName);

            orderService = new PhoneOrderService();
            orderService.onSiteOrderRegister(customerName);
        } else if (customerAnswerForPaymentMethod == 3) {
            orderService = new PhoneOrderService();
            orderService.phoneOrderRegister(customerName);
        }
    }
}
```

```

        System.out.println("Pay Price:");
        if (orderService instanceof PhoneOrderService) {
            orderService.phoneOrderPayment(order.getTotalPrice());
        } else if (orderService instanceof OnlineOrderService) {
            orderService.onlineOrderPayment(order.getTotalPrice());
        } else if (orderService instanceof OnSiteOrderService) {
            orderService.onSiteOrderPayment(order.getTotalPrice());
        }
        System.out.println(order);
    }
}

```

OrderManager:

```

import java.util.ArrayList;
import java.time.LocalDateTime;

public class OrderManager {
    private ArrayList<Food> foods;
    private String customerName;

    public OrderManager(String customerName) {
        this.customerName = customerName;
        this.foods = new ArrayList<>();
    }

    public void addItem(Food food) {
        this.foods.add(food);
    }

    public int getTotalPrice() {
        return this.foods.stream().mapToInt(food -> food.getPrice()).sum();
    }

    @Override
    public String toString() {
        StringBuilder orders = new StringBuilder();
        for (Food food : this.foods) {
            orders.append(food.getFoodName()).append(" -> ")
                .append(food.getPrice()).append("\n");
        }
        return "Customer: " + customerName + "\nOrders are:\n" +
            orders.toString() + "Total Price: " + this.getTotalPrice();
    }
}

```

PaymentMethodManager:

```

public class PaymentMethodManager {
    public OrderService selectOrderService(int
customerAnswerForPaymentMethod) {
        if (customerAnswerForPaymentMethod == 1) {
            return new OnlineOrderService();
        }
    }
}

```

```

        } else if (customerAnswerForPaymentMethod == 2) {
            return new OnSiteOrderService();
        } else {
            return new PhoneOrderService();
        }
    }
}

```

UserInputManager:

```

import java.util.Scanner;

public class UserInputManager {
    private Scanner scanner;

    public UserInputManager() {
        this.scanner = new Scanner(System.in);
    }

    public String getCustomerName() {
        System.out.println("Enter Customer Name:");
        return scanner.nextLine();
    }

    public int getCustomerAnswerForOrder() {
        System.out.println("For Ordering Sandwich enter 1.\nFor Ordering Pizza enter 2.\nFor submit your order enter 3");
        return scanner.nextInt();
    }

    public int getCustomerAnswerForPaymentMethod() {
        System.out.println("Enter Your Payment Method (1 for online, 2 for on-site, 3 for phone order):");
        return scanner.nextInt();
    }
}

```

OrderPayment:

```

public interface OrderPayment {
    void pay(int totalPrice);
}

```

OrderRegistration:

```

public interface OrderRegistration {
    void register(String customerName);
}

```

PhoneOrderService:

```
public class PhoneOrderService implements OrderRegistration, OrderPayment {
    @Override
    public void register(String customerName) {
        System.out.println("Phone order registered for: " + customerName);
    }

    @Override
    public void pay(int totalPrice) {
        System.out.println("Phone order payment: " + totalPrice + " Tomans");
    }
}
```