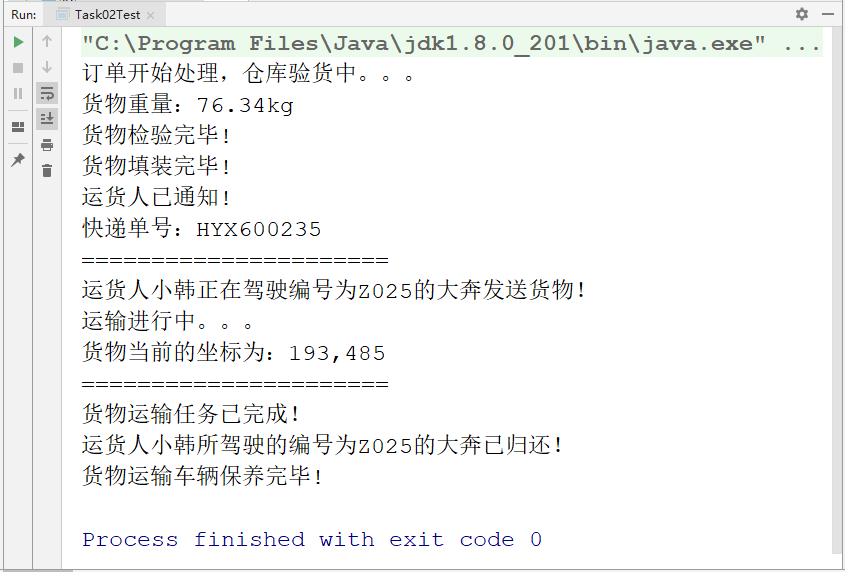
【案例4-8】模拟物流快递系统程序设计

**【案例介绍】**

1. **案例描述**

网购已成为人们生活的重要组成部分，当人们在购物网站中下订单后，订单中的货物就会在经过一系列的流程后，送到客户的手中。而在送货期间，物流管理人员可以在系统中查看所有物品的物流信息。编写一个模拟物流快递系统的程序，模拟后台系统处理货物的过程。

1. **运行结果**



**【案例目标】**

* 学会分析“模拟物流快递系统程序设计”任务实现的逻辑思路。
* 能够独立完成“模拟物流快递系统程序设计”的源代码编写、编译及运行。
* 掌握面向对象封装、继承和多态的概念和使用。
* 掌握抽象类和接口的使用。

**【案例分析】**

（1）运输货物首先需要有交通工具，所以需要定义一个交通工具类。由于交通工具可能有很多，所以可以将该交通工具类定义成一个抽象类，类中需要包含该交通工具的编号，型号以及运货负责人等属性，还需要定义一个抽象的运输方法。

（2）当运输完成后，需要对交通工具进行保养，所以需要定义保养接口，具备交通工具的保养功能。

（3）交通工具可能有很多种，这里可以定义一个专用运输车类，该类需要继承交通工具类，并实现保养接口。

（4）有了运输的交通工具后，就可以开始运送货物了。货物在运输前，运输时和运输后，都需要检查和记录，并且每一个快递都有快递单号，这时可以定义一个快递任务类包含快递单号和货物重量的属性，以及送前、发送货物途中和送后的方法。

（5）在货物运输过程中，需要对运输车辆定位，以便随时跟踪货物的位置信息。定位功能可以使用GPS，而考虑到能够实现定位功能的设备可能有很多（如手机、专用定位仪器等），这时可以定义一个包含定位功能的GPS接口，以及实现了该接口的仪器类（如Phone等）。

（6）编写测试类，运行查看结果。

**【案例实现】**

定义交通工具类Transportation，该类是一个抽象类，包含交通工具信息和运输货物方法，其代码如文件4-26所示。

Transportation.java

1. /\*
2. \* 交通工具类
3. \*/
4. public abstract class Transportation {
5. private String number; // 编号
6. private String model; // 型号
7. private String admin; // 运货负责人
8. public Transportation() {
9. super();//可省略
10. }
11. public Transportation(String number, String model, String admin) {
12. this.number = number;
13. this.model = model;
14. this.admin = admin;
15. }
16. // 运输方法
17. public abstract void transport();
18. // 编号
19. public void setNumber(String number) {
20. this.number = number;
21. }
22. public String getNumber() {
23. return number;
24. }
25. // 型号
26. public void setModel(String model) {
27. this.model = model;
28. }
29. public String getModel() {
30. return model;
31. }
32. // 负责人
33. public void setAdmin(String admin) {
34. this.admin = admin;
35. }
36. public String getAdmin() {
37. return admin;
38. }
39. }

在上述代码中，分别定义了车辆编号、车辆型号和车辆负责人的属性，以及其各自的getter和setter方法，同时还定义了一个抽象的运输方法transport()。

定义交通工具保养接口Careable，该接口中包含车辆保养的方法，其实现代码如下所示。

Careable.java

1. /\*
2. \* 定义保养接口，具备保养功能。
3. \*/
4. public interface Careable{
5. //保养方法
6. public abstract void upKeep();
7. }

定义专用运输车类Ztransportation，该类继承了交通工具类，并实现了保养接口，其实现代码如下所示。

Ztransportation.java

1. /\*
2. \* 专用运输车类
3. \*/
4. public class ZTransportation extends Transportation implements Careable{
5. //无参构造
6. public ZTransportation() {
7. super();
8. }
9. //有参构造：车辆编号、型号、负责人
10. public ZTransportation(String number, String model, String admin) {
11. super(number, model, admin);
12. }
13. // 运输方法
14. public void transport() {
15. System.out.println("运输进行中。。。");
16. }
17. // 重写车辆保养方法
18. public void upKeep() {
19. System.out.println("货物运输车辆保养完毕!");
20. }
21. }

定义快递任务类SendTask，该类的具体实现代码如下所示。

SendTask.java

1. /\*
2. \* 快递任务类
3. \*/
4. public class SendTask {
5. private String number; // 快递单号
6. private double goodsWeight; // 货物重量
7. public SendTask() {
8. super(); //可省略
9. }
10. public SendTask(String number, double goodsWeight) {
11. this.number = number;
12. this.goodsWeight = goodsWeight;
13. }
14. //送前准备
15. public void sendBefore () {
16. System.out.println("订单开始处理，仓库验货中。。。");
17. System.out.println("货物重量："+this.getGoodsWeight()+"kg");
18. System.out.println("货物检验完毕!");
19. System.out.println("货物填装完毕!");
20. System.out.println("运货人已通知!");
21. System.out.println("快递单号："+this.getNumber());
22. }
23. //发送货物
24. public void send(Transportation t,GPS tool) {
25. System.out.println("运货人"+t.getAdmin()
26. +"正在驾驶编号为"+t.getNumber()
27. +"的"+t.getModel()+"发送货物！");
28. t. transport();
29. String showCoordinate = tool.showCoordinate();
30. System.out.println("货物当前的坐标为："+showCoordinate);
31. }
32. //送后操作
33. public void sendAfter(Transportation t) {
34. System.out.println("货物运输任务已完成！");
35. System.out.println("运货人"+t.getAdmin()
36. +"所驾驶的编号为"+t.getNumber()
37. +"的"+t.getModel()+"已归还！");
38. }
39. public String getNumber() {
40. return number;
41. }
42. public void setNumber(String number) {
43. this.number = number;
44. }
45. public double getGoodsWeight() {
46. return goodsWeight;
47. }
48. public void setGoodsWeight(double goodsWeight) {
49. this.goodsWeight = goodsWeight;
50. }
51. }

定义包含显示位置功能的GPS接口和实现类Phone，其实现代码如GPS.java和Phone.java所示。

GPS.java

1. /\*
2. \* 定义GPS接口，具备GPS定位功能。
3. \*/
4. public interface GPS{
5. //显示坐标的方法
6. public String showCoordinate();
7. }

Phone.java

1. /\*
2. \*定义一个手机类，实现GPS接口，拥有定位功能。
3. \*/
4. class Phone implements GPS{
5. public Phone() { //空参构造
6. super();
7. }
8. //定位方法
9. public String showCoordinate() {
10. String location = "193,485";
11. return location;
12. }
13. }

定义测试类，实例化对象并传入数据，测试运行结果，其代码如下所示。

Task02Test.java

1. /\*
2. \* 定义测试类
3. \*/
4. public class Task02Test {
5. public static void main(String[] args) {
6. // 快递任务类对象
7. SendTask task = new SendTask("HYX600235",76.34);
8. // 调用送前准备方法
9. task.sendBefore();
10. System.out.println("======================");
11. // 创建交通工具对象
12. ZTransportation t = new ZTransportation("Z025","大奔","小韩");
13. //创建GPS工具对象
14. Phone p = new Phone();
15. // 将交通工具与GPS工具传入送货方法
16. task.send(t,p);
17. System.out.println("======================");
18. // 调用送后操作方法
19. task.sendAfter(t);
20. t.upKeep();
21. }
22. }