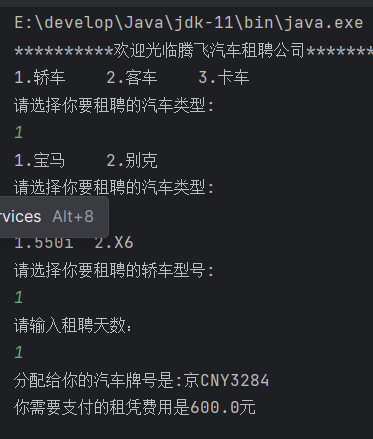
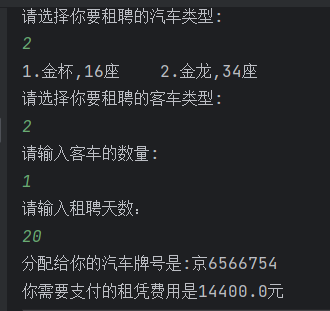
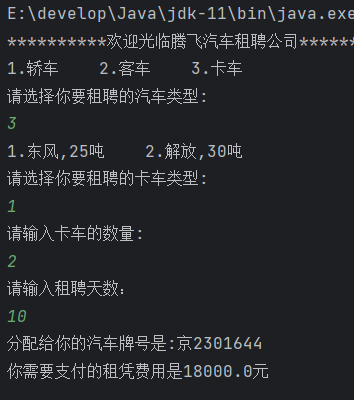
**《汽车租赁系统V1.0》实验报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 202231060920 | 姓名 | 李浩楠 |
| 专业年级 | 软件工程2022级 | 得分 |  |

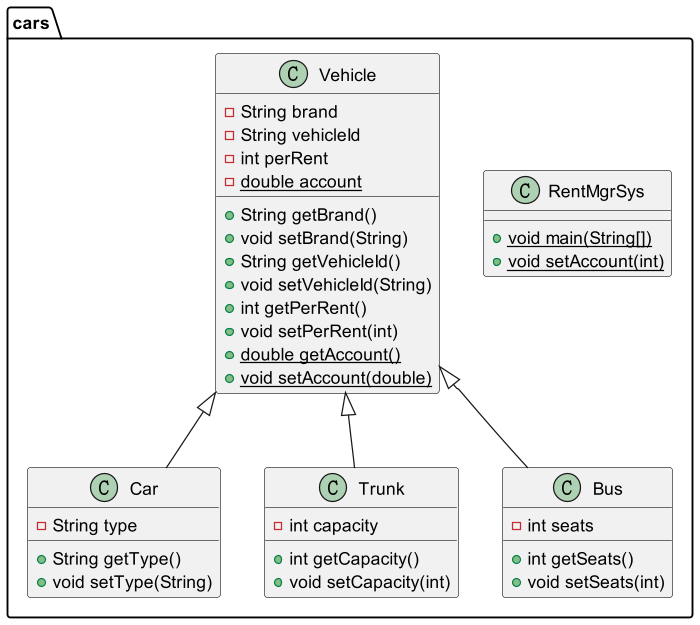
一、功能展示及核心代码

1、汽车租赁系统V 1.0界面展示





2、类图（Car）



1. 功能核心代码（请将你的核心算法详尽注释和添加说明）：

package cars;  
  
import java.text.DecimalFormat;  
import java.util.ArrayList;  
import java.util.Scanner;  
  
public class RentMgrSys {  
 public static void main(String[] args) {  
 *//创建轿车对象并存入list中* Car BWM1 = new Car("宝马", "京NY28588", 800, "X6");  
 Car BWM2 = new Car("宝马", "京CNY3284", 600, "550i");  
 Car BUICK1 = new Car("别克", "京NT37465", 300, "林荫大道");  
 Car BUICK2 = new Car("别克", "京NT96968", 600, "GL8");  
 ArrayList<Car> cars = new ArrayList<>();  
 cars.add(BWM1);  
 cars.add(BWM2);  
 cars.add(BUICK1);  
 cars.add(BUICK2);  
 *//创建客车对象并存入list中* Bus jl1 = new Bus("金龙", "京6566754", 800, 16);  
 Bus jl2 = new Bus("金龙", "京8696997", 800, 16);  
 Bus jb1 = new Bus("金杯", "京9696996", 1500, 34);  
 Bus jb2 = new Bus("金杯", "京8696998", 1500, 34);  
 ArrayList<Bus> buses = new ArrayList<>();  
 buses.add(jl1);  
 buses.add(jl2);  
 buses.add(jb1);  
 buses.add(jb2);  
 *//创建卡车对象并存入list中* Trunk df1 = new Trunk("东风", "京2301644", 1000, 25);  
 Trunk df2 = new Trunk("东风", "京awf9861", 1500, 30);  
 Trunk jf1 = new Trunk("解放", "京66fwv94", 1200, 25);  
 Trunk jf2 = new Trunk("解放", "京ag97h26", 2000, 30);  
 ArrayList<Trunk> trunks = new ArrayList<>();  
 trunks.add(df1);  
 trunks.add(df2);  
 trunks.add(jf1);  
 trunks.add(jf2);  
  
 Scanner sc = new Scanner(System.*in*);  
 int days = 0;  
 System.*out*.println("\*\*\*\*\*\*\*\*\*\*欢迎光临腾飞汽车租聘公司\*\*\*\*\*\*\*\*\*");  
 System.*out*.println("1.轿车 2.客车 3.卡车");  
 int choice1 = 0, choice2 = 0, choice3 = 0;  
 String brand = null, type = null;  
 System.*out*.println("请选择你要租聘的汽车类型:");*//选择汽车类型* choice1 = sc.nextInt();  
 switch (choice1) {  
 case 1:  
 System.*out*.println("1.宝马 2.别克");  
 System.*out*.println("请选择你要租聘的汽车类型:");*//选择汽车品牌* choice2 = sc.nextInt();  
   
 *//保存品牌与型号* switch (choice2) {  
 case 1:  
 brand = "宝马";  
 System.*out*.println("1.550i 2.X6");  
 System.*out*.println("请选择你要租聘的轿车型号:");  
 choice3 = sc.nextInt();  
 switch (choice3) {  
 case 1:  
 type = "550i";  
 break;  
 case 2:  
 type = "X6";  
 break;  
 }  
 break;  
 case 2:  
 brand = "别克";  
 System.*out*.println("1.林荫大道 2.GL8");  
 System.*out*.println("请选择你要租聘的轿车型号:");  
 choice3 = sc.nextInt();  
 switch (choice3) {  
 case 1:  
 type = "林荫大道";  
 break;  
 case 2:  
 type = "GL8";  
 break;  
 }  
 break;  
 }  
 break;  
   
 case 2:  
 System.*out*.println("1.金杯,16座 2.金龙,34座");  
 System.*out*.println("请选择你要租聘的客车类型:");*//选择客车类型* choice2 = sc.nextInt();  
 *//保存客车品牌* switch (choice2) {  
 case 1:  
 brand = "金杯";  
 break;  
 case 2:  
 brand = "金龙";  
 break;  
 }  
 System.*out*.println("请输入客车的数量:");*//选择客车数量* choice3 = sc.nextInt();  
 break;  
 case 3:  
 System.*out*.println("1.东风,25吨 2.解放,30吨");  
 System.*out*.println("请选择你要租聘的卡车类型:");*//选择货车类型* choice2 = sc.nextInt();  
 *//保存货车品牌* switch (choice2) {  
 case 1:  
 brand = "东风";  
 break;  
 case 2:  
 brand = "解放";  
 break;  
 }  
 System.*out*.println("请输入卡车的数量:");*//选择客车数量* choice3 = sc.nextInt();  
 break;  
 default:  
 System.*out*.println("输入错误");  
 break;  
 }  
 System.*out*.println("请输入租聘天数：");*//选择租聘天数* days = sc.nextInt();  
 *setAccount*(days);  
 DecimalFormat df = new DecimalFormat(".0");  
   
 *//输出租聘车牌与费用* switch (choice1) {  
 case 1:  
 for (Car car : cars) {  
 if (car.getType().equals(type)) {  
 double cost = days \* car.getPerRent() \* Vehicle.*getAccount*();  
 System.*out*.println("分配给你的汽车牌号是:"+car.getVehicleId());  
 System.*out*.print("你需要支付的租凭费用是");  
 System.*out*.println(df.format(cost) + "元");*//输出相应的费用* break;  
 }  
 }  
 break;  
 case 2:  
 for (Bus bus : buses) {  
 if (bus.getBrand().equals(brand)) {  
 double cost = choice3 \* days \* bus.getPerRent() \* Vehicle.*getAccount*();  
 System.*out*.println("分配给你的汽车牌号是:"+bus.getVehicleId());  
 System.*out*.print("你需要支付的租凭费用是");  
 System.*out*.println(df.format(cost) + "元");*//输出相应的费用* break;  
 }  
 }  
 break;  
 case 3:  
 for (Trunk trunk : trunks) {  
 if (trunk.getBrand().equals(brand)) {  
 double cost = choice3 \* days \* trunk.getPerRent() \* Vehicle.*getAccount*();  
 System.*out*.println("分配给你的汽车牌号是:"+trunk.getVehicleId());  
 System.*out*.print("你需要支付的租凭费用是");  
 System.*out*.println(df.format(cost) + "元");*//输出相应的费用* break;  
 }  
 }  
 break;  
 }  
 }  
 *//设置折扣* public static void setAccount(int days) {  
 if (days > 150) Vehicle.*setAccount*(0.7);  
 else if (days > 30) Vehicle.*setAccount*(0.8);  
 else if (days > 7) {  
 Vehicle.*setAccount*(0.9);  
 } else Vehicle.*setAccount*(1.0);  
 }  
}

二、收获，体会及问题

(请详细书写，写得越详细、越个性化、越真实越好，否则指导教师不知道你做这个实验的心路历程，也就无法充分地判断你是否是独立完成的这个实验、你是否在做这个实验时进行了认真仔细地思考、通过这个实验你是否在实践能力上得到了提高)

1. 收获：通过将系统划分为对象，每个对象都有自己的属性和方法，可以有效地组织和管理代码。掌握了Java中的类和对象的创建和使用方式。熟悉了封装、继承和多态的概念和用法，通过封装，可以隐藏对象内部的实现细节；通过继承，可以实现代码的重用和扩展；通过多态，可以灵活地处理不同类型的对象。了解了如何使用类的继承和接口的实现来实现代码的复用和扩展，通过继承可以从已有的类中派生出新的类，并继承其属性和方法；通过接口可以定义一组规范，使得不同类可以实现相同的接口，提供统一的方法。
2. 体会：面向对象编程的思想和方法可以帮助我们更好地组织和管理代码，提高代码可读性和可维护性。通过合理地设计类和对象之间的关系，可以使代码结构更加清晰和灵活，便于后续的扩展和修改。封装、继承和多态是面向对象编程的重要特性，合理地运用它们可以使代码更加健壮和可扩展。
3. 问题：是否考虑了异常处理机制？在代码中是否有对潜在的异常情况进行捕获和处理？在实现过程中是否遵循了面向对象编程的原则和规范，如单一职责原则、开闭原则等？

在测试过程中是否发现了潜在的问题或漏洞？是否对代码进行了充分的测试和调试？

三、附件

请将你的所有项目源代码和本实验报告一起打包上传PTA系统。