Bài tập lớn môn lập trình hướng đối tượng

Số: 22

Project chỉ sử dụng Java core library của JDK 16.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STT | Họ và Tên | Mã sinh viên | Email | Công việc | Mức độ hoàn thành | Ghi chú |
| 1 | Hà Trung Kiên | 20190078 | kien.ht190078@sis.hust.edu.vn | - Lập trình để thực hiện chức năng thêm, sửa, xóa và tìm kiếm hợp đồng | 100% |  |
| 2 | Đinh Tuấn Anh | 20194219 | anh.dt194219@sis.hust.edu.vn | - Lập trình để thực hiện chức năng tìm kiếm hợp đồng có giá trị lớn hơn 1 số nhập vào | 100% |  |
| 3 | Vũ Đình Duy | 20183909 | duy.vd183909@sis.hust.edu.vn | - Lập trình để thực hiện chức năng tính chi phí thu được của bãi xe trong 1 khoảng thời gian | 100% |  |

PHẦN I: Giới thiệu bài toán(BTL số 22): **Quản lý cửa hàng thuê xe**:

Một cửa hàng cho thuê xe máy du lịch.

Khi có người đến thuê thì cần lưu lại hợp đồng thuê gồm các thông tin tênngười thuê, số CMT, thuê xe nào, thuê trong thời gian nào. Với hợp đồng thuê xe máy trong ngày thì giá thuê là 150.000đ/ngày. Với hợp đồng thuê xe máy theo tháng thì giá thuê là 3.000.000đ/tháng + phí bổ sung 1.000.000đ/hợp đồng.

Xây dựng chương trình quản lý bãi đỗ gồm các chức năng:

- Thêm, bớt, sửa, xóa một hợp đồng trông giữ xe

- Tìm kiếm hợp đồng trông giữ theo tên chủ xe, biển số xe, loại xe

- Tìm kiếm các hợp đồng có giá trị lớn hơn một số nhập vào

- Tính chi phí thu được của bãi xe trong một khoảng thời gian nhập vào

PHẦN II: Phân tích hệ thống:

1. Phân tích yêu cầu hệ thống:

Một ứng dụng có thể lưu trữ, thêm, sửa, xóa các hợp đồng với các tính năng tìm kiếm theo thông tin hợp đồng và giá trị hợp đồng.

1. Phân tích thành phần:

Object trung tâm của hệ thống là các Hợp đồng, chia làm 2 loại, theo tháng và theo ngày. Ứng dụng được xây dựng xung quanh class HopDong.

* 1. Các field của class được xây dựng từ input của user:

//Có thể kết tập 2 field tên khách hàng và số CMND thành class KhachHang

- Tên khách hàng: String

- Số CMND: String

- Biển số xe: String

- Thời gian thuê(ngày bắt đầu thuê, ngày kết thúc thuê): Date

- Loại hợp đồng(theo tháng/theo ngày): int

* 1. Các field của class được xây dựng bởi hệ thống:

- Giá trị của hợp đồng: int

- ID của hợp đồng: int

- số hợp đồng hiện tại: static int

* 1. Class HopDongTheoThang, HopDongTheoNgay extends HopDong

- 2 subclass của HopDong phân theo loại hợp đồng.

1. Xây dựng các class:
   1. Abstract class HopDong: class này abstract để tránh việc tạo nhầm một instance của class này. Class này không có abstract method. Các method getter, setter và ultility không được nêu ra.
      1. HopDongTheoThang extends HopDong
      2. HopDongTheoNgay extends HopDong
   2. Class XuLyDuLieu: private constructor, dùng để chứa các static method cho việc lưu dữ liệu.
      1. Public static ArrayList<HopDong> docDuLieu(){

//method trả về ArrayList chứa các hợp đồng được lưu trong file "data.dat"

}

* + 1. Public static void appendDuLieu(){

//method trung gian sẽ append thêm 1 dòng dữ liệu – tương ứng 1 dòng hợp đồng vào file dữ liệu "data.dat"

}

* + 1. Public static void writeFileData(ArrayList<HopDong> x){

//method sẽ ghi lại toàn bộ dữ liệu của file "data.dat" bởi dữ liệu được đưa vào từ ArrayList x

}

* 1. Class TimKiemDuLieu: private constructor, dùng để chứa các static method cho việc tìm kiếm dữ liệu.
     1. Public static void timTenKH(String tuKhoa, ArrayList<HopDong> x){

//method này sẽ xóa các hợp đồng không phù hợp với điều kiện về tên khách hàng khỏi ArrayList x

//tương tự ta có các method cho biển số xe, số CMND, ngày bắt đầu, ngày kết thúc và giá trị

}

* + 1. Public static int tongGiaTri(ArrayList<> x){

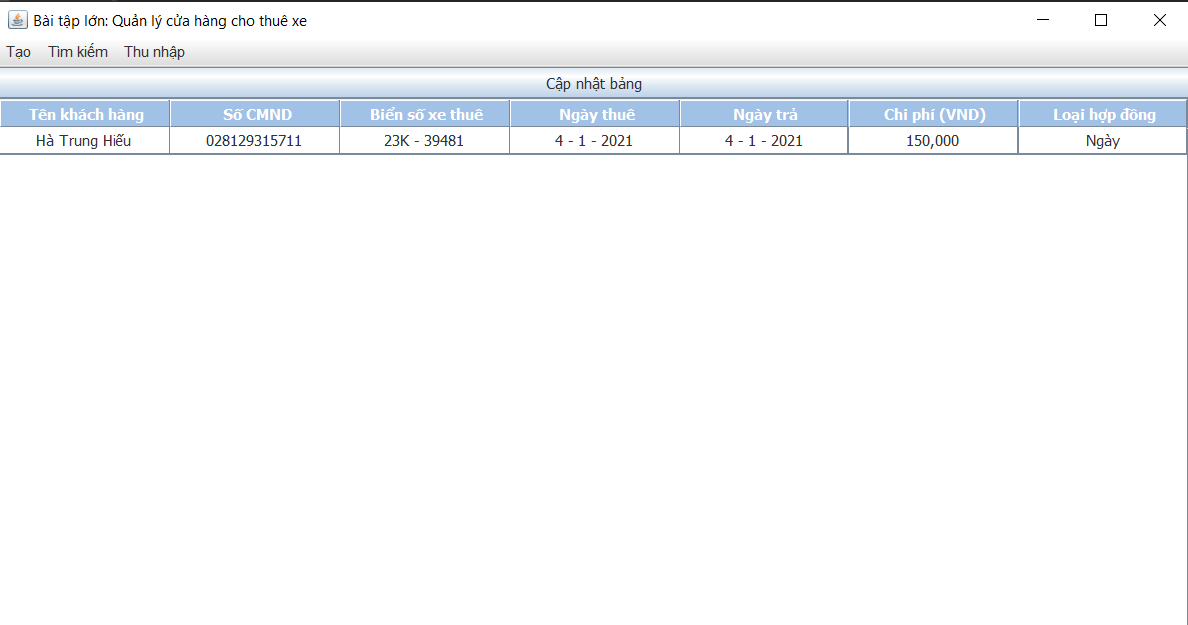
//method trả về tổng giá trị các hợp đồng trong ArrayList x

}

* 1. Class HopDongMa: là class trung gian dùng để chứa các giá trị đang được người dùng sửa đổi của một hợp đồng. Class có đầy đủ các field do người dùng nhập vào cùng với field giá trị của hợp đồng. Class này được tạo ra để giữ thông tin hợp đồng đang được sửa đổi, đồng thời tránh ảnh hưởng tới ID hợp đồng và field số hợp đồng.

//class này nhóm em thấy khá là ngớ ngẩn nhưng đây là phương án ít code nhất nhóm em tìm ra

* 1. Class Main

1. Các class GUI:
   1. ManHinhChinh: Class chứa màn hình làm việc chính của ứng dụng.
   2. TaoHopDongTheoThang: Class chứa màn hình tạo hợp đồng theo tháng
   3. TaoHopDongTheoNgay: Class chứa màn hình tạo hợp đồng theo ngày
   4. ManHinhTimKiem: Class chứa màn hình tìm kiếm
   5. ManHinhKetQuaTimKiem: Class chứa màn hình trả về kết quả tìm kiếm
   6. ManHinhSuaHopDong: Class chứa màn hình sửa hợp đồng
   7. ManHinhTimKiemChiPhi: Class chứa màn hình tìm kiếm hợp đồng theo giá trị
   8. ManHinhTongGiaTri: Class chứa màn hình phục vụ việc tính tổng giá trị các hợp đồng được chọn
   9. CustomCellRenderer: Class tạo DefaultTableCellRenderer phục vụ cho việc chỉnh sửa bảng

Phần III: Đánh giá

Hạn chế: có nhiều phần được code chưa tối ưu, có nhiều field/method đáng lẽ nên để là static thì lại là instance. Các class GUI có nhiều lỗi, layout chưa hợp lý.