import java.time.LocalDateTime;

import java.time.temporal.ChronoUnit;

// Interface untuk menampilkan informasi

interface Info {

void tampilkanInfo();

}

// Abstract class untuk entitas dengan ID

abstract class Entitas {

private String id;

public Entitas(String id) {

this.id = id;

}

public String getId() {

return id;

}

}

// Class Supir mengimplementasikan interface Info dan mewarisi Entitas

class Supir extends Entitas implements Info {

private String nip;

private String nama;

public Supir(String nip, String nama) {

super(nip);

this.nip = nip;

this.nama = nama;

}

public String getNip() {

return nip;

}

public String getNama() {

return nama;

}

@Override

public void tampilkanInfo() {

System.out.println("--- Informasi Supir ---");

System.out.println("NIP: " + nip);

System.out.println("Nama: " + nama);

}

}

// Class Mobil mengimplementasikan interface Info dan mewarisi Entitas

class Mobil extends Entitas implements Info {

private String nomorPolisi;

private String merkMobil;

private String warna;

private double tarifPerHari;

public Mobil(String nomorPolisi, String merkMobil, String warna, double tarifPerHari) {

super(nomorPolisi);

this.nomorPolisi = nomorPolisi;

this.merkMobil = merkMobil;

this.warna = warna;

this.tarifPerHari = tarifPerHari;

}

public String getNomorPolisi() {

return nomorPolisi;

}

public String getMerkMobil() {

return merkMobil;

}

public String getWarna() {

return warna;

}

public double getTarifPerHari() {

return tarifPerHari;

}

@Override

public void tampilkanInfo() {

System.out.println("--- Informasi Mobil ---");

System.out.println("Nomor Polisi: " + nomorPolisi);

System.out.println("Merk Mobil: " + merkMobil);

System.out.println("Warna: " + warna);

System.out.println("Tarif Per Hari: Rp " + tarifPerHari);

}

}

// Class Rental

class Rental {

private Mobil mobil;

private Supir supir;

private LocalDateTime waktuPinjam;

private LocalDateTime waktuKembali;

public void pinjamMobil(Supir supir, Mobil mobil) {

this.supir = supir;

this.mobil = mobil;

this.waktuPinjam = LocalDateTime.now();

System.out.println("Mobil " + mobil.getNomorPolisi() + " dipinjam oleh Supir " + supir.getNama() + " pada " + waktuPinjam);

}

public void kembalikanMobil() {

this.waktuKembali = LocalDateTime.now();

long durasiHari = ChronoUnit.DAYS.between(waktuPinjam, waktuKembali);

double biaya = hitungBiayaRental(durasiHari);

System.out.println("Mobil " + mobil.getNomorPolisi() + " dikembalikan pada " + waktuKembali);

System.out.println("Durasi Peminjaman: " + durasiHari + " hari");

System.out.println("Total Biaya Rental: Rp " + biaya);

this.mobil = null;

this.supir = null;

this.waktuPinjam = null;

this.waktuKembali = null;

}

public double hitungBiayaRental(long durasi) {

if (mobil != null) {

return durasi \* mobil.getTarifPerHari();

} else {

System.out.println("Error: Mobil belum dipinjam.");

return 0;

}

}

public Mobil getMobil() {

return mobil;

}

public Supir getSupir() {

return supir;

}

public LocalDateTime getWaktuPinjam() {

return waktuPinjam;

}

public LocalDateTime getWaktuKembali() {

return waktuKembali;

}

}

public class MainRentalMobil {

public static void main(String[] args) {

Supir supir1 = new Supir("12345", "Joko Widodo");

Mobil mobil1 = new Mobil("B 123 AB", "Toyota Avanza", "Hitam", 250000);

Mobil mobil2 = new Mobil("D 456 CD", "Honda Jazz", "Merah", 200000);

Rental transaksi1 = new Rental();

transaksi1.pinjamMobil(supir1, mobil1);

// Simulasi 3 hari kemudian

try {

Thread.sleep(3 \* 24 \* 60 \* 60 \* 1000);

} catch (InterruptedException e) {

Thread.currentThread().interrupt();

}

transaksi1.kembalikanMobil();

System.out.println("\n--- Informasi Setelah Transaksi ---");

supir1.tampilkanInfo();

mobil1.tampilkanInfo();

}

}