```
class ParkingLotFeeManagement {

    int totalParkingSpaces = 2;

    public int trunkParkingFeeCalculator(Trunk trunk) {

        int fee = trunk.getHours() * 10;

        if (fee > 120)
            return 120;
        else
            return fee;
    }

    public int carParkingFeeCalculator(Car car) {
        int fee = car.getHours() * 5;

        if (fee > 60)
            return 60;
        else
            return fee;
    }

    public void setCarParkingHours(Car car, int hours) throws Exception {
        if (isAvailable()) {
            car.setHours(hours);
            totalParkingSpaces--;
        } else {
            throw new Exception("no more parking space");
        }
    }

    public void setTrunkParkingHours(Trunk trunk, int hours) throws Exception {
        if(isAvailable()) {
            trunk.setHours(hours);
            totalParkingSpaces--;
        } else
            throw new Exception("no more parking space");
    }

    public void remove(Model model) {
        model.setHours(0);
        totalParkingSpaces++;
    }

    public boolean isAvailable() {
        return totalParkingSpaces >= 1;
    }
}

abstract class Model {
    int hours;

    public abstract int getHours();
    public abstract void setHours(int hours);
}

class Car extends Model{

    public Car(){}
    public Car(int hours) {
        this.hours = hours;
    }

    @Override
    public int getHours() {
        return hours;
    }

    @Override
    public void setHours(int hours) {
        this.hours = hours;
    }

}

class Trunk extends Model {

    public Trunk(){}
    public Trunk(int hours) {
        this.hours = hours;
    }

    @Override
    public int getHours() {
        return hours;
    }

    @Override
    public void setHours(int hours) {
        this.hours = hours;
    }
}

public class SystemDotOutDotPrintlnTest {

    public static void main(String args[]) throws Exception {
        Trunk trunk1 = new Trunk(0);  // before entering the parking lot
        Car car1 = new Car(0);  // before entering the parking lot

        ParkingLotFeeManagement feeManagement = new ParkingLotFeeManagement();

        // after entering the parking lot
        feeManagement.setTrunkParkingHours(trunk1, 5);
        // about exiting the parking lot
        System.out.println(feeManagement.trunkParkingFeeCalculator(trunk1));

        feeManagement.setCarParkingHours(car1, 10);
        System.out.println(feeManagement.carParkingFeeCalculator(car1));

        feeManagement.remove(trunk1);
        feeManagement.remove(car1);
    }

}
```