

src\final_exam\week12\quiz_3\CarDemo.java

```

1 package easyjava.final_exam.week12.quiz_3;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 // CarConsumer 인터페이스: Car 객체를 받아 처리하는 메소드 정의
7 interface CarConsumer {
8     void apply(Car car);
9 }
10
11 // CarPredicate 인터페이스: Car 객체를 테스트하는 메소드 정의
12 interface CarPredicate {
13     boolean test(Car car);
14 }
15
16 public class CarDemo {
17     public static void main(String[] args) {
18         // 디젤 자동차 목록 찾기
19         List<Car> dieselCars = findCars(Car.cars, c -> !c.isGasoline());
20         System.out.println("디젤 자동차 = " + dieselCars);
21
22         // 오래된 자동차 목록 찾기 (10년 이상)
23         List<Car> oldCars = findCars(Car.cars, c -> c.getAge() > 10);
24         System.out.println("오래된 자동차 = " + oldCars);
25
26         // 오래된 디젤 자동차 목록 찾기 (10년 이상이고 디젤인 경우)
27         List<Car> oldDieselCars = findCars(Car.cars, c -> c.getAge() > 10 && !c.isGasoline());
28         System.out.println("오래된 디젤 자동차 = " + oldDieselCars);
29
30         // 디젤 자동차 목록 출력
31         System.out.print("디젤 자동차 = ");
32         printCars(dieselCars, c -> System.out.printf("%s(%d) ", c.getModel(), c.getAge()));
33
34         // 오래된 자동차 목록 출력
35         System.out.print("\n오래된 자동차 = ");
36         printCars(oldCars, c -> System.out.printf("%s(%d, %d) ", c.getModel(), c.getAge(),
37             c.getMileage()));
38     }
39
40     // 주어진 조건에 맞는 자동차 목록 찾기
41     public static List<Car> findCars(List<Car> all, CarPredicate cp) {
42         List<Car> result = new ArrayList<>();
43
44         // 모든 자동차를 순회하면서 조건에 맞는 자동차를 결과 리스트에 추가
45         for (Car car : all) {
46             if (cp.test(car))
47                 result.add(car);
48         }
49         return result;
50     }
51
52     // 주어진 소비자 조건에 따라 자동차 목록 출력
53     public static void printCars(List<Car> all, CarConsumer cc) {
54         // 모든 자동차를 순회하면서 소비자 조건에 따라 처리
55         for (Car car : all) {
56             cc.apply(car);
57         }
58     }
59 }

```

```
57     }  
58 }  
59
```