


```

1  import org.w3c.dom.NameList;
2  import java.util.ArrayList;
3  import java.util.List;
4  import java.util.Map;
5  import java.util.stream.Collectors;
6
7  class Vehicle {
8      // Implement the Vehicle with constructor and getter setter method...
9      String name;
10     double price;
11     public Vehicle(String name, double price){
12         super();
13         this.name=name;
14         this.price=price;
15     }
16     public String getName(){
17         return name;
18     }
19     public void setName(String name){
20         this.name=name;
21     }
22     public Double getPrice(){
23         return price;
24     }
25     public void setPrice(Double price){
26         this.price=price;
27     }
28 }
29 class VehicleImplementation {
30     // Implement the VehicleImplementation method...
31     public double sumOfPrices(List<Vehicle>list)
32     {
33         Double sum=list.stream()
34         .mapToDouble(x->x.price)
35         .sum();
36         return sum;
37     }

```

Autocomplete reconnecting... 

Test Results

Custom Input

```
1 class Vehicle {
2     private String name;
3     private double price;
4     public Vehicle(String name, double price) {
5         super();
6         this.name = name;
7         this.price = price;
8     }
9 }
10 class VehicleImplementation {
11     public double sumOfPrices(List<Vehicle> list){
12         Double sum = list.stream()
13             .mapToDouble(x -> x.cost)
14             .sum();
15         return sum;
16     }
17
18
19     public List<String> getVehicleList(List<Vehicle> list) {
20         return list.stream()
21             .filter(x -> x.cost>25000)
22             .map(x -> x.name)
23             .collect(Collectors.toList());
24     }
25     public double maxPrice(List<Vehicle> list) {
26         Double sum = list.stream()
27             .mapToDouble(x -> x.cost)
28             .max().orElse(0.0);
29         return sum;
30     }
31 }
32 }
```

Normal text file

Type here to search