```
31          public String toString() {
32              return "Product{" +
33                      "productName='" + productName + '\'' +
34                      ", productCost=" + productCost +
35                      '}';
36          }
37      }
38
39      class ProductImplementation {
40          public Integer summationOfCost(List<Product> list){
41              Integer sumCost=list.stream()
42              .filter(s->s.getProductCost()>25)
43              .mapToInt(Product::getProductCost)
44
45              //.collect(Collectors.toList())
46              .sum();
47              return sumCost;
48          }
49          public List<Product> sortByCost(List<Product> list){
50              //list.stream().sorted((a,b) ->(a.getProductCost()-b.getProductCost));
51              Collections.sort(list,new Comparator<Product>(){
52                  public int compare(Product x, Product y){
53                      return x.getProductCost()-y.getProductCost();
54                  }
55              });
56              return list;
57          }
58      }
59
60      public class Source {
            public static void main(String args[] ) throws Exception {
```

Autocomplete reconnecting...

Ln 1, Col 1 Java 8

Test Results Custom Input

RUN CODE

SUBMIT

Help

All

```
List<Product> products =                    <>();
        products.                           (       ));
        products.                           (       ));
        products.                           (       ))
        products.                           (       ));
```

Sample Output

```
260
--Output for summationOfCost method--
[Product{productName='Lemon', productCost=15},
--Output for sortByCost method--
```

NOTE

- You can make suitable function calls and use RUN CODE button to check your main() method output.

Execution time limit

10 seconds

REPORT AN ISSUE

```
1  import java.io.*;
2  import java.util.*;
3  import java.util.stream.*;
4  import java.text.*;
5  import java.math.*;
6  import java.util.regex.*;
7
8  class Product{
9      private String productName;
10     private Integer productCost;
11
12     public Product(String productName, Integer productCost){
13         this.productName=productName;
14         this.productCost=productCost;
15     }
16     public void setProductName(String productName){
17         this.productName=productName;
18     }
19     public void setProductCost(Integer productCost){
20         this.productCost=productCost;
21     }
22     public String getProductName(){
23         return productName;
24     }
25     public Integer getProductCost(){
26         return productCost;
27     }
28
29     @Override
30     public String toString() {
31         return "Product{" +
32                 "productName='" + productName + '\'' +
33                 ", productCost=" + productCost +
```

Autocomplete reconnecting...

Ln 1, Col 1 Java 8

Test Results | Custom Input

RUN CODE | SUBMIT

29m 28s left

Help

All

1

2

3

4

- define parameterized constructor, getters and

Class ProductImplementation

Implement the below methods for this class using StreamApi:

1. Integer summationOfCost(List<Product> list):
   return the sum of costs above 25. Return zero if there is no product with cost above 25.
2. List<Product> sortByCost(List<Product> product) :
   return sorted list by cost(Refer to sample Output for clarity)

Implement using Lambda expressions and Stream API.

Note

- Following has been done for you:
- String toString() method, it's part of code stub, don't edit it else your *test-cases might fail*

Sample Input

```
List<Product> products = new ArrayList<>();
        products.add(new Product("Lemon", 15));
        products.add(new Product("Honey", 60));
        products.add(new Product("Sauce", 200));
        products.add(new Product("Egg", 25));
```

Sample Output

Java 8

```
1  > impo
10     pr
11
12     pu
13
14
15     }
16     pu
17
18     }
19     pu
20
21     }
22     p
23
24     }
25     p
26
27     }
28
29
30
31
32
33
35
36   }
37
38   cl
39
40
41
```

Autocom

29m 33s left

Help

All

```
        visibility: public
    Implement getters and setters
        visibility: public

class ProductImplementation:
  method definitions:
      summationOfCost(List<Product> list):
          return type: Integer
          visibilty: public

      sortByCost(List<Product> list):
          return type: List<Product>
          visibilty: public
```

Task:

Class Product

- define the String variable productName
- define the Integer variable productCost
- define parameterized constructor, getters and setters

Class ProductImplementation

Implement the below methods for this class using StreamApi:

1. Integer summationOfCost(List<Product> list):
   return the sum of costs above 25. Return zero if there is no product with cost above 25
2. List<Product> sortByCost(List<Product> product):
   return sorted list by cost(Refer to sample Output for

29m 40s left

Help

All

3. Product in Stock- I

Coding

Description

Your task here is to implement a Java code based on the following specifications. Note that your code should match the specifications in a precise manner. Consider default visibility of classes, data fields and methods unless mentioned otherwise.

Specifications:

```
class definitions:
 class Product:
     data fields:
         productName: String
           visibility: private
         productCost: Integer variable
           visibility: private

method definitions:
       Define parameterized constructor:
          Product(String productName, Integer pr
            visibility: public
       Implement getters and setters
          visibility: public

class ProductImplementation:
 method definitions:
     summationOfCost(List<Product> list):
         return type: Integer
```