

# AnalyticEngine.java

```
1 package analytics;
2
3 import java.util.Map;
4
5 public class AnalyticEngine
6 {
7     //list of attribute name
8     public static final String ORDER = "order";
9     public static final String CLIENTS = "clients";
10
11     private static AnalyticEngine instance = null;
12
13     private AnalyticEngine()
14     {
15     }
16
17     public static AnalyticEngine getInstance()
18     {
19         if (instance == null)
20             instance = new AnalyticEngine();
21         return instance;
22     }
23
24     public double computeAveAddItemInterval(Map<String, UserActivity> users)
25     {
26         long totalTime = 0;
27         int numberOfIntervals = 0;
28         for(UserActivity user : users.values())
29         {
30             long last = user.startTime;
31             int user_c = user.counter;
32             if (user_c > 0)
33             {
34                 long[] timeLog = user.addItemTime;
35                 for(int i = 0; i < user_c; i++)
36                 {
37                     totalTime = totalTime + timeLog[i] - last;
38                     last = timeLog[i];
39                     numberOfIntervals ++;
40                 }
41             }
42         }
43         if (numberOfIntervals < 1 ) //in case there is no one has entered yet.
44             return -1;
45         double ave = (double)totalTime/1000/numberOfIntervals;
46         return ave;
47     }
48 }
```

AnalyticEngine.java

```
49 public double computeCheckOutInterval(Map<String, UserActivity> users)
50 {
51     long totalTime = 0;
52     int numberOfUsers = 0;
53     for(UserActivity user: users.values())
54     {
55         long first = user.startTime;
56         long checkOut = user.checkoutTime;
57         if(checkOut > 0 )
58         {
59             totalTime = totalTime + checkOut - first;
60             numberOfUsers ++;
61         }
62     }
63     if (numberOfUsers < 1 ) //in case there is no one has entered yet.
64         return -1;
65     return (double)totalTime / numberOfUsers;
66 }
67
68
69
70
71 }
```