Bhavin Chavda

[bhavinchavda096@gmail.com](mailto:bhavinchavda096@gmail.com)

+91 9898418818

**Bluemingo Assessment Report**

**Used Data Structures:**

* List
* Map

**Classes created for the solution(OOP Concept):**

* OrderManagement
* GradeDetail
* GradeMix
* Order

**Exception Handling:**

* Parsing Dates

Click Here to get Full Java Code:

GitHub Link : [Bluemingo\_Assesment](https://github.com/Bhavin-Chavda/Bluemingo_Assesment)

**Module Wise Explanation**

**1)Input Data**

Order.java 🡺 This class will be used to store order Details, initially we will assign given details from Orders main data and rest of the field will be calculated later, this class itself store the final output

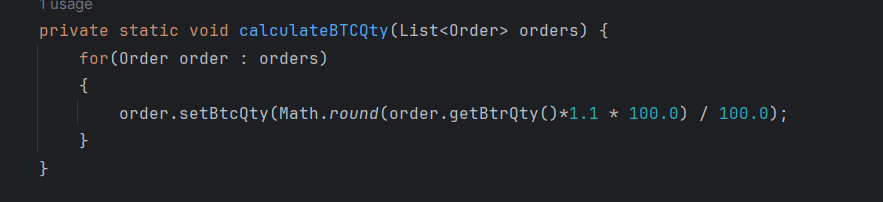
GradeDetail.java 🡺 This class will be used to map GradeDetail data as given

GradeMix.java 🡺 This class will be used to map GradeMix data as given

OrderManagement.java 🡺 This cis our main class from where the execution begin, all the calculation for final output will be done here by creating separate methods for each requirements

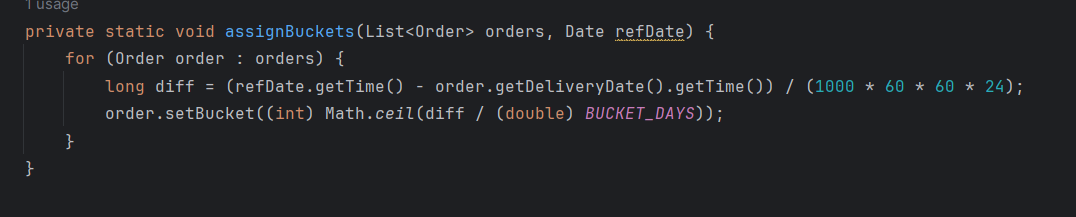
**2)Calculation Functions**

1. **Calculate BTC\_Qty = BTR\_Qty \* 1.1**

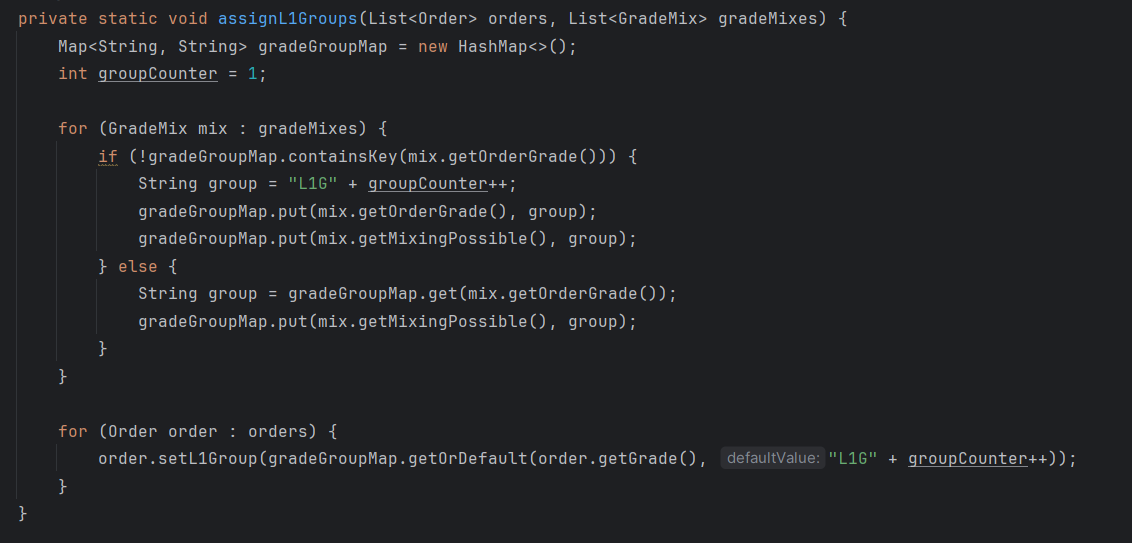


1. **Calculate Order Bucket**

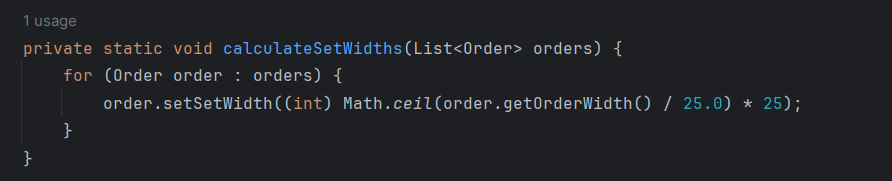
****

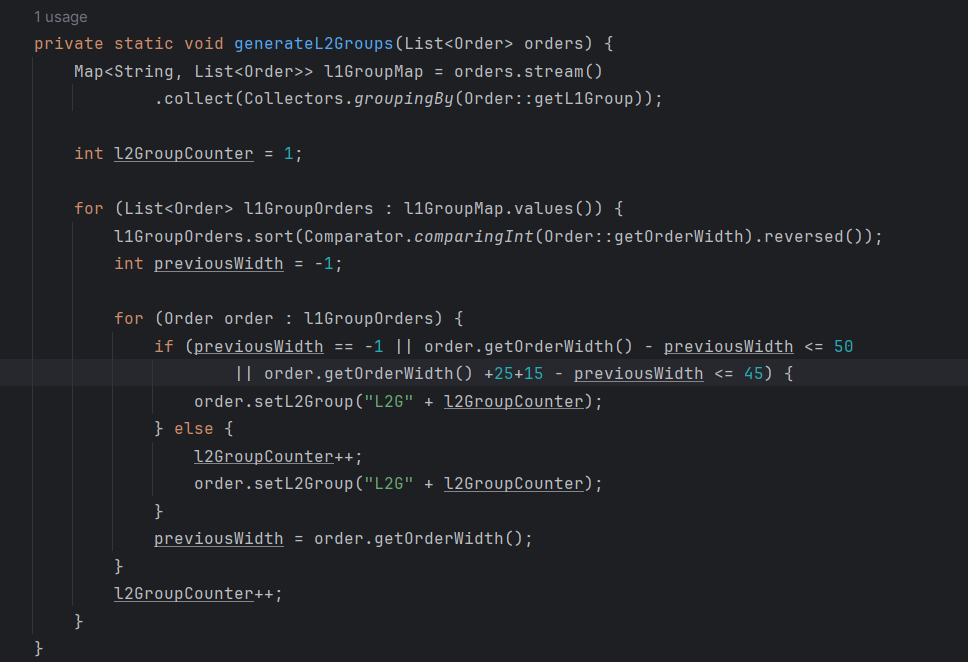
****

1. **Generate Level-1 Group**

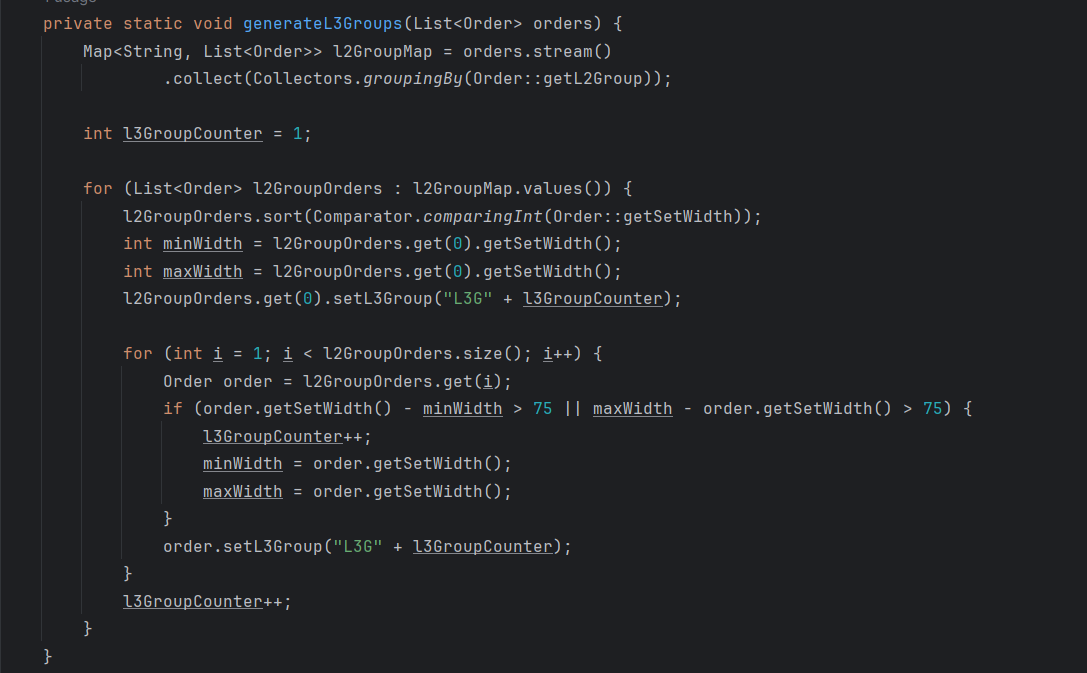
****

1. **Generate Level-2 Group and Set Width**

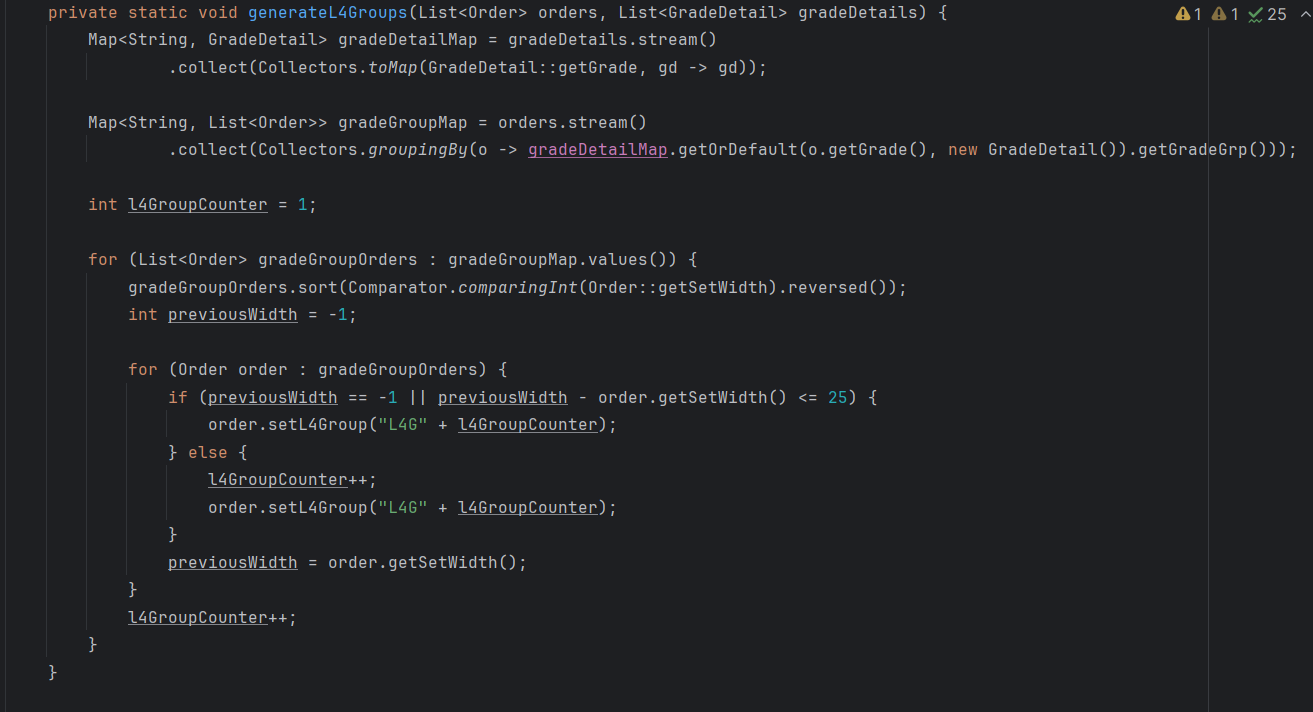
****

****

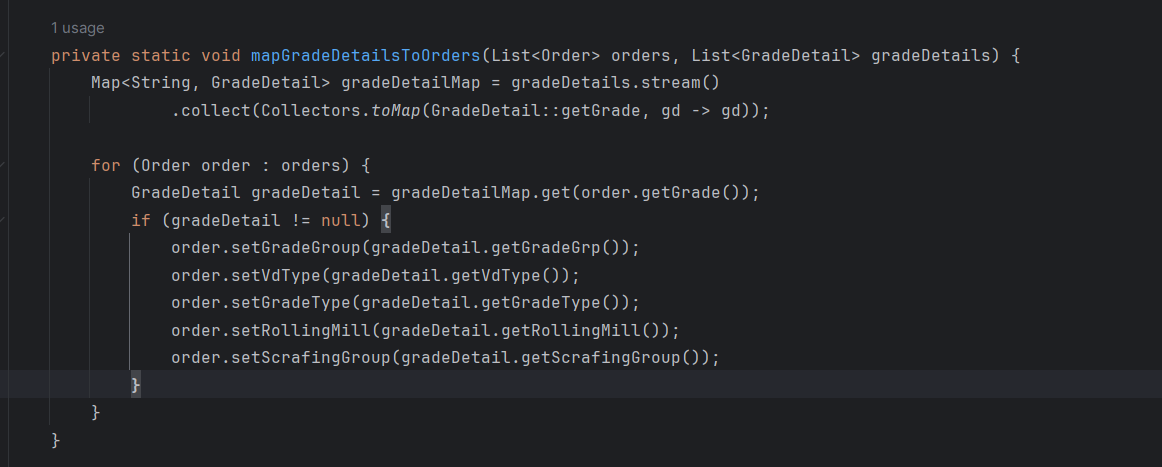
1. **Generate Level-3 Group**

****

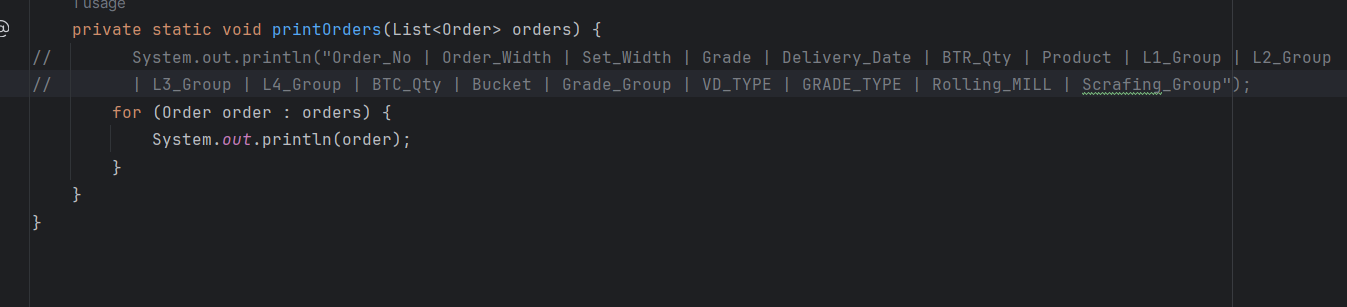
1. **Generate Level-4 Group**

****

1. **MapGradeDetailsToOrders**

****

1. **Print Final Output**

****

1. **Final Output(in CSV format) (also csv file attached)**

Order\_No,Order\_Width,Set\_Width,Grade,Delivery\_Date,BTR\_Qty,Product,L1\_Group,L2\_Group,L3\_Group,L4\_Group,BTC\_Qty,Bucket,Grade\_Group,VD\_TYPE,GRADE\_TYPE,Rolling\_MILL,Scrafing\_Group

1,1200,1200,ABCD1,10-07-2024,48.0,P6,L1G1,L2G4,L3G2,L4G1,52.8,2,AG15,VD,DG6,M2,G1

2,1175,1175,ABCD2,15-07-2024,48.0,P6,L1G2,L2G3,L3G3,L4G1,52.8,0,AG15,VD,DG6,M2,G1

3,1160,1175,ABCD3,05-07-2024,350.0,P5,L1G3,L2G2,L3G5,L4G1,385.0,4,AG15,VD,DG3,M2,G1

4,1122,1125,ABCD4,05-07-2024,350.0,P5,L1G4,L2G1,L3G7,L4G2,385.0,4,AG15,VD,DG3,M2,G3

5,1100,1100,ABCD5,12-07-2024,40.0,P2,L1G1,L2G4,L3G1,L4G2,44.0,1,AG15,VD,DG3,M2,G3

5,1080,1100,ABCD6,12-07-2024,40.0,P2,L1G2,L2G3,L3G3,L4G2,44.0,1,AG15,VD,DG3,M2,G3

5,975,975,ABCD7,12-07-2024,40.0,P2,L1G3,L2G2,L3G4,L4G3,44.0,1,AG15,VD,DG3,M2,G3

5,925,925,ABCD8,12-07-2024,40.0,P2,L1G4,L2G1,L3G6,L4G4,44.0,1,AG15,VD,DG3,M2,G3

**Assumptions Taken**

* The input data is provided in a format that can be easily parsed.
* The delivery date and other dates are formatted consistently.
* The reference date for buckets and other configurations are set as constants in the code.
* The grade details and grade mix details cover all possible grades in the orders.

**Limitations and Extension**   
  
I have used minimal dataset to test the functionalities due to strict deadline shown in below screenshot and code is runnable with bigger dataset as well by including it in dataset and for bigger dataset we might also need to handle more exception and can be easily handled.

I have tried to take data to accordingly to cover multiple scenarioes

List<Order> orders = Arrays.*asList*(  
 new Order(1, 1200, "ABCD1", *sdf*.parse("10-07-2024"), 48, "P6"),  
 new Order(2, 1175, "ABCD2", *sdf*.parse("15-07-2024"), 48, "P6"),  
 new Order(3, 1160, "ABCD3", *sdf*.parse("05-07-2024"), 350, "P5"),  
 new Order(4, 1122, "ABCD4", *sdf*.parse("05-07-2024"), 350, "P5"),  
 new Order(5, 1100, "ABCD5", *sdf*.parse("12-07-2024"), 40, "P2"),  
 new Order(5, 1080, "ABCD6", *sdf*.parse("12-07-2024"), 40, "P2"),  
 new Order(5, 975, "ABCD7", *sdf*.parse("12-07-2024"), 40, "P2"),  
 new Order(5, 925, "ABCD8", *sdf*.parse("12-07-2024"), 40, "P2")  
);  
  
List<GradeDetail> gradeDetails = Arrays.*asList*(  
 new GradeDetail("ABCD1", "AG15", "PG18", "VD", "DG6", "M2", "G1"),  
 new GradeDetail("ABCD2", "AG15", "PG18", "VD", "DG6", "M2", "G1"),  
 new GradeDetail("ABCD3", "AG15", "PG18", "VD", "DG3", "M2", "G1"),  
 new GradeDetail("ABCD4", "AG15", "PG18", "VD", "DG3", "M2", "G3")  
 ,new GradeDetail("ABCD5", "AG15", "PG18", "VD", "DG3", "M2", "G3")  
 ,new GradeDetail("ABCD6", "AG15", "PG18", "VD", "DG3", "M2", "G3")  
 ,new GradeDetail("ABCD7", "AG15", "PG18", "VD", "DG3", "M2", "G3")  
 ,new GradeDetail("ABCD8", "AG15", "PG18", "VD", "DG3", "M2", "G3")  
);  
  
List<GradeMix> gradeMixes = Arrays.*asList*(  
 new GradeMix("ABCD1", "ABCD5"),  
 new GradeMix("ABCD2", "ABCD6"),  
 new GradeMix("ABCD3", "ABCD7"),  
 new GradeMix("ABCD4", "ABCD8")  
);

There might be some miss understanding in calculating functions, as all the functions are created as per my understanding of logic from the given assessment and can be easily modified and after discussing the correct logic, also I have pushed the code on github that is easily runnable by cloning the repository to your local system.

Thanks for giving this opportunity.