Task

1. Implement 3 Classes as follows:

```
class OrdersAnalyzer {
        class Order(int orderId, LocalDateTime creationDate, List<OrderLine> orderliness) ,
        class OrderLine(int productId, String name, int quantity, BigDecimal   unitPrice ) ,

        *function* Map<DayOfWeek, Integer>  totalDailySales(List<Order> orders): {
//TODO("implement the function")
                }
        }
```

2. In Class `OrdersAnalyzer` Implement `totalDailySales` method that returns a map with a `DayOfWeek` as a key and the sum of ordered quantity for all products, as a value.

3. The method should always return a map. If there are no orders for some weekdays, then skip this day entry in the map.

Example Output

Your implementation should return the following output for example data:
```
{
"DayOfWeek.MONDAY" : 17 ,
"DayOfWeek.SATURDAY" : 6 ,
"DayOfWeek.SUNDAY" : 9
}
```
In the example above, there are three orders placed on Saturday: first with a quantity equal to 3 and second with quantity equal to 2 , and third with a quantity equal to 1 so the sum of quantities is 6 . For Monday the total quantity is equal to 17 , and for Sunday it's 9 .

Example Input

Orders collection:
```
[
{
orderId: 554 ,
creationDate: "2017-03-25T10:35:20" , // Saturday
```

```
orderLines: [
{productId: 9872 , name: 'Pencil' , quantity: 3 , unitPrice: 3.00 }
]
},
{
orderId: 555 ,
creationDate: "2017-03-25T11:24:20" , // Saturday
orderLines: [
{productId: 9872 , name: 'Pencil' , quantity: 2 , unitPrice: 3.00 },
{productId: 1746 , name: 'Eraser' , quantity: 1 , unitPrice: 1.00 }
]
},
{
orderId: 453 ,
creationDate: "2017-03-27T14:53:12" , // Monday
orderLines: [
{productId: 5723 , name: 'Pen' , quantity: 4 , unitPrice: 4.22 },
{productId: 9872 , name: 'Pencil' , quantity: 3 , unitPrice: 3.12 },
{productId: 3433 , name: 'Erasers Set' , quantity: 1 , unitPrice: 6.15 }
]
},
{
orderId: 431 ,
creationDate: "2017-03-20T12:15:02" , // Monday
orderLines: [
{productId: 5723 , name: 'Pen' , quantity: 7 , unitPrice: 4.22 },
{productId: 3433 , name: 'Erasers Set' , quantity: 2 , unitPrice: 6.15 }
]
},
{
orderId: 690 ,
creationDate: "2017-03-26T11:14:00" , // Sunday
orderLines: [
{productId: 9872 , name: 'Pencil' , quantity: 4 , unitPrice: 3.12 },
{productId: 4098 , name: 'Marker' , quantity: 5 , unitPrice: 4.50 }
]
}
];
```