

```

1 public class HarryPotterBookstore {
2
3     Integer DEFAULT_PRICE = 8;
4
5     Integer MAIN_LEVEL_DISCOUNT = 5;
6
7     public double buy(Order order) {
8         if (order.getBooks().size() == 1) {
9             return DEFAULT_PRICE * order.getBooks().firstEntry().getValue();
10        }
11        return getTotalPrice(order);
12    }
13
14    private double getTotalPrice(Order order) {
15        MAIN_LEVEL_DISCOUNT = order.getBooks().size();
16        List<Double> results = new ArrayList<>();
17        for (int i = 0; i < MAIN_LEVEL_DISCOUNT; i++) {
18            LinkedHashMap<String, Integer> basket = order.getBooks().entrySet()
19                .stream()
20                .sorted(Map.Entry.comparingByValue((o1, o2) -> (-1) *
21                    o1.compareTo(o2)))
22                .collect(Collectors.toMap(Map.Entry::getKey, Map.Entry::getValue,
23                    (e1, e2) -> e1, LinkedHashMap::new));
24            double total = 0;
25            while (!basket.isEmpty()) {
26                Integer discountLevel = calculateDiscountLevel(basket);
27                double discount = getDiscount(discountLevel);
28                double price = getPriceForBiggestDiscount(basket, discountLevel);
29                total += price - (discount * price);
30            }
31            MAIN_LEVEL_DISCOUNT--;
32            results.add(total);
33        }
34        return Collections.min(results);
35    }
36
37    private double getPriceForBiggestDiscount(LinkedHashMap<String, Integer> basket,
38        Integer discountLevel) {
39        double price = 0;
40        Iterator<Map.Entry<String, Integer>> iterator = basket.entrySet().iterator();
41        for (int i = 0; i < discountLevel; i++) {
42            Map.Entry<String, Integer> basketItem = iterator.next();
43            basketItem.setValue(basketItem.getValue() - 1);
44            price += DEFAULT_PRICE;
45            if (basketItem.getValue() == 0) {
46                iterator.remove();
47            }
48        }
49        return price;
50    }
51
52    private Integer calculateDiscountLevel(LinkedHashMap<String, Integer> basket) {
53        if (MAIN_LEVEL_DISCOUNT <= basket.size()) {
54            return MAIN_LEVEL_DISCOUNT;
55        }
56        return basket.size();
57    }
58
59    private double getDiscount(Integer discountLevel) {
60        if (discountLevel == 2) {
61            return 0.05;
62        }
63        if (discountLevel == 3) {
64            return 0.1;
65        }
66        if (discountLevel == 4) {
67            return 0.2;
68        }
69        if (discountLevel == 5) {
70            return 0.25;
71        }
72        return 0;
73    }
74

```

