

```

1  /*
2  * To change this license header, choose License Headers in Project
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package fr.insalyon.dasi.gustatif.test;
7
8  import fr.insalyon.dasi.gustatif.metier.modele.Client;
9  import fr.insalyon.dasi.gustatif.metier.modele.Commande;
10 import fr.insalyon.dasi.gustatif.metier.modele.LigneDeCommande;
11 import fr.insalyon.dasi.gustatif.metier.modele.Livreur;
12 import fr.insalyon.dasi.gustatif.metier.modele.Restaurant;
13 import fr.insalyon.dasi.gustatif.metier.service.ServiceMetier;
14 import java.util.ArrayList;
15 import java.util.List;
16
17 /**
18  *
19  * @author Nico
20  */
21 public class CreationLivraisonSansInteraction {
22     private static final ServiceMetier serviceMetier = new
23         ServiceMetier();
24
25     public static void main(String[] args) {
26         // Créer une classe avec une méthode « main » permettant de
27         // tester
28         // la création d'une nouvelle livraison sans interaction avec
29         // l'utilisateur
30         // (valeurs des paramètres « en dur » dans le code).
31
32         List<Client> clients = serviceMetier.findAllClients();
33         if(clients == null) {
34             System.err.println("Aucun client dans la base de donnée. "
35                 + "Insérez en pour exécuter ce code.");
36             return;
37         }
38         Client client = clients.get(0);
39
40         List<Livreur> livreurs = serviceMetier.findAllLivreurs();
41         if(livreurs == null) {
42             System.err.println("Aucun livreur dans la base de donnée. "
43                 + "Insérez en pour exécuter ce code.");
44             return;
45         }
46         List<Restaurant> restaurants =
47             serviceMetier.findAllRestaurants();
48         if(restaurants == null) {
49             System.err.println("Aucun restaurant dans la base de
50                 donnée. "
51                 + "Insérez en pour exécuter ce code.");
52             return;
53         }
54     }
55 }

```

```
49     Restaurant restaurant = restaurants.get(0);
50     if(restaurant.getProduits().isEmpty()) {
51         System.err.println("Aucun produits pour le restaurant (id="
52             + restaurant.getId() + " dans la base de donnée.
53             Insérez en "
54             + "pour exécuter ce code.");
55         return;
56     }
57     Commande commande = new Commande(restaurant, client);
58     List<LigneDeCommande> lignesDeCommande = new
59     ArrayList<LigneDeCommande>();
60
61     lignesDeCommande.add(new
62     LigneDeCommande(restaurant.getProduits().get(0), new
63     Integer(1)));
64     commande.setLignesDeCommande(lignesDeCommande);
65
66     livreurs = serviceMetier.findAllLivresDisponibles();
67     boolean result = false;
68     for(Livreur l : livreurs) {
69         if(l.getChargeMaxEnGrammes() >=
70             commande.getPoidsEnGrammes()) {
71             result = true;
72         }
73     }
74     if(!result) {
75         System.err.println("Aucun livreur disponible pouvant
76         effectuer cette "
77         + "livraison (poids = " +
78         commande.getPoidsEnGrammes()
79         + "g) dans la base de donnée. Insérez en pour
80         exécuter ce code.");
81         return;
82     }
83     serviceMetier.createCommande(commande);
84 }
```