

Test

```

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.Commande;
9  import fr.insalyon.dasi.gustatif.metier.modele.Livreur;
10 import fr.insalyon.dasi.gustatif.metier.service.ServiceMetier;
11 import java.util.Date;
12 import java.util.List;
13 import java.util.Scanner;
14
15 /**
16  *
17  * @author Nico
18  */
19 public class ClotureLivraisonAvecInteraction {
20
21     private static final ServiceMetier serviceMetier = new
22     ServiceMetier();
23     private static final Scanner scanner = new Scanner(System.in);
24
25     public static void main(String[] args) {
26         // Créer une classe avec une méthode « main » permettant de
27         // clôturer d'une livraison avec interaction avec l'utilisateur
28         // (saisie de l'id du livreur ou de la livraison)
29         List<Commande> commandes =
30         serviceMetier.findAllCommandesNotEnded();
31
32         if(commandes == null) {
33             System.err.println("Aucune commande a cloturer.");
34             return;
35         }
36
37         String cmd;
38
39         do {
40             System.out.println("Cloturer une commande par id commande
41             : ic, "
42             + "Cloturer une commande par id livreur : il ,
43             lister commande"
44             + " non cloturée : lnc, quitter : q");
45             cmd = scanner.nextLine();
46             try {
47                 if(cmd.equals("ic")) {
48                     System.out.println("Saisir id commande : ");
49                     Long id = Long.parseLong(scanner.nextLine());
50                     Commande c = serviceMetier.findCommandeById(id);
51                     if(c != null) {

```

```
48         c.setDateFin(new Date());
49         serviceMetier.validateCommande(c);
50         Commande v = serviceMetier.findCommandeById(id);
51         System.out.println(v);
52     }
53     } else if(cmd.equals("il")) {
54         System.out.println("Saisir id livreur : ");
55         Long id = Long.parseLong(scanner.nextLine());
56         Livreur l = serviceMetier.findLivreurById(id);
57         Commande c = serviceMetier.findNotEndedByLivreur(l);
58         if(c != null) {
59             id = c.getId();
60             c.setDateFin(new Date());
61             serviceMetier.validateCommande(c);
62             Commande v = serviceMetier.findCommandeById(id);
63             System.out.println(v);
64         }
65     } else if(cmd.equals("lnc")) {
66         commandes = serviceMetier.findAllCommandesNotEnded();
67         for (Commande commande : commandes) {
68             System.out.println("\tCommande n°" + commande.getId()
69                                 + " - livreur n°" + commande.getLivreur().getId());
70         }
71     }
72     } catch (NumberFormatException e)
73     {
74     }
75     }
76     } while(!cmd.equals("q"));
77 }
78 }
79 }
```

```
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 }
```

```

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.dao.JpaUtil;
9  import fr.insalyon.dasi.gustatif.dao.LivreurDao;
10 import fr.insalyon.dasi.gustatif.metier.modele.Livreur;
11 import fr.insalyon.dasi.gustatif.metier.modele.LivreurDrone;
12 import fr.insalyon.dasi.gustatif.metier.modele.LivreurVelo;
13 import fr.insalyon.dasi.gustatif.util.GeoTest;
14 import java.util.logging.Level;
15 import java.util.logging.Logger;
16
17 /**
18  *
19  * @author Nico
20  */
21 public class JeuDeDonnees {
22     public static void main(String[] args) {
23         //créer des livreurs drones et velos, des commandes,
24         //cloturées non cloturées
25         //lancer le script sql??
26         Livreur lv1 = new LivreurVelo("Dupond", "Arthur",
27             "arthur@livreur.com",
28             "mdp", "145 Rue Anatole France, 69628
29             Villeurbanne", new Double(30000.F));
30         lv1.setCoordonnees(GeoTest.getLatLng(lv1.getAdresse()));
31         Livreur lv2 = new LivreurVelo("Durant", "Bob",
32             "bob@livreur.com", "mdp",
33             "30 Avenue Albert Einstein, 69100 Villeurbanne",
34             new Double(40000.F));
35         lv2.setCoordonnees(GeoTest.getLatLng(lv2.getAdresse()));
36         Livreur lv3 = new LivreurVelo("Petit", "Charlie",
37             "charlie@livreur.com",
38             "mdp", "Domaine de la Doua, 20 Avenue Gaston
39             Berger, 69100 Villeurbanne",
40             new Double(50000.F));
41         lv3.setCoordonnees(GeoTest.getLatLng(lv3.getAdresse()));
42         Livreur lv4 = new LivreurVelo("Martin", "Damien",
43             "estelle@livreur.com",
44             "mdp", "200 Rue Léon Blum, 69100 Villeurbanne",
45             new Double(20000.F));
46         lv4.setCoordonnees(GeoTest.getLatLng(lv4.getAdresse()));
47         Livreur lv5 = new LivreurVelo("Roux", "Estelle",
48             "estelle@livreur.com",
49             "mdp", "146 Boulevard Pinel, 69008 Lyon", new
50             Double(10000.F));
51         lv5.setCoordonnees(GeoTest.getLatLng(lv5.getAdresse()));
52         Livreur ld1 = new LivreurDrone(new Double(10.F),

```

```
43         "drone@gmail.com", "mdp",  
44         "17 Rue du Dr Bouchut, 69003 Lyon", new  
45         Double(1000.F));  
46         ld1.setCoordonnees(GeoTest.getLatLng(ld1.getAdresse()));  
47         Livreur ld2 = new LivreurDrone(new Double(10.F),  
48         "drone@gmail.com", "mdp",  
49         "17 Rue du Dr Bouchut, 69003 Lyon", new  
50         Double(1000.F));  
51         ld2.setCoordonnees(GeoTest.getLatLng(ld2.getAdresse()));  
52         Livreur ld3 = new LivreurDrone(new Double(10.F),  
53         "drone@gmail.com", "mdp",  
54         "17 Rue du Dr Bouchut, 69003 Lyon", new  
55         Double(1000.F));  
56         ld3.setCoordonnees(GeoTest.getLatLng(ld3.getAdresse()));  
57         Livreur ld4 = new LivreurDrone(new Double(10.F),  
58         "drone@gmail.com", "mdp",  
59         "17 Rue du Dr Bouchut, 69003 Lyon", new  
60         Double(1000.F));  
61         ld4.setCoordonnees(GeoTest.getLatLng(ld4.getAdresse()));  
62         Livreur ld5 = new LivreurDrone(new Double(10.F),  
63         "drone@gmail.com", "mdp",  
64         "17 Rue du Dr Bouchut, 69003 Lyon", new  
65         Double(1000.F));  
66         ld5.setCoordonnees(GeoTest.getLatLng(ld5.getAdresse()));  
67  
68         LivreurDao livreurDao = new LivreurDao();  
69         try{  
70             JpaUtil.creerEntityManager();  
71             JpaUtil.ouvrirTransaction();  
72  
73             livreurDao.create(lv1);  
74             livreurDao.create(lv2);  
75             livreurDao.create(lv3);  
76             livreurDao.create(lv4);  
77             livreurDao.create(lv5);  
78  
79             livreurDao.create(ld1);  
80             livreurDao.create(ld2);  
81             livreurDao.create(ld3);  
82             livreurDao.create(ld4);  
83             livreurDao.create(ld5);  
84  
85             JpaUtil.validerTransaction();  
86             JpaUtil.fermerEntityManager();  
87         } catch (Throwable ex) {  
88             Logger.getLogger(  
89                 JeuDeDonnees.class.getName()).log(Level.SEVERE,  
90                 null, ex);  
91         }  
92     }  
93 }
```