

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

import java.util.ArrayList;
import java.util.GregorianCalendar;
import java.util.List;
import predictif.modele.Client;
import predictif.modele.Medium;
import predictif.modele.Predictions.AmourPrediction;
import predictif.modele.Predictions.Prediction;
import predictif.modele.Predictions.SantePrediction;
import predictif.modele.Predictions.TravailPrediction;
import predictif.service.PrepareBD;
import predictif.service.Service;
import predictif.Util.Saisie;

/**
 *
 * @author Administrateur
 */
public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // Petite protection : si on ne sait pas quoi faire, affiche le message d'help
        if (args.length == 0) {
            System.out.println("Erreur, commande absente. Tapez 'aide' pour l'aide.");

            System.exit(-1); // arrête le programme
        }

        // Ces 2 lignes permettent de préparer la BD en "dur"
        PrepareBD serviceBD = new PrepareBD();
        serviceBD.populate();

        Service service = new Service();

        // Maintenant on va chercher à comprendre les arguments passés
        // malheureusement en Java, on ne peut pas faire un switch sur un String
        if (args[0].equals("creer-client") == true) {
            // On demande les champs "normaux"
            String nom = Saisie.lireChaine("Entrez le nom du client : ");
            String prenom = Saisie.lireChaine("Entrez le prenom du client : ");
            String adresse = Saisie.lireChaine("Entrez l'adresse : ");
            String email = Saisie.lireChaine("Entrez l'adresse email : ");
            String tel = Saisie.lireChaine("Entrez le numéro de téléphone : ");

            // La date de naissance demande plusieurs étapes
            String dateNaissance = Saisie.lireChaine("Entrez la date de naissance du client (format DD/MM/YYYY) : ");
            String date[] = dateNaissance.split("/");

            if (date.length != 3) {
                System.out.println("Erreur, date de naissance invalide");
                return;
            }

            GregorianCalendar calendar = new GregorianCalendar(Integer.parseInt(date[2]),
Integer.parseInt(date[1]), Integer.parseInt(date[0]));

            // Dernière étape : les médiums
            List<Medium> mediums = service.getAllMediums();

            for (int i = 0; i < mediums.size(); i++) {
                System.out.println("Medium n° " + i + " - " + mediums.get(i).getNom());
            }
        }
    }
}

```

```
    }
    String mediumsStr = Saisie.lireChaine("Entrez les numéros des médiums favoris du
client (séparés par des espaces) : ");
    String mediumsTab[] = mediumsStr.split(" ");

    List<Medium> mediumsFavoris = new ArrayList<Medium>();

    for (int i = 0; i < mediumsTab.length; i++) {
        mediumsFavoris.add(mediums.get(Integer.parseInt(mediumsTab[i]))); // un peu
long, récupère le médium dont on a le numéro
    }

    if (service.createClient(nom, prenom, adresse, email, tel, calendar,
mediumsFavoris)) {
        System.out.println("Création du client Ok");
    }
} else if (args[0].equals("modifier-client") == true) {
    // On liste les clients
    List<Client> clients = service.getAllClients();

    // Les affiche
    for (int i = 0; i < clients.size(); i++) {
        System.out.println("Client n° " + clients.get(i).getNumClient() + " - " +
clients.get(i).getPrenom() + " " + clients.get(i).getNom());
    }

    // On demande lequel on souhaite modifier
    String numStr = Saisie.lireChaine("Entrez le numéro du client à modifier : ");
    Client client = service.retrieveClient(Integer.parseInt(numStr));

    System.out.println("Nous allons vous demander les nouvelles valeurs. Pour ne pas
modifier la valeur, faites entrer.");

    // On demande les champs "normaux"
    String nom = Saisie.lireChaine("Entrez le nouveau nom du client : ");
    if (nom.equals("") == false) {
        client.setNom(nom);
    }

    String prenom = Saisie.lireChaine("Entrez le nouveau prenom du client : ");
    if (prenom.equals("") == false) {
        client.setPrenom(prenom);
    }

    String adresse = Saisie.lireChaine("Entrez la nouvelle adresse : ");
    if (adresse.equals("") == false) {
        client.setAdressePostale(adresse);
    }

    String email = Saisie.lireChaine("Entrez le nouvel email : ");
    if (email.equals("") == false) {
        client.setEmail(email);
    }

    String tel = Saisie.lireChaine("Entrez le nouveau numéro de téléphone : ");
    if (tel.equals("") == false) {
        client.setTel(tel);
    }

    // Mise à jour
    if (service.updateClient(client)) {
        System.out.println(client.getNom() + " mis à jour");
    } else {
        System.out.println("Mise à jour de " + client.getNom() + " ratée");
    }
} else if (args[0].equals("supprimer-client") == true) {
    // On liste les clients
    List<Client> clients = service.getAllClients();
```

```

// Les affiche
for (int i = 0; i < clients.size(); i++) {
    System.out.println("Client n° " + clients.get(i).getNumClient() + " - " +
clients.get(i).getPrenom() + " " + clients.get(i).getNom());
}

// On demande lequel on souhaite modifier
String numStr = Saisie.lireChaine("Entrez le numéro du client à supprimer : ");
Client client = service.retrieveClient(Integer.parseInt(numStr));

String confirm = Saisie.lireChaine("Êtes-vous certain de vouloir supprimer le
client " + client.getPrenom() + " " + client.getNom() + " ? o/n");

if (confirm.equals("o") == true) {
    service.deleteClient(client);
    System.out.println("Client supprimé");
} else {
    System.out.println("Modification annulée");
}
} else if (args[0].equals("aide") == true) {
    System.out.println("Bienvenue dans l'aide.");
    System.out.println("Les commandes suivantes sont disponibles (en mode interactif)
:");
    System.out.println("'creer-client' pour ajouter un nouveau client");
    System.out.println("'modifier-client' pour afficher la liste des clients et en
modifier un");
    System.out.println("'supprimer-client' pour afficher la liste des clients et en
supprimer un");
    System.out.println("'realiser-horoscope' pour ajouter un nouveau client");
} else if (args[0].equals("realiser-horoscope") == true) {
    // Tout d'abord, on demande à la personne de se connecter
    String numEmp = Saisie.lireChaine("Veuillez entrer votre numéro d'employé : ");
    String passEmp = Saisie.lireChaine("Entrez votre mot de passe : ");

    // Test de la connexion des employés
    if (service.connectEmploye(Integer.parseInt(numEmp), passEmp)) {
        System.out.println("Connexion ok");
    } else {
        System.out.println("Nom d'utilisateur ou mot de passe invalide");

        System.exit(-1); // arrête le programme
    }

    // On liste les clients
    List<Client> clients = service.getAllClients();

    // Les affiche
    for (int i = 0; i < clients.size(); i++) {
        System.out.println("Client n° " + clients.get(i).getNumClient() + " - " +
clients.get(i).getPrenom() + " " + clients.get(i).getNom());
    }

    // On demande lequel on souhaite modifier
    String numStr = Saisie.lireChaine("Entrez le numéro du client pour lequel vous
souhaitez réaliser un horoscope : ");
    Client client = service.retrieveClient(Integer.parseInt(numStr));

    List<Medium> mediumsFavoris = client.getMediumsFavoris();

    // Les affiche
    for (int i = 0; i < mediumsFavoris.size(); i++) {
        System.out.println("Médium n° " + i + " - " +
mediumsFavoris.get(i).getNom());
    }

    String signature = Saisie.lireChaine("Entrez le numéro du médium qui va réaliser
cet horoscope : ");

    // Récupération de tous les horoscopes

```

```
List<Prediction> amours = service.getPrediction(Service.PredictionType.AMOUR);
List<Prediction> travaux = service.getPrediction(Service.PredictionType.TRAVAIL);
List<Prediction> santes = service.getPrediction(Service.PredictionType.SANTE);

String predicAmour = Saisie.lireChaine("Entrez le numéro de la prédiction Amour :
");
String predicSante = Saisie.lireChaine("Entrez le numéro de la prédiction Santé :
");
String predicTravail = Saisie.lireChaine("Entrez le numéro de la prédiction
Travail : ");

service.createHoroscope((AmourPrediction)
amours.get(Integer.parseInt(predicAmour)), (TravailPrediction)
travaux.get(Integer.parseInt(predicTravail)), (SantePrediction)
santes.get(Integer.parseInt(predicSante)),
client.getMediumsFavoris().get(Integer.parseInt(signature)), client);

System.out.println("Horoscope créé !");
} else {
    System.out.println("Erreur, commande non reconnue. Tapez 'aide' pour l'aide.");
}
}
```