

```

/*****
 * @author Ayoub BEN KHIROUN && Zied BEN YOUSSEF
 * class Banque_locale_serveur
 *****/
*/

import java.util.*;

public class Banque_locale_serveur {

    ArrayList<Account> all_accounts; // Collection de comptes

    /**
     * Constructeur de la classe Banque_locale_serveur
     */
    public Banque_locale_serveur() {

        // Instanciation de la collection de comptes
        all_accounts = new ArrayList<Account>();

    }

    /**
     * Méthode d'ouverture d'un nouveau compte bancaire
     * @param name Nom du nouveau compte
     * @param password Mot de passe du nouveau compte
     */
    public void open_account(String name, String password) {

        if (verify(name, password) == null) {
            Account new_account = new Account(0, name, password);
            all_accounts.add(new_account);
            System.out.println("\t-- Le compte a été créé avec succès --");
        } else {
            System.out.println("\t-- Le compte existe déjà !! --");
        }

    }

    /**
     * Méthode de vérification de l'existence d'un compte
     * @param name Nom du compte à chercher
     * @param password Mot de passe du compte à chercher
     * @return (null | Account) Retourne le compte trouvé, sinon "null"
     */
    public Account verify(String name, String password) {

        int i = 0;
        Boolean test = false;
        Account new_account = null;
        while ((i < all_accounts.size()) && (!test)) {
            if ((all_accounts.get(i).name.equals(name))
                && (all_accounts.get(i).password.equals(password))) {
                test = true;
                new_account = (Account) all_accounts.get(i);
            } else {
                i++;
            }
        }
        return new_account;

    }

}

```

```

/**
 * Méthode de fermeture d'un compte
 * @param name Nom du compte à fermer
 * @param password Mot de passe du compte à fermer
 * @return (-1 | 0) Code d'erreur
 */
public int close_account(String name, String password) {

    int code_erreur = (-1);
    Account tempAcc = verify(name, password);
    if (tempAcc != null) {
        // Suppression du compte de la collection "all_accounts"
        all_accounts.remove(tempAcc);
        code_erreur = 0;
        System.out.println("\t-- Le compte a été supprimé avec succès --");
    } else
        System.out.println("\t-- Le compte n'existe pas !! --");
    return (code_erreur);

}

/**
 * Méthode de versement d'argent dans un compte bancaire
 * @param name Nom du compte
 * @param password Mot de passe du compte
 * @param money Montant à verser
 */
public void deposit(String name, String password, float money) {

    Account new_account = verify(name, password);
    if (new_account != null) {
        new_account.balance = new_account.balance + money;
    }

}

/**
 * Méthode de retrait d'argent
 * @param name Nom du compte
 * @param password Mot de passe du compte
 * @param amount Montant à retirer
 * @return (-1 | 0) Code d'erreur (succès de l'opération ou échec dans le
 * cas de compte introuvable ou solde insuffisant)
 */
public int withdraw(String name, String password, float amount) {

    int erreur_test = (-1);
    Account new_account = verify(name, password);
    if (new_account != null) {
        // Test si le solde est suffisant pour le retrait
        if (new_account.balance >= amount) {
            new_account.balance -= amount;
            erreur_test = 0;
        } else
            System.out
                .println("\t-- Votre demande est refusée, le solde de votre
compte n'est pas suffisant !! --");

    } else
        System.out.println("\t-- Le compte n'existe pas !! --");
    return (erreur_test);

}

```

```

/**
 * Méthode de consultation de solde
 * @param name Nom du compte
 * @param password Mot de passe du compte
 * @return (float | -1) Solde du compte, sinon si le compte est inexistant
 *         alors -1
 */
public float get_balance(String name, String password) {

    Account tempAcc = verify(name, password);
    if (tempAcc != null) {
        return (tempAcc.balance);
    } else {
        return (-1);
    }

}

/**
 * Méthode d'affichage de la liste des comptes sous forme d'un tableau *
 */
public void affiche_liste_comptes() {

    System.out.println(" _____ ");
    System.out.println("|NAME\t\t|PASSWORD\t|BALANCE\t|");
    System.out.println("| _____ | _____ |");
    for (int i = 0; i < all_accounts.size(); i++) {
        System.out.println("|" + all_accounts.get(i).name + "\t\t|"
            + all_accounts.get(i).password + "\t\t|"
            + all_accounts.get(i).balance + "\t\t|");
    }
    System.out.println("| _____ | _____ |");

}

}

```