

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP ĐỀ MINH HỌA THI GIỮA KÌ 02
THÀNH PHỐ HỒ CHÍ MINH Môn thi: Lập trình JAVA trong CNTT

KHOA CÔNG NGHỆ THÔNG TIN Lớp: DHTH18
Ngày thi: 2023
Thời gian: 70 phút

Họ tên: MSSV.....

Mô tả hệ thống

Một ngân hàng cần quản lý danh sách khách hàng. Thông tin đối tượng khách hàng gồm có mã số tài khoản khách hàng `customer_id`, họ tên `customer_name`, tài khoản `customer_account`, phân loại khách hàng `customer class`. Nếu tài khoản trên 500 triệu là khách hàng Gold, từ 100 triệu đến dưới 500 triệu là khách hàng bạc Silver, dưới 100 triệu là bình thường Normal.

Sinh viên sử dụng danh sách liên kết LinkedList viết chương trình có menu thực hiện các chức năng:

Câu 1: Đọc danh sách khách hàng từ file (`customer_id`, `customer_name`)

Câu 2: Lưu thông tin khách hàng ra file

Câu 3: Xuất thông tin danh sách khách hàng

Câu 4: Thêm khách hàng vào danh sách

Câu 5: Tìm kiếm khách hàng theo mã số tài khoản khách hàng

Câu 6: Sắp xếp danh sách khách hàng theo họ tên khách hàng

Câu 7: Xóa khách hàng có `customer_id` khỏi danh sách

-----**Hết**-----

1. Đọc danh sách khách hàng từ file

Ý tưởng:

- Sử dụng `ObjectInputStream` để đọc danh sách khách hàng từ file.
- File lưu trữ danh sách khách hàng dưới dạng đối tượng (`LinkedList<Customer>`).

Code mẫu:

```
@SuppressWarnings("unchecked")
public static LinkedList<Customer> readFromFile(String path) {
    LinkedList<Customer> customers = null;
    try {
        FileInputStream fis = new FileInputStream(path);
        ObjectInputStream ois = new ObjectInputStream(fis);
        customers = (LinkedList<Customer>) ois.readObject();
        ois.close();
        fis.close();
        System.out.println("Successfully read customer list from file.");
    } catch (Exception e) {
        e.printStackTrace();
    }
    return customers;
}
```

2. Lưu danh sách khách hàng ra file

Ý tưởng:

- Sử dụng `ObjectOutputStream` để ghi danh sách khách hàng vào file dưới dạng đối tượng.

Code mẫu:

```
public static boolean saveToFile(LinkedList<Customer> customers, String path)
{
    try {
        FileOutputStream fos = new FileOutputStream(path);
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        oos.writeObject(customers);
        oos.close();
        fos.close();
        System.out.println("Customer list successfully saved to file.");
        return true;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return false;
}
```

3. Hiển thị danh sách khách hàng

Ý tưởng:

- Duyệt qua danh sách và in thông tin của từng khách hàng.

Code mẫu:

```
public static void displayCustomerList(LinkedList<Customer> customers) {
    System.out.printf("%-3s %-8s %-20s %-15s %-10s\n", "#", "ID", "Name",
"Account Balance", "Class");
    int index = 1;
    for (Customer customer : customers) {
        System.out.printf("%-3d %-8s %-20s %-15.2f %-10s\n", index++,
customer.getCustomerId(),
        customer.getCustomerName(), customer.getCustomerAccount(),
customer.getCustomerClass());
    }
}
```

4. Thêm khách hàng vào danh sách

Ý tưởng:

- Nhập thông tin khách hàng từ người dùng và thêm vào danh sách.

Code mẫu:

```
public static void addCustomer(LinkedList<Customer> customers) {
    Scanner scan = new Scanner(System.in);
    System.out.println("Enter Customer ID: ");
    String id = scan.nextLine();
    System.out.println("Enter Customer Name: ");
    String name = scan.nextLine();
    System.out.println("Enter Account Balance: ");
    double account = scan.nextDouble();

    Customer customer = new Customer(id, name, account);
    customers.add(customer);
    System.out.println("Customer added successfully.");
}
```

5. Tìm kiếm khách hàng theo ID

Ý tưởng:

- Duyệt qua danh sách để tìm khách hàng có customerId khớp với giá trị nhập vào.

Code mẫu:

```
public static int findCustomerById(LinkedList<Customer> customers, String id)
{
    for (int i = 0; i < customers.size(); i++) {
        if (customers.get(i).getCustomerId().equals(id)) {
            return i; // Trả về vị trí của khách hàng trong danh sách
        }
    }
    return -1; // Không tìm thấy
}
```

Sử dụng:

```
System.out.println("Enter Customer ID to search: ");
String id = scan.nextLine();
int index = findCustomerById(customers, id);
if (index != -1) {
    System.out.println("Customer found: " + customers.get(index));
} else {
    System.out.println("Customer not found.");
}
```

6. Sắp xếp danh sách khách hàng theo tên

Ý tưởng:

- Sử dụng Comparator để sắp xếp danh sách dựa trên tên khách hàng.

Code mẫu:

```
public static void sortCustomersByName(LinkedList<Customer> customers) {
    customers.sort((c1, c2) ->
        c1.getCustomerName().compareToIgnoreCase(c2.getCustomerName()));
    System.out.println("Customer list sorted by name.");
}
```

7. Xoá khách hàng theo ID

Ý tưởng:

- Sử dụng hàm `findCustomerById` để tìm vị trí của khách hàng, sau đó xoá khỏi danh sách.

Code mẫu:

```
public static void deleteCustomerById(LinkedList<Customer> customers, String id) {
    int index = findCustomerById(customers, id);
    if (index != -1) {
        customers.remove(index);
        System.out.println("Customer deleted successfully.");
    } else {
        System.out.println("Customer not found.");
    }
}
```

8. Chương trình chính với menu

Code mẫu:

```
public static void main(String[] args) {
    LinkedList<Customer> customers = new LinkedList<>();
    Scanner scan = new Scanner(System.in);
    int choice;
    do {
        System.out.println("\nCUSTOMER MANAGEMENT PROGRAM");
        System.out.println("1. Load customer list from file");
        System.out.println("2. Display customer list");
        System.out.println("3. Add a customer");
        System.out.println("4. Find customer by ID");
        System.out.println("5. Delete customer by ID");
        System.out.println("6. Sort customer list by name");
        System.out.println("7. Save customer list to file");
        System.out.println("0. Exit");
        System.out.print("Enter your choice: ");
        choice = scan.nextInt();
        scan.nextLine(); // Consume newline
    }
```

```

switch (choice) {
    case 1:
        customers = readFromFile("CustomerList.dat");
        break;
    case 2:
        displayCustomerList(customers);
        break;
    case 3:
        addCustomer(customers);
        break;
    case 4:
        System.out.print("Enter customer ID to search: ");
        String searchId = scan.nextLine();
        int index = findCustomerById(customers, searchId);
        if (index != -1) {
            System.out.println("Customer found: " + customers.get(index));
        } else {
            System.out.println("Customer not found.");
        }
        break;
    case 5:
        System.out.print("Enter customer ID to delete: ");
        String deleteId = scan.nextLine();
        deleteCustomerById(customers, deleteId);
        break;
    case 6:
        sortCustomersByName(customers);
        break;
    case 7:
        saveToFile(customers, "CustomerList.dat");
        break;
    case 0:
        System.out.println("Exiting program...");
        break;
    default:
        System.out.println("Invalid choice. Please try again.");
}
} while (choice != 0);

scan.close();
}

```

-----*Hết*-----