

Đề bài: Quản lý tài khoản ngân hàng.

Ngân hàng ABC muốn lưu trữ thông tin của mỗi tài khoản như sau:

Mỗi tài khoản chứa các thông tin:

- Số tài khoản (Kiểu long)
- Tên tài khoản (kiểu chuỗi)
- Số tiền trong tài khoản (kiểu double)

Thiết kế lớp Account để lưu trữ các thông tin, lớp bao gồm các phương thức sau:

- Constructor: Có 2 constructor (mặc định và đầy đủ tham số)
- Các phương thức get, set cho từng thuộc tính
- Phương thức toString để trả về chuỗi chứa toàn bộ thông tin tài khoản, yêu cầu định dạng tiền tệ.

Thêm các thông tin sau vào lớp Account:

- Hằng số lãi suất có giá trị khởi tạo 0.035
- Constructor có 2 đối số: số tài khoản, tên tài khoản. Constructor này sẽ khởi tạo số tiền mặc định là 50
- Phương thức nạp tiền vào tài khoản: Lấy số tiền hiện tại trong tài khoản + số tiền nạp vào
- Phương thức rút tiền: Lấy số tiền hiện tại trong tài khoản – (số tiền muốn rút + phí rút tiền)
- Phương thức đáo hạn: Mỗi lần đến kỳ đáo hạn thì số tiền trong tài khoản = số tiền trong tài khoản + số tiền trong tài khoản * LAISUAT
- Phương thức chuyển khoản từ tài khoản này sang tài khoản khác

Chú ý: Mỗi thao tác phải kiểm tra số tiền nạp, rút, chuyển có hợp lệ hay không? (VD: tiền nạp vào < 0, tiền rút nhiều hơn tiền trong tài khoản thì thông báo không hợp lệ và yêu cầu nhập lại)

// Tài khoản

```
import java.text.NumberFormat;
import java.util.Scanner;
```

```

public class Account {
    //khai báo các thuộc tính
    private long soTK;
    private String tenTK;
    private double soTienTrongTK;

    Scanner sc = new Scanner(System.in);

    //khởi tạo constructor mặc định
    public Account() {
    }

    //khởi tạo constructor có tham số
    public Account(long soTK, String tenTK, double soTienTrongTK) {
        this.soTK = soTK;
        this.tenTK = tenTK;
        this.soTienTrongTK = soTienTrongTK;
    }

    //-----begin getter and setter-----
    public long getSoTK() {
        return this.soTK;
    }

    public void setSoTK(long soTK) {
        this.soTK = soTK;
    }

    public String getTenTK() {
        return this.tenTK;
    }

    public void setTenTK(String tenTK) {
        this.tenTK = tenTK;
    }

    public double getSoTienTrongTK() {
        return this.soTienTrongTK;
    }

    public void setSoTienTrongTK(double soTienTrongTK) {
        this.soTienTrongTK = soTienTrongTK;
    }

    //-----end getter and setter-----
    @Override
    public String toString() {
        // TODO Auto-generated method stub
        NumberFormat currencyEN = NumberFormat.getCurrencyInstance();
        String str1 = currencyEN.format(soTienTrongTK);
        return soTK + "-" + tenTK + "-" + str1;
    }

    //khởi tạo phương thức nạp tiền
    public double napTien() {
        double nap;
    }
}

```

```

        System.out.print("Nhập số tiền bạn cần nạp: ");
        nap = sc.nextDouble();
        //nếu số tiền nạp vào lớn hơn 0 thì hợp lệ
        if (nap >= 0) {
            soTienTrongTK = nap + soTienTrongTK;
            NumberFormat currencyEN = NumberFormat.getCurrencyInstance();
            String str1 = currencyEN.format(nap);
            System.out.println("bạn vừa nạp " + str1 + " vào tài khoản.");
        } else { //ngược lại nếu số tiền nộp vào bé hơn 0 thì không hợp lệ
            System.out.println("Số tiền nạp vào không hợp lệ!");
        }
        return nap;
    }

    //khởi tạo phương thức rút tiền
    public double rutTien() {
        double phi = 5;
        double rut;
        System.out.print("Nhập số tiền bạn cần rút: ");
        rut = sc.nextDouble();
        //nếu số tiền rút bé hơn hoặc bằng số tiền còn trong tài khoản + phí
        thì hợp lệ
        if (rut <= (soTienTrongTK - phi)) {
            soTienTrongTK = soTienTrongTK - (rut + phi);
            NumberFormat currencyEN = NumberFormat.getCurrencyInstance();
            String str1 = currencyEN.format(rut);
            System.out.println("Bạn vừa rút " + str1 + "Đ từ tài khoản. Phí là
$5.");
        } else { //ngược lại nếu số tiền rút lớn hơn số tiền có trong tài khoản
        thì không hợp lệ
            System.out.println("Số tiền muốn rút không hợp lệ!");
            return rutTien();
        }
        return rut;
    }

    //khởi tạo phương thức đáo hạn
    public double daoHan() {
        double laiSuat = 0.035;
        soTienTrongTK = soTienTrongTK + soTienTrongTK * laiSuat;
        NumberFormat currencyEN = NumberFormat.getCurrencyInstance();
        String str1 = currencyEN.format(soTienTrongTK);
        System.out.println("Bạn vừa được " + str1 + " từ đáo hạn 1 tháng");
        return soTienTrongTK;
    }

    //khởi tạo phương thức in kết quả ra màn hình
    void inTK() {
        NumberFormat currencyEN = NumberFormat.getCurrencyInstance();
        String str1 = currencyEN.format(soTienTrongTK);
        System.out.printf("%-10d %-20s %-20s \n", soTK, tenTK, str1);
    }
}

```

// Tài khoản

```

import java.util.Scanner;
public class Main {
    static Scanner sc = new Scanner(System.in);
    static void nhapTK(Account tk) {
        System.out.println("Nhập số tài khoản: ");
        tk.setSoTK(sc.nextInt());
        sc.nextLine();
        System.out.println("Nhập tên tài khoản: ");
        tk.setTenTK(sc.nextLine());
        tk.setSoTienTrongTK(50);
    }
    public static void main(String[] args) {
        Account a[] = null;
        int k, b, n = 0;
        long s, d, c, f;
        boolean flag = true;
        do {
            System.out.println("Bạn chọn làm gì: ");
            System.out.println("1.Nhập thông tin của các khách hàng\n"
                + "2.Xuất danh sách thông tin của các khách hàng\n" +
                "3.Nạp tiền\n" + "4.Rút tiền\n"
                + "5.Đáo hạn\n" + "6.Chuyển khoản\n" + "Số khác để
            thoát");
            b = sc.nextInt();
            switch (b) {
                case 1:
                    System.out.println("Nhập số lượng khách hàng bạn muốn
            nhập: ");
                    n = sc.nextInt();
                    a = new Account[n];
                    for (int i = 0; i < n; i++) {
                        System.out.println("Khách hàng số: " + (i+1));
                        a[i] = new Account();
                        nhapTK(a[i]);
                    }
                    break;
                case 2:
                    System.out.printf("%-10s %-20s %-20s\n", "Số TK", "Tên
            TK", "Số tiền trong TK");
                    for (int i = 0; i < n; i++) {
                        a[i].inTK();
                    }
                    break;
                case 3:
                    System.out.println("Nhập số tài khoản khách hàng cần nạp
            tiền: ");
                    s = sc.nextLong();
                    for (int i = 0; i < n; i++) {
                        d = a[i].getSoTK();
                        if (s == d) {
                            System.out.println("Bạn chọn tài khoản: " + d);
                            a[i].napTien();
                        } else {
                            System.out.println("");
                        }
                    }
            }
        } while (flag);
    }
}

```

```

        }
        break;
    case 4:
        System.out.println("Nhập số tài khoản khách hàng cần rút
tiền: ");

        s = sc.nextLong();
        for (int i = 0; i < n; i++) {
            d = a[i].getSoTK();
            if (s == d) {
                System.out.println("Bạn chọn tài khoản: " + d);
                a[i].rutTien();
            }
        }
        break;
    case 5:
        System.out.println("Nhập số tài khoản khách hàng cần đảo
hạn: ");

        s = sc.nextLong();
        for (int i = 0; i < n; i++) {
            d = a[i].getSoTK();
            if (s == d) {
                System.out.println("Bạn chọn tài khoản: " + d);
                a[i].daoHan();
            } else {
                System.out.println("");
            }
        }
        break;
    case 6:
        double chuyen,
            nhan,
            tienChuyen;
        System.out.print("Nhập số tài khoản khách hàng chuyển
tiền: ");

        s = sc.nextLong();
        System.out.print("Nhập số tài khoản khách hàng nhận tiền:
");

        c = sc.nextLong();
        for (int i = 0; i < n; i++) {
            d = a[i].getSoTK();
            if (s == d) {
                chuyen = a[i].getSoTienTrongTK();
                for (int j = 0; j < n; j++) {
                    f = a[j].getSoTK();
                    if (c == f) {
                        nhan = a[j].getSoTienTrongTK();
                        System.out.println("Nhập số tiền cần
chuyển");

                        tienChuyen = sc.nextDouble();
                        if (tienChuyen <= chuyen) {
                            chuyen = chuyen - tienChuyen;
                            nhan = nhan + tienChuyen;
                            a[i].setSoTienTrongTK(chuyen);
                            a[j].setSoTienTrongTK(nhan);
                            System.out.println("Tài khoản số " + d
+ " vừa chuyển: $" + tienChuyen);

```

```

        System.out.println("Tài khoản số " + f
+ " vừa nhận: $" + tienChuyen);
    } else {
        System.out.println("Số tiền nhập
không hợp lệ!");
    }
} else {
    System.out.println("");
}
}
} else {
    System.out.println("");
}
}
break;
default:
    System.out.println("Bye!!");
    flag = false;
    break;
}
} while (flag);
}
}

```