**Trang bìa**

**Nội dung**

**Lớp và đối tượng**

**Yêu cầu:** Tạo một project có tên theo cú pháp LopHP\_LT\_Tuan04\_STT\_MSSV\_Ten. Làm các bài tập theo yêu cầu sau:

*Toàn bộ bài làm phải chú thích rõ ràng và phải tuân theo Naming Conventions http://java.sun.com/docs/codeconv/html/CodeConventions.doc8.html*

**Bài toán** Quản lý các Tài khoản ngân hàng *(BankAccount)* trong một ngân hàng *(Bank)*

**1.1***.* Một tài khoản ngân hàng có các thuộc tính sau: số tài khoản *(accountNumber)*, số dư *(balance)*, tên chủ tài khoản *(ownerName)*.

Viết 2 constructors cho lớp BankAccount. Constructor không có tham số và constructor với đầy đủ tham số.

public BankAccount()

Khởi tạo số tài khoản là "0000-0000-0000", tên chủ tài khoản là "No name", số dư là 0.0

public BankAccount(String accountNumber, String ownerName, double balance)

Khởi tạo đầy đủ các thuộc tính của lớp BankAccount

Viết các phương thức để thực hiện các chức năng sau:

1/ Gửi tiền

public void deposit(double amount) //Amount phải lớn hơn 0

2/ Rút tiền

public void withdraw(double amount) //Amount phải lớn hơn 0 và nhỏ hơn hoặc bằng số dư hiện có

3/ Lấy số dư hiện có

public double getBalance()

4/ Chuyển tiền

public void transfer(BankAccount other, double amount)

5/ Setters/Getters cho accountNumbervà ownerName

6/ Trả về thông tin tài khoản

public String toString()

**1.2** Một ngân hàng (Bank), có các thuộc tính sau: tên ngân hàng *(name)*, danh sách các tài khoản ngân hàng *(accounts)*.

Viết 2 constructors cho lớp Bank. Constructor không có tham số, khởi tạo số tài khoản là 100 và constructor với hai tham số là tên ngân hàng và số tài khoản cần khởi tạo.

public Bank()

public Bank(String name, int numberOfAccounts)

Viết các phương thức để thực hiện các chức năng sau:

1/ Mở một tài khoản mới và thêm vào danh sách các tài khoản ngân hàng

public void addNew(String accountNumber, String accountName, double balance)

2/ Tìm một tài khoản ngân hàng theo số tài khoản.

public BankAccount find(String accountNumber)

3/ Tính tổng số tiền trong tất cả các tài khoản ngân hàng

public double getTotalBalance()

4/ Tính số tài khoản có trong ngân hàng

public int getNumberOfAccounts()

5/ Lấy tất cả các tài khoản ngân hàng

public BankAccount[] getAccounts()

6/ Setters và getters cho thuộc tính tên ngân hàng

Ràng buộc toàn vẹn dữ liệu:

|  |  |  |
| --- | --- | --- |
| **Valid** | **Exception** | **Message** |
| Số tài khoản trong ngân hàng > 0 | IllegalArgumentException | Number of accounts must be greater than 0 |
| Mã số tài khoản trong ngân hàng không được trùng | IllegalArgumentException | Account number already exists |
| Số tài khoản mở mới không vượt quá kích thước số tài khoản trong ngân hàng cho phép | IllegalArgumentException | Bank is full |
| Tên ngân hàng không rỗng hoặc trống | IllegalArgumentException | The bank name must not be null |
| Số tài khoản không rỗng hoặc trống | IllegalArgumentException | The account number must not be null |
| Tên chủ tài khoản không rỗng hoặc trống | IllegalArgumentException | The owner name must not be null |
| Số dư tài khoản >=0 | IllegalArgumentException | Balance must be greater than or equal to 0 |
| Số tiền nộp vào phải > 0 | IllegalArgumentException | Amount must be greater than 0 |
| Số tiền rút ra phải lớn hơn 0 và nhỏ hơn hoặc bằng số dư | IllegalArgumentException | Amount must be greater than 0 and less than or equal to balance |

**1.3** Dùng một phần mềm thiết kế UML, biểu diễn mô hình lớp cho mục 1.1, 1.2

A screenshot of a computer

Description automatically generated

**1.4** Hiện thực 1.1 và 1.2

A screenshot of a computer

Description automatically generated

A computer screen with many colorful text

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer screen with many colorful lines

Description automatically generated

A computer screen with many colorful text

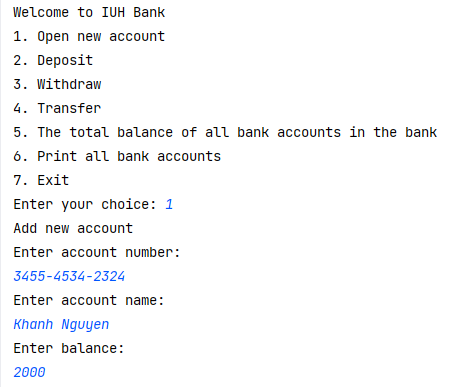
Description automatically generated

**1.5** Viết lớp và tạo các Unit test để kiểm nghiệm các phương thức 1.1 và 1.2

**1.6** Lập bảng báo cáo kiểm thử *(file excel)*

**1.7** Viết lớp BankTester chứa phương thức main, tạo menu các lựa chọn, cho phép người dùng mở tài khoản, nạp tiền, rút tiền, chuyển khoản … và in các thông tin khác nếu muốn, dữ liệu được nhập từ bàn phím.

Menu có dạng như sau:



A computer screen with many colorful text

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated