Exceptions

1. Create a class MyNumber having the following features:

Attributes

int first number

int second number

result double stores the result of arithmetic operations performed on a

and b

Member functions

MyNumber(x, y) constructor to initialize the values of a and b

add() stores the sum of a and b in result

sub() stores difference of a and b in result

mul() stores product in result

div() stores a divided by b in result

a) Test to see if b is zero (0) and throw an appropriate exception since division by zero is undefined.
b) Display a menu to the user to perform the above four arithmetic operations.

2. Create a class BankAccount having the members as given below:

accNo integer

custName string

accType string (indicates 'Savings' or 'Current')

balance float

a) Include the following methods in the BankAccount class:

void deposit(float amt);

void withdraw(float amt);

float getBalance();

- deposit(float amt) method allows you to credit an amount into the current balance. If amount is negative, throw an exception NegativeAmount to block the operation from being performed.

- withdraw(float amt) method allows you to debit an amount from the current balance. Please ensure a minimum balance of Rs. 1000/- in the account for savings account and Rs. 5000/-for current account, else throw an exception InsufficientFunds and block the withdrawal operation.

Also throw an exception NegativeAmount to block the operation from being performed if the "amt" parameter passed to this function is negative.

- getBalance() method returns the current balance. If the current balance is below the minimum required balance, then throw an exception LowBalanceException accordingly.

b) Have constructor to which you will pass, accno, cust_name, acctype and initial balance. Check whether the balance is less than 1000 or not in case of savings account and less than 5000 in case of a current account. If so, then raise a LowBalanceException.

In either case if the balance is negative then raise the NegativeAmount exception accordingly.

3. Create class Employee

int empNumber;

String name;

float exp;

Employee(int empNumber,String name,float exp);

While creating employee object , if exp is >2.5 years and exp < 3.6 years, create an object otherwise throw IllegalArgumentException .

4. Create a class with following specifications.

Class Emp

empId int

empName string

designation string

basic double

hra double readOnly

Methods

printDET()

calculateHRA()

printDET() methods will show details of the EMP.

calculateHRA() method will calculate HRA based on basic.

There will be 3 designations supported by the application.

If designation is "Manager" - HRA will be 10% of BASIC

if designation is “Officer” - HRA will be 12% of BASIC

if category is “CLERK” - HRA will be 5% of BASIC

Have constructor to which you will pass empId, designation, basic and hra.

And check whether the BASIC is less than 500 or not. If it is less than 500 raise a custom exception as given below:

Create LowSalException class with proper user message to handle BASIC less than

500.

5. Create class DbConnection

public static DbConnection getObject()

private DbConnection()

which allows to create only single object, if there is an attempt to create another object , it should throw an exception.