Expno: Exceptions 23IT111

AIM:

To implement user exceptions such as NegativeValueException, InvalidIdException, NumberFormatException for Employee Management system.

Procedure:

* Create NegativeValueException, InvalidIdException, NumberFormatException which is extended by Excpetion class.
* Create a Employee class to input the details of a employee such as name, ID, phnum, designation, salary.
* Get the details from parameterized constructors.
* Define three methods checksal, checknum, checkid to check the salary, phnum, id respectively for validation.
* checksal, checknum, checkid is extended to Exception classes NegativeValueException, InvalidIdException, NumberFormatException respectively.
* NegativeValueException throws exception if the salary is negative, InvalidIdException throws exception if the Id does not startsWith ‘IT’ and NumberFormatException throws exception if the phnum is not ten digits.
* Create an Employee object in main class with suitable arguments.
* Use three try and catch blocks to validate the condition and to catch if the exception occurs.
* End procedure.

CODE:

import java.util.\*;

class NegativeValueException extends Exception{

String msg; NegativeValueException(String msg){

super(msg); this.msg=msg;}

public String toString(){

return msg; }}

class InvalidIdException extends Exception{

String msg; InvalidIdException(String msg){

super(msg); this.msg=msg;}

public String toString(){

return msg; }}

class NumberFormatException extends Exception{

String msg; NumberFormatException(String msg){

super(msg); this.msg=msg; }

public String toString(){ return msg;}}

class Employee{

String name,desg,id,phno;float sal;

Employee(String a,String b,String c,String d,float sal){

name=a;desg=b;id=c;phno=d;this.sal=sal;}

public void checksal() throws NegativeValueException{

if(sal<0){

throw new NegativeValueException("Invalid Input: Salary can't be negative");

}else{

System.out.println("Salary accepted");}}

public void checkId() throws InvalidIdException{

if(!id.startsWith("IT")){

throw new InvalidIdException("Invalid Input: Wrong Id"); }

else{ System.out.println("Id accepted");}}

public void checknum() throws NumberFormatException{

if(phno.length()!=10){

throw new NumberFormatException("Invalid Input: Wrong phone number");}

else{ System.out.println("NUmber accepted"); }}}

class oops6a{

public static void main(String args[]){

Employee ob=new Employee("Praveen","Intern","111","12345",-10000);

try{ ob.checknum();}

catch(NumberFormatException e){

System.out.println(e);}

try{ ob.checksal();}

catch(NegativeValueException e){

System.out.println(e);}

try{ ob.checkId();}

catch(InvalidIdException e){

System.out.println(e);}}}

RESULT:

The above program is successfully implemented, executed and verified.

CLASS DIAGRAM:

|  |
| --- |
| CLASS: NumberFormatException |
| VARIABLES: String msg |
| METHODS: +String toString() |

|  |
| --- |
| CLASS: InvalidIdException |
| VARIABLES: String msg |
| METHODS: +String toString() |

|  |
| --- |
| CLASS: NegativeValueException |
| VARIABLES: String msg |
| METHODS: +String toString() |

|  |
| --- |
| CLASS: Employee |
| VARIABLES: String, String, String, String, float |
| METHODS:  Void checksal(), void checknum(), void checkid(). |

Sample Input/Output:

Employee ob=new Employee("Praveen","Intern","111","12345",-10000);

Invalid Input: Wrong phone number.

Invalid Input: Salary can't be negative.

Invalid Input: Wrong Id.