/\*Aviansh Gupta

185001028

CSE A

Ex5

Q1. Create a class named "Person" which consists of name, age, aadharnumber. Create

methods getInput(), display(), canVote(), hasAadhar(). Create and handle the following

Exceptions.

a. For age -> if you give alphabets then throw NumberFormatException

b. For voting -> if age is less than 18 then throw MinorCitizenException

c. For aadhar -> if no valid aadhar then throw NullPointerException \*/

import java.util.\*;

import java.lang.Character;

class MinorCitizenException extends Exception{

String message;

MinorCitizenException(String message){

super(message);

this.message = message;

}

}

class Person{

protected String name;

protected int age;

private int aadharnumber;

public void getInput(){

Scanner s = new Scanner(System.in);

System.out.print("Enter the name: ");

name = s.nextLine();

do{

try{

System.out.print("Enter the age: ");

String age1 = s.nextLine();

for(int i=0;i<age1.length();i++){

if(Character.isLetter(age1.charAt(i)))

throw new NumberFormatException("For input age: "+age1);

}

age = Integer.parseInt(age1);

}

catch(NumberFormatException n){

System.out.println("Number Format Exception: "+n.getMessage());

}

}while(age <= 0);

System.out.print("Enter the aadhar number: ");

aadharnumber = s.nextInt();

}

public void display(){

System.out.println("-----------------------------------------------");

System.out.printf("Name: %s\nAge: %d\nAadhar No: %d\n",name,age,aadharnumber);

System.out.println("-----------------------------------------------");

}

public void canVote(){

try{

if(age < 18)

throw new MinorCitizenException("Minor Citizen Exception: You are a minor.");

else

System.out.println("You can Vote.");

}

catch(MinorCitizenException e){

System.out.println(e.message);

}

}

public void hasAadhar(){

try{

if(aadharnumber < 0)

throw new NullPointerException("Invalid Aadhar Number");

else

System.out.println("The person has a valid aadhar number.");

}

catch(NullPointerException a){

System.out.println("Null Pointer Exception: "+a.getMessage());

}

}

}

class Ex5\_Q1{

public static void main(String []args){

Person p = new Person();

p.getInput();

p.display();

p.canVote();

p.hasAadhar();

}

}

/\*OUTPUT

PS C:\Users\AVIANSH GUPTA\Desktop\programs\java\Ex5> java Ex5\_Q1

Enter the name: Danish

Enter the age: 1e

Number Format Exception: For input age: 1e

Enter the age: 17

Enter the aadhar number: 12345

-----------------------------------------------

Name: Danish

Age: 17

Aadhar No: 12345

-----------------------------------------------

Minor Citizen Exception: You are a minor.

The person has a valid aadhar number.

PS C:\Users\AVIANSH GUPTA\Desktop\programs\java\Ex5> java Ex5\_Q1

Enter the name: Danish

Enter the age: 19

Enter the aadhar number: -98

-----------------------------------------------

Name: Danish

Age: 19

Aadhar No: -98

-----------------------------------------------

You can Vote.

Null Pointer Exception: Invalid Aadhar Number \*/

--------------------------------------------------------------------------------------------------------------------------------

/\*Ex5 Ouestion2 \*/

import java.util.\*;

class PANRequiredException extends Exception{

public String message;

PANRequiredException(String message){

super(message);

this.message = message;

}

}

class PANFormatMismatchException extends Exception{

public String message;

PANFormatMismatchException(String message){

super(message);

this.message = message;

}

}

class BranchNotFoundException extends Exception{

public String message;

BranchNotFoundException(String message){

super(message);

this.message = message;

}

}

class AccountNotFoundException extends Exception{

public String message;

AccountNotFoundException(String message){

super(message);

this.message = message;

}

}

class MinBalRequiredException extends Exception{

public String message;

MinBalRequiredException(String message){

super(message);

this.message = message;

}

}

class NotEnoughMoneyInAccountException extends Exception{

public String message;

NotEnoughMoneyInAccountException(String message){

super(message);

this.message = message;

}

}

class Account{

protected String name;

private int acct\_num;

private String branch;

private float balance;

private String PAN\_num;

public Account(String name,int actnum,float balance){

this.name = name;

acct\_num = actnum;

this.balance = balance;

}

public float getBalance(){

return balance;

}

public int getAccnum(){

return acct\_num;

}

public String getName(){

return name;

}

public String getBranch(){

if(branch == null)

return "NA";

return branch;

}

public int setPAN(String PAN){

int flag=1;

for(int i=0;i<PAN.length();i++){

if(i<5 && !Character.isLetter(PAN.charAt(i))){

flag = 0;

break;

}

else if(i>4 && i<9 && !Character.isDigit(PAN.charAt(i))){

flag = 0;

break;

}

else if(i==9 && !Character.isLetter(PAN.charAt(i))){

flag = 0;

break;

}

}

try{

if(flag == 0)

throw new PANFormatMismatchException("Invalid PAN Number Entered "+PAN);

else

this.PAN\_num = PAN;

}

catch(PANFormatMismatchException pe){

System.out.println("PANFormatMismatchException: "+pe.message);

}

return flag;

}

public String getPAN(){

if(PAN\_num == null)

return "NA";

return PAN\_num;

}

public int accSearch(int accnum){

int flag = 0;

try{

if(accnum != acct\_num)

throw new AccountNotFoundException("Account Number "+accnum+" Not Found" );

else

flag = 1;

}

catch(AccountNotFoundException ae){

System.out.println("AccountNoyFoundException: "+ae.message);

}

return flag;

}

public int setBranch(String branch){

int flag = 0;

String []branchArr = {"Branch1","Branch2","Branch3","Branch4","Branch5"};

for(int i=0;i<branchArr.length;i++){

if(branch.equalsIgnoreCase(branchArr[i])){

flag = 1;

break;

}

}

try{

if(flag == 0)

throw new BranchNotFoundException("Branch "+branch+" not found");

else

this.branch = branch;

}

catch(BranchNotFoundException be){

System.out.println("BranchNotFoundException: "+be.message);

}

return flag;

}

public void deposit(float amt){

Scanner s = new Scanner(System.in);

try{

if(amt<=25000){

balance += amt;

System.out.println("\nAmount Deposited Successfully\n");

}

else{

throw new PANRequiredException("PAN number required for amount more than 25000");

}

}

catch(PANRequiredException pr){

System.out.println("PANRequiredException: "+pr.message);

}

if(amt>25000){

int boolval;

do{

System.out.print("Enter PAN number: ");

boolval = setPAN(s.nextLine());

}while(boolval != 1);

balance += amt;

System.out.println("\nAmount Deposited Successfully\n");

}

}

public int withdrawal(float amt){

int minBal = 500,flag = 1;

try{

if(amt < minBal){

flag = 0;

throw new MinBalRequiredException("Amount Entered Less Than Minimum Withdrawal Limit("+minBal+")");

}

else if(amt > getBalance()){

flag = 0;

throw new NotEnoughMoneyInAccountException("Amount Entered More Than Balance "+getBalance());

}

else{

balance -= amt;

System.out.println("\nAmount Withdrawn Successfully\n");

}

}

catch(MinBalRequiredException me){

System.out.println("MinBalRequiredException: "+me.message);

}

catch(NotEnoughMoneyInAccountException ne){

System.out.println("NotEnoughMoneyInAccountException: "+ne.message);

}

return flag;

}

}

class TestExceptions{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

String name,branch,pan;

int accnum,boolval1,boolval2,boolval3,boolval4;

float bal;

System.out.print("Enter the name of the account holder: ");

name = sc.nextLine();

System.out.print("Enter the accnum: ");

accnum = sc.nextInt();

System.out.print("Enter balance: ");

bal = sc.nextFloat();

Account a = new Account(name, accnum, bal);

int choice;

System.out.println("CHOICES\n1: Deposit\n2: Withdraw\n3: Enter PAN\n4: Enter Branch\n5: Search Account\n6: Display\n");

System.out.print("Enter your choice: ");

choice = sc.nextInt();

do{

switch(choice){

case 1: System.out.print("Enter amount to deposit: ");

int damt = sc.nextInt();

a.deposit(damt);

break;

case 2: do{

System.out.print("Enter amount to withdraw: ");

int wamt = sc.nextInt();

boolval4 = a.withdrawal(wamt);

}while(boolval4 != 1);

break;

case 3: sc.nextLine();

do{

System.out.print("Enter PAN Number: ");

pan = sc.nextLine();

boolval1 = a.setPAN(pan);

}while(boolval1 != 1);

break;

case 4: sc.nextLine();

do{

System.out.print("Enter the branch: ");

branch = sc.nextLine();

boolval2 = a.setBranch(branch);

}while(boolval2 != 1);

break;

case 5: do{

System.out.print("Enter the account number to be searched: ");

int acn = sc.nextInt();

boolval3 = a.accSearch(acn);

}while(boolval3 != 1);

System.out.println("\nAccount Found\n");

display(a);

break;

case 6: display(a);

break;

}

System.out.println("CHOICES\n1: Deposit\n2: Withdraw\n3: Enter PAN\n4: Enter Branch\n5: Search Account\n6: Display\n");

System.out.print("Enter your choice(To stop enter -1): ");

choice = sc.nextInt();

}while(choice != -1);

}

static public void display(Account obj){

System.out.println("-----------------------------------------------");

System.out.printf("Name: %s\nAccount Number: %d\nPAN Number: %s\nBranch: %s\nBalance: %.2f\n",obj.getName(),obj.getAccnum(),obj.getPAN(),obj.getBranch(),obj.getBalance());

System.out.println("-----------------------------------------------");

}

}

/\*OUTPUT

Enter the name of the account holder: Danish

Enter the accnum: 123456

Enter balance: 2000

CHOICES

1: Deposit

2: Withdraw

3: Enter PAN

4: Enter Branch

5: Search Account

6: Display

Enter your choice: 1

Enter amount to deposit: 28000

PANRequiredException: PAN number required for amount more than 25000

Enter PAN number: ABCFE1237H

Amount Deposited Successfully

CHOICES

1: Deposit

2: Withdraw

3: Enter PAN

4: Enter Branch

5: Search Account

6: Display

Enter your choice(To stop enter -1): 6

-----------------------------------------------

Name: Danish

Account Number: 123456

PAN Number: ABCFE1237H

Branch: NA

Balance: 30000.00

-----------------------------------------------

CHOICES

1: Deposit

2: Withdraw

3: Enter PAN

4: Enter Branch

5: Search Account

6: Display

Enter your choice(To stop enter -1): 4

Enter the branch: Branch9

BranchNotFoundException: Branch Branch9 not found

Enter the branch: branch5

CHOICES

1: Deposit

2: Withdraw

3: Enter PAN

4: Enter Branch

5: Search Account

6: Display

Enter your choice(To stop enter -1): 6

-----------------------------------------------

Name: Danish

Account Number: 123456

PAN Number: ABCFE1237H

Branch: branch5

Balance: 30000.00

-----------------------------------------------

CHOICES

1: Deposit

2: Withdraw

3: Enter PAN

4: Enter Branch

5: Search Account

6: Display

Enter your choice(To stop enter -1): 2

Enter amount to withdraw: 288

MinBalRequiredException: Amount Entered Less Than Minimum Withdrawal Limit(500)

Enter amount to withdraw: 35000

NotEnoughMoneyInAccountException: Amount Entered More Than Balance 30000.0

Enter amount to withdraw: 10000

Amount Withdrawn Successfully

CHOICES

1: Deposit

2: Withdraw

3: Enter PAN

4: Enter Branch

5: Search Account

6: Display

Enter your choice(To stop enter -1): 6

-----------------------------------------------

Name: Danish

Account Number: 123456

PAN Number: ABCFE1237H

Branch: branch5

Balance: 20000.00

-----------------------------------------------

CHOICES

1: Deposit

2: Withdraw

3: Enter PAN

4: Enter Branch

5: Search Account

6: Display

Enter your choice(To stop enter -1): 5

Enter the account number to be searched: 12345

AccountNoyFoundException: Account Number 12345 Not Found

Enter the account number to be searched: 123456

Account Found

-----------------------------------------------

Name: Danish

Account Number: 123456

PAN Number: ABCFE1237H

Branch: branch5

Balance: 20000.00

-----------------------------------------------

CHOICES

1: Deposit

2: Withdraw

3: Enter PAN

4: Enter Branch

5: Search Account

6: Display

Enter your choice: 3

Enter PAN Number: ABCD124AC

PANFormatMismatchException: Invalid PAN Number Entered ABCD124AC

Enter PAN Number: 12abcg564

PANFormatMismatchException: Invalid PAN Number Entered 12abcg564

Enter PAN Number: ABCFE1237H

CHOICES

1: Deposit

2: Withdraw

3: Enter PAN

4: Enter Branch

5: Search Account

6: Display

Enter your choice: 6

-----------------------------------------------

Name: Danish

Account Number: 123456

PAN Number: ABCFE1237H

Branch: NA

Balance: 20000.00

-----------------------------------------------

CHOICES

1: Deposit

2: Withdraw

3: Enter PAN

4: Enter Branch

5: Search Account

6: Display

Enter your choice(To stop enter -1): -1 \*/

--------------------------------------------------------------------------------------------------------------------------------