```
//S Harshini-185001058
//1
import java.util.Scanner;
class MinorCitizenException extends Throwable
{
	int age;
	MinorCitizenException(int n)
	{
		age=n;
	}
	public String toString()
	{
		return age+" ";
	}
}

class NullPointerException extends Throwable
{
	String details;
	NullPointerException(String aadhar)
	{
		details=aadhar;
	}
	public String toString()
	{
		return details;
	}
}

class NumberFormatException extends Throwable
{
	String details;
	NumberFormatException(String age)
	{
		details=age;
	}
	public String toString()
	{
		return details;
	}
}

class Person
```

```
{
	private String name,aadhar;
	private int age;

	void getInput() throws
MinorCitizenException,NullPointerException,NumberFormatException
	{
		Scanner in=new Scanner(System.in);
		System.out.print("name:");
		name=in.nextLine();
		System.out.print("aadhar:");
		aadhar=in.nextLine();
		System.out.print("age:");
		String r_age=in.nextLine();
		String []st=new String[10];
		st=r_age.split("");
		for(int i=0;i<st.length;i++)
			if(Character.isLetter(st[i].charAt(0)))
				throw new NumberFormatException(r_age);
		age=Integer.parseInt(r_age);
		if(aadhar.length()<12)
			throw new NullPointerException(aadhar);
		if(age<18)
			throw new MinorCitizenException(age);

	}

	void display()
	{
		System.out.println("Details");
		System.out.println("Name:"+name+"\nAge:"+age+"\nAadhar"+aadhar);
	}

	boolean canVote()
	{
		if(age>=18)
			return true;
		return false;
	}

	boolean hasAadhar()
	{
		if(aadhar.equals("\n"))
```

```
                        return false;
                return true;
        }
}

class Main
{
        public static void main(String args[])
        {
                Person p=new Person();
                try
                {
                        p.getInput();
                }
                catch(MinorCitizenException e){
                        System.out.println("MinorCitizenException " +e);
                }
                catch(NullPointerException e){
                        System.out.println("NullPointerException " +e);
                }
                catch(NumberFormatException e){
                        System.out.println("NumberFormatException " +e);
                }
                p.display();
        }
}

/*SAMPLE INPUT/OUTPUT

C:\Users\Harshini\Desktop>javac Main.java

C:\Users\Harshini\Desktop>java Main
name:harshu
aadhar:123456789101
age:23
Details
Name:harshu
Age:23
Aadhar123456789101

C:\Users\Harshini\Desktop>java Main
name:rani
aadhar:1234
```

```
age:12
NullPointerException 1234
Details
Name:rani
Age:12
Aadhar1234

C:\Users\Harshini\Desktop>java Main
name:tanu
aadhar:112233445566
age:11
MinorCitizenException 11
Details
Name:tanu
Age:11
Aadhar112233445566

C:\Users\Harshini\Desktop>java Main
name:renu
aadhar:112233445566
age:sd
NumberFormatException sd
Details
Name:renu
Age:0
Aadhar112233445566
*/


//2
import java.util.Scanner;

class PanRequiredException extends Exception
	{
	public String toString()
		{
		return "PanRequiredException:Required";
		}
	}

class MinBalRequiredException extends Exception
	{
	public String toString()
```

```
		{
		return "MinBalRequiredException:below Minimum Limit";
		}
	}

class NotEnoughMoneyInAccountException extends Exception
	{
	public String toString()
		{
		return "NotEnoughMoneyInAccountException: Insufficient";
		}
	}

class AccountNotFoundException extends Exception
	{
	public String toString()
		{
		return "AccountNotFoundException: Requested Account";
		}
	}

class Account
	{
	private String cname,branch;
	private int pan,accno;
	private double balance;

	public Account(String cname,int pan,int accno,String branch,double balance)
		{
		this.cname=cname;
		this.pan=pan;
		this.accno=accno;
		this.branch=branch;
		this.balance=balance;
		}

	public String getCName()
		{return cname;}
	public int getPAN()
		{return pan;}
	public int getAccNo()
		{return accno;}
	public String getBranch()
```

```
		{return branch;}
	public double getBalance()
		{return balance;}

	public void deposit(int accno,double amt) throws PanRequiredException
		{
		if(amt>50000)
			throw new PanRequiredException();
		else
			balance+=amt;
		}

	public void withdraw(int accno,double amt) throws
MinBalRequiredException,NotEnoughMoneyInAccountException
		{
		if(balance<amt)
			throw new NotEnoughMoneyInAccountException();
		else if((balance-amt)<1000.0)
			throw new MinBalRequiredException();
		else
			balance-=amt;
		}

	public void search(int accno)throws AccountNotFoundException
		{
		if(this.accno==accno)
			{
			System.out.println("\nName: "+cname);
			System.out.println("PAN: "+pan);
			System.out.println("Account Number: "+accno);
			System.out.println("Branch: "+branch);
			System.out.println("Current Balance: "+balance);
			}
		else
			throw new AccountNotFoundException();
		}

	public String toString()
		{
		return "Account Number "+accno+" has produced ";
		}
}
```

```
public class Main{
	public static void main(String args[]){

		String name,branch;
		int pin,accno;
		double bal;
		int num;
		double amt;
		int ano;
		int opt;

		Scanner in=new Scanner(System.in);

		System.out.print("\nEnter no of accounts");num=in.nextInt();

		Account acc[]=new Account[100];
		for(int i=0;i<num;i++){
			System.out.println("account Details ");
			in.nextLine();
			System.out.print("Name: ");name=in.nextLine();
			System.out.print("PIN number: ");pin=in.nextInt();
			System.out.print("Account number: ");accno=in.nextInt();
			in.nextLine();
			System.out.print("Branch: ");branch=in.nextLine();
			System.out.print("Balance: ");bal=in.nextDouble();

			acc[i]=new Account(name,pin,accno,branch,bal);
		}

		do{


			System.out.println("\nChoice:  \n 1-Deposit\n 2-Withdraw");
			System.out.println(" 0-Exit");
			System.out.print(" Your choice: ");
			opt=in.nextInt();

			if(opt==1){
				System.out.print("Enter amount to deposit: ");
				amt=in.nextDouble();
				System.out.print("Enter account number: ");
				ano=in.nextInt();
```

```
                        for(int i=0;i<num;i++){
                                try{
                                        acc[i].search(ano);
                                        try{
                                                acc[i].deposit(ano,amt);
                                                System.out.println("\nAfter deposit ");
                                                acc[i].search(ano);
                                                break;
                                        }
                                        catch(PanRequiredException pre){

System.out.println(acc[i].toString()+pre.toString());break;
                                        }
                                }
                                catch(AccountNotFoundException anfe){
                                        System.out.println(acc[i].toString()+anfe.toString());
                                }
                        }
                }
                else if(opt==2){
                        System.out.print("Enter amount to withdraw: ");
                        amt=in.nextDouble();
                        System.out.print("Enter account number: ");
                        ano=in.nextInt();

                        for(int i=0;i<num;i++){
                                try{
                                        acc[i].search(ano);
                                        try{
                                                acc[i].withdraw(ano,amt);
                                                System.out.println("After deposit ");
                                                acc[i].search(ano);
                                                break;
                                        }
                                        catch(MinBalRequiredException pre){

System.out.println(acc[i].toString()+pre.toString());break;
                                        }
                                        catch(NotEnoughMoneyInAccountException
nemiae){

System.out.println(acc[i].toString()+nemiae.toString());break;
                                        }
```

```
                                        }
                                        catch(AccountNotFoundException anfe){
                                                System.out.println(acc[i].toString()+anfe.toString());
                                        }
                                }
                        }
                        else if(opt!=0)
                                System.out.println("Invalid option ");
                }while(opt!=0);
        }
}


/*SAMPLE INPUT/OUTPUT
C:\Users\Harshini\Desktop>java Main

Enter no of accounts1
account Details
Enter Name: harshu
Enter PIN number: 1244
Enter Account number: 65664
Enter Branch: adambakkam
Enter Balance: 50000

Menu:
 1-Deposit
 2-Withdraw
 0-Exit
 Your choice: 1
Enter amount to deposit: 30000
Enter account number: 65664

Name: harshu
PAN: 1244
Account Number: 65664
Branch: adambakkam
Current Balance: 50000.0

After deposit

Name: harshu
PAN: 1244
Account Number: 65664
```

Branch: adambakkam
Current Balance: 80000.0

Menu:
 1-Deposit
 2-Withdraw
 0-Exit
 Your choice: 2
Enter amount to withdraw: 90000
Enter account number: 65664

Name: harshu
PAN: 1244
Account Number: 65664
Branch: adambakkam
Current Balance: 80000.0
Account Number 65664 has produced NotEnoughMoneyInAccountException: Insufficient

Menu:
 1-Deposit
 2-Withdraw
 0-Exit
 Your choice: 2
Enter amount to withdraw: 40000
Enter account number: 3333
Account Number 65664 has produced AccountNotFoundException: Requested Account

Menu:
 1-Deposit
 2-Withdraw
 0-Exit
 Your choice: 0
 */