JAVA Assignment

Question-1:

import java.util.\*;

import java.lang.\*;

class superclass{

public int sum(int a,int b){

int c=a+b;

return c;

}

}

class subclass extends superclass {

}

class thirdclass{

public static void main(String[ ] args){

subclass obj=new subclass();

Scanner sc = new Scanner(System.in);

System.out.println("enter the two nummbers");

int a=sc.nextInt();

int b=sc.nextInt();

int sum=obj.sum(a,b);

System.out.println("Sum = "+(sum));

}

}

Question-2:

import java.util.\*;

import java.lang.\*;

interface divisors{

public int dsum(int num);

}

class div implements divisors{

public int dsum(int num){

int sum=0;

for(int i=1;i<=num;i++){

if(num%i==0){

sum=sum+i;

}

}

return sum;

}

}

class main{

public static void main(String[ ] args){

div ans = new div();

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

int a=ans.dsum(n);

System.out.println("the sum of divisors =" + (a));

}

}

Question-3:

import java.util.\*;

import java.lang.\*;

class stringsorting{

public static void main(String[ ] args){

Scanner sc = new Scanner(System.in);

System.out.println("enter the size of array");

int n=sc.nextInt();

String[] st=new String[n];

for(int i=0;i<n;i++){

st[i]=sc.next();

}

Arrays.sort(st, new Comparator<String>() {

public int compare( String s1, String s2){

double d1 = Double.parseDouble(s1);

double d2 = Double.parseDouble(s2);

return Double.compare(d2, d1);

}

}

);

for(int i=0;i<n;i++){

System.out.print(" "+st[i]);

}

}

}

Question-4:

import java.util.\*;

import java.lang.\*;

class exception {

public static void main(String[ ] args) {

try {

Scanner sc= new Scanner(System.in);

System.out.println("Enter the value of a and b");

int a = sc.nextInt();

int b = sc.nextInt();

System.out.println(a/b);

} catch (ArithmeticException e) {

System.out.println(e);

}

catch (InputMismatchException ex) {

System.out.println(ex.getClass());

}

}

}

Question-5:

// SBIBankAccount class

class SBIBankAccount {

protected double balance;

public void deposit(double amount) {

balance += amount;

System.out.println("Amount " + amount + " deposited successfully.");

}

public void withdraw(double amount) {

if (balance >= amount) {

balance -= amount;

System.out.println("Amount " + amount + " withdrawn successfully.");

} else {

System.out.println("Insufficient balance. Withdrawal failed.");

}

}

public void openAccount() {

System.out.println("Account opened successfully.");

}

}

// PremiumAccount subclass

class PremiumAccount extends SBIBankAccount {

@Override

public void deposit(double amount) {

if (amount >= 5000) {

balance += amount;

System.out.println("Amount " + amount + " deposited successfully to Premium Account.");

} else {

System.out.println("Minimum deposit amount for Premium Account is 5000.");

}

}

@Override

public void withdraw(double amount) {

if (balance - amount >= 5000) {

balance -= amount;

System.out.println("Amount " + amount + " withdrawn successfully from Premium Account.");

} else {

System.out.println("Withdrawal failed. Minimum balance of 5000 required for Premium Account.");

}

}

@Override

public void openAccount() {

balance = 5000;

System.out.println("Premium Account opened successfully with initial balance of 5000.");

}

}

// JundhanAccount subclass

class JundhanAccount extends SBIBankAccount {

@Override

public void deposit(double amount) {

balance += amount;

System.out.println("Amount " + amount + " deposited successfully to Jundhan Account.");

}

@Override

public void withdraw(double amount) {

if (balance - amount >= 0) {

balance -= amount;

System.out.println("Amount " + amount + " withdrawn successfully from Jundhan Account.");

} else {

System.out.println("Withdrawal failed. Insufficient balance in Jundhan Account.");

}

}

@Override

public void openAccount() {

balance = 0;

System.out.println("Jundhan Account opened successfully with initial balance of 0.");

}

}

// Main class to test the bank accounts

class bank {

public static void main(String[] args) {

// Testing PremiumAccount

PremiumAccount premiumAccount = new PremiumAccount();

premiumAccount.openAccount();

premiumAccount.deposit(6000);

premiumAccount.withdraw(3000);

premiumAccount.withdraw(4000);

System.out.println("Premium Account balance: " + premiumAccount.balance);

System.out.println();

// Testing JundhanAccount

JundhanAccount jundhanAccount = new JundhanAccount();

jundhanAccount.openAccount();

jundhanAccount.deposit(2000);

jundhanAccount.withdraw(1500);

jundhanAccount.withdraw(1000);

System.out.println("Jundhan Account balance: " + jundhanAccount.balance);

}

}

Question-6:

import java.util.\*;

// Account interface

interface Account {

void deposit(double amount);

void withdraw(double amount);

double calculateInterest();

double viewBalance();

}

// SavingsAccount class implementing Account interface

class SavingsAccount implements Account {

private double balance;

private double interestRate;

public SavingsAccount(double balance, double interestRate) {

this.balance = balance;

this.interestRate = interestRate;

}

@Override

public void deposit(double amount) {

balance += amount;

System.out.println("Amount " + amount + " deposited successfully to Savings Account.");

}

@Override

public void withdraw(double amount) {

if (balance >= amount) {

balance -= amount;

System.out.println("Amount " + amount + " withdrawn successfully from Savings Account.");

} else {

System.out.println("Insufficient balance. Withdrawal failed.");

}

}

@Override

public double calculateInterest() {

return balance \* interestRate;

}

@Override

public double viewBalance() {

return balance;

}

public void addInterest() {

double interest = calculateInterest();

balance += interest;

System.out.println("Interest added to Savings Account: " + interest);

}

}

// CurrentAccount class implementing Account interface

class CurrentAccount implements Account {

private double balance;

private double overdraftLimit;

public CurrentAccount(double balance, double overdraftLimit) {

this.balance = balance;

this.overdraftLimit = overdraftLimit;

}

@Override

public void deposit(double amount) {

balance += amount;

System.out.println("Amount " + amount + " deposited successfully to Current Account.");

}

@Override

public void withdraw(double amount) {

if (balance + overdraftLimit >= amount) {

balance -= amount;

System.out.println("Amount " + amount + " withdrawn successfully from Current Account.");

} else {

System.out.println("Insufficient balance. Withdrawal failed.");

}

}

@Override

public double calculateInterest() {

return 0; // Current account doesn't earn interest

}

@Override

public double viewBalance() {

return balance;

}

public void setOverdraftLimit(double overdraftLimit) {

this.overdraftLimit = overdraftLimit;

System.out.println("Overdraft limit updated: " + overdraftLimit);

}

}

// Bank class

class Bank {

private List<Account> accounts;

public Bank() {

accounts = new ArrayList<>();

}

public void addAccount(Account account) {

accounts.add(account);

}

}

// Main class to test the banking system

class savings {

public static void main(String[] args) {

Bank bank = new Bank();

SavingsAccount savingsAccount = new SavingsAccount(5000, 0.05);

bank.addAccount(savingsAccount);

CurrentAccount currentAccount = new CurrentAccount(2000, 5000);

bank.addAccount(currentAccount);

savingsAccount.deposit(1000);

savingsAccount.withdraw(2000);

double savingsBalance = savingsAccount.viewBalance();

System.out.println("Savings Account Balance: " + savingsBalance);

currentAccount.deposit(3000);

currentAccount.withdraw(5000);

double currentBalance = currentAccount.viewBalance();

System.out.println("Current Account Balance: " + currentBalance);

savingsAccount.addInterest();

double updatedSavingsBalance = savingsAccount.viewBalance();

System.out.println("Updated Savings Account Balance: " + updatedSavingsBalance);

currentAccount.setOverdraftLimit(10000);

}

}