**Write a java program to print person details in the format as shown below:**

**Person Details:**

**\_\_\_\_\_\_\_\_\_\_\_\_**

**First Name: Suyash**

**Last Name: Bharambe**

**Gender: M**

**Age: 23**

**Weight: 80.00**

**Solution:**

**public** **class** ProjectDetails {

String firstname,lastname;

**char** gender;

**int** age;

**float** weight;

ProjectDetails(String firstname, String lastname, **char** gender, **int** age, **float** weight)

{

**this**.firstname=firstname;

**this**.lastname=lastname;

**this**.gender=gender;

**this**.age=age;

**this**.weight=weight;

}

**void** display()

{

System.***out***.println("Person Details");

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

System.***out***.println("First Name: " + **this**.firstname);

System.***out***.println("Last Name: " + **this**.lastname);

System.***out***.println("Gender: " + **this**.gender);

System.***out***.println("Age: " + **this**.age);

System.***out***.println("Weight: " + **this**.weight);

}

**public** **static** **void** main(String[] args)

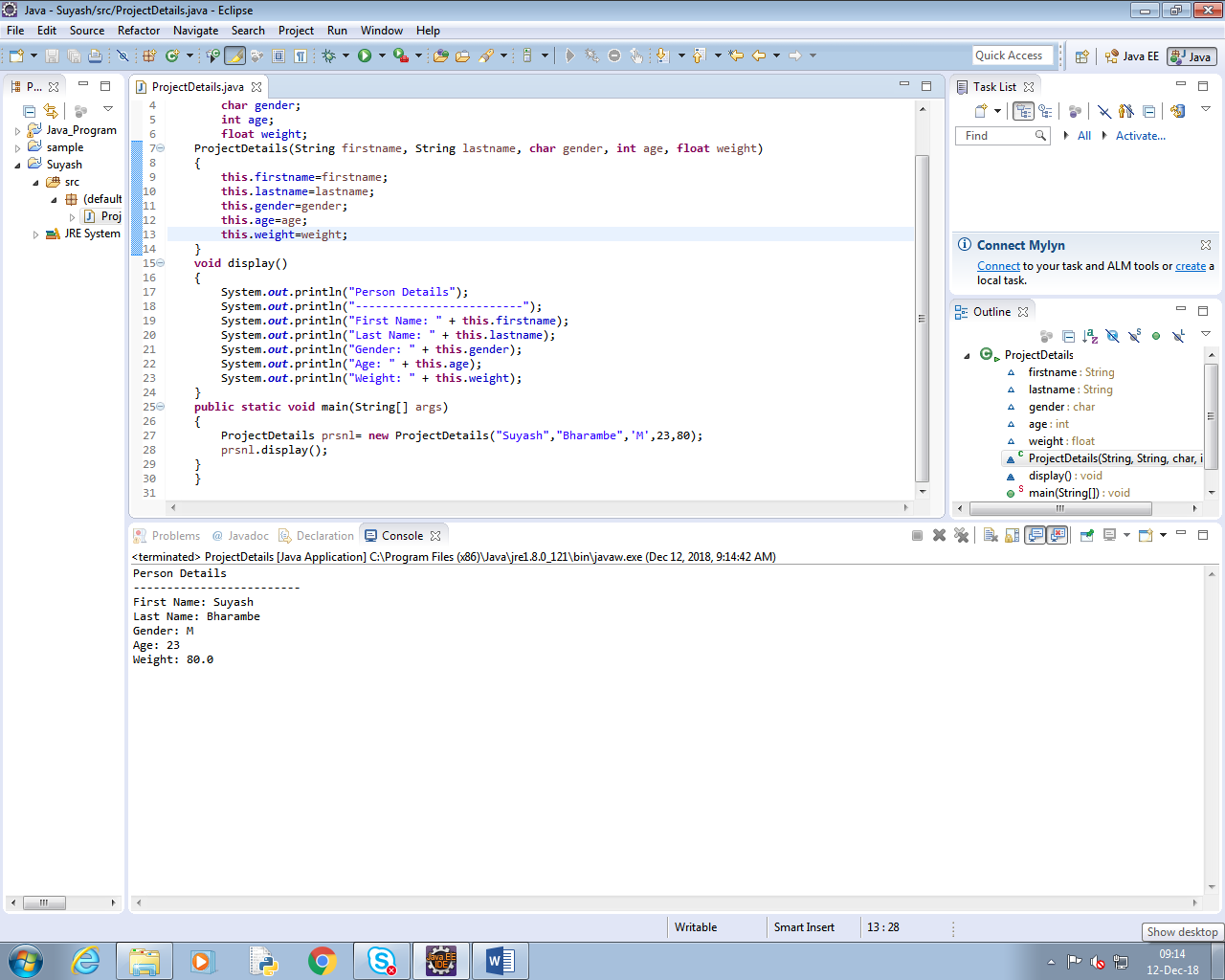
{

ProjectDetails prsnl= **new** ProjectDetails("Suyash","Bharambe",'M',23,80);

prsnl.display();

}

}



**Write a program to accept a number from user as a command line argument and check whether the given number is positive or negative number.**

**Solution:**

**public** **class** number {

**int** num;

number(**int** num)

{

**this**.num=num;

}

**void** checkNumber()

{

**if**(num>=0)

System.***out***.println("no. is positive");

**else**

System.***out***.println("no. is negative");

}

**public** **static** **void** main(String[] args)

{

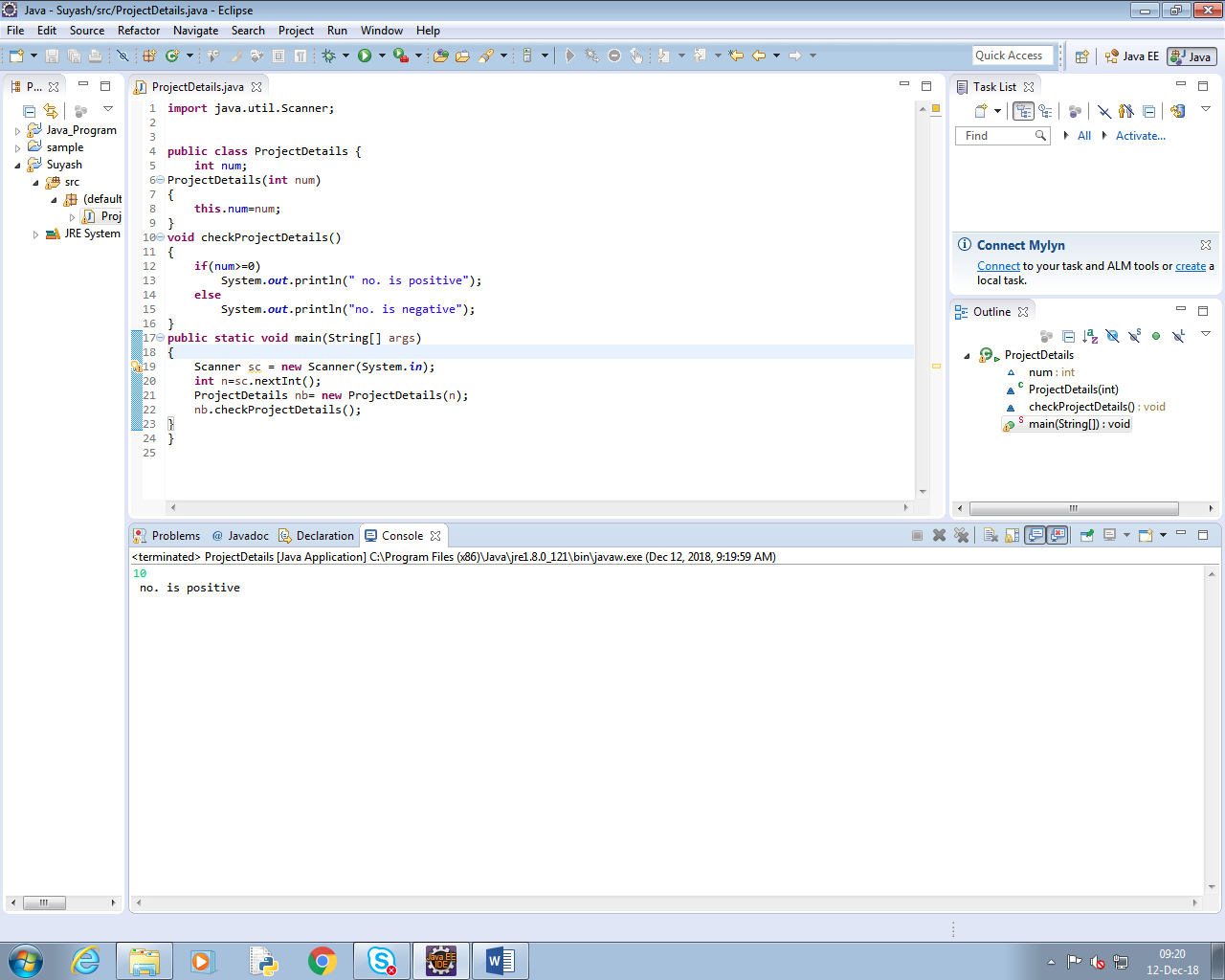
**int** n=Integer.*parseInt*(args[0]);

number nb= **new** number(n);

nb.checkNumber();

}

}



**Refer the class diagram given below and create a person class.**



**Create default and parameterized constructor for Person class.**

**Solution:**

**public** **class** Person {

**private** String firstname;

**private** String lastname;

**private** **char** gender;

Person(String firstname, String lastname, **char** gender)

{

**this**.firstname =firstname;

**this**.lastname= lastname;

**this**.gender=gender;

}

**void** display()

{

System.***out***.println("First Name: "+**this**.firstname);

System.***out***.println("Last Name: "+**this**.lastname);

System.***out***.println("Gender: "+**this**.gender);

}

**public** **static** **void** main(String[] args)

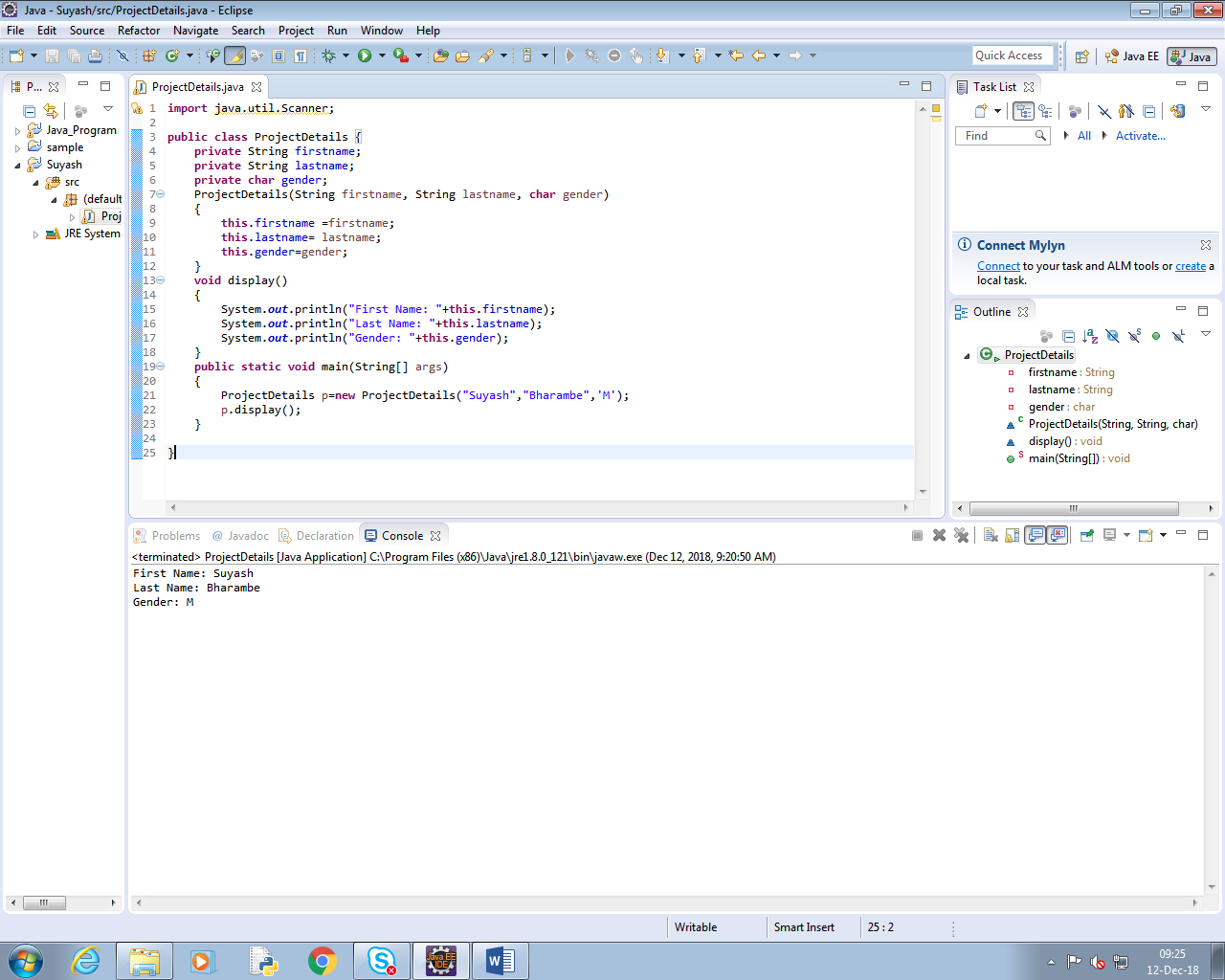
{

Person p=**new** Person("Suyash","Bharambe",'M');

p.display();

}

}



**create Account Class as shown below in class diagram. Ensure minimum balance of INR 500 in a bank account is available.**

****

**a) Create Account for smith with initial balance as INR 2000 and for Kathy with initial balance as 3000.(accNum should be auto generated).**

**b) Deposit 2000 INR to smith account.**

**c) Withdraw 2000 INR from Kathy account.**

**d) Display updated balances in both the account.**

**e) Generate toString() method.**

**Solution:**

**public** **class** ProjectDetails {

String name;

Float age;

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** Float getAge() {

**return** age;

}

**public** **void** setAge(Float age) {

**this**.age = age;

}

**public** String toString1() {

**return** "ProjectDetails[Name=" + name + ",age=" + age + "]";

}**long** accnum;

**static** **long** *accno*=7001;

**double** bal;

**public** **long** getAccnum() {

**return** accnum;

}

**public** **void** setAccnum(**long** accnum) {

**this**.accnum=*accno*;

*accno*++;

//this.accnum = accnum;

}

**public** **double** getBal() {

**return** bal;

}

**public** **void** setBal(**double** bal) {

**this**.bal = bal;

}

**void** deposit(**double** n)

{

**this**.bal=**this**.bal+n;

}

**void** withdraw(**double** n)

{

**this**.bal=**this**.bal-n;

}

**double** getBalance()

{

**return** **this**.bal;

}

**public** String toString() {

**return** "ProjectDetails[Accountno.=" + accnum+ ",Balance=" + bal + "]";

}

**public** **static** **void** main(String[] args)

{

ProjectDetails a1=**new** ProjectDetails();

ProjectDetails a2 = **new** ProjectDetails();

a1.setName("Smith");

a1.setAge((**float**)15.0);

a1.setAccnum(0);

a1.setBal(2000);

a1.deposit(1000);

System.***out***.println(a1.name +","+a1.getAccnum()+" = " +a1.getBalance());

a2.setName("Kathy");

a2.setAge((**float**)45.0);

a2.setAccnum(0);

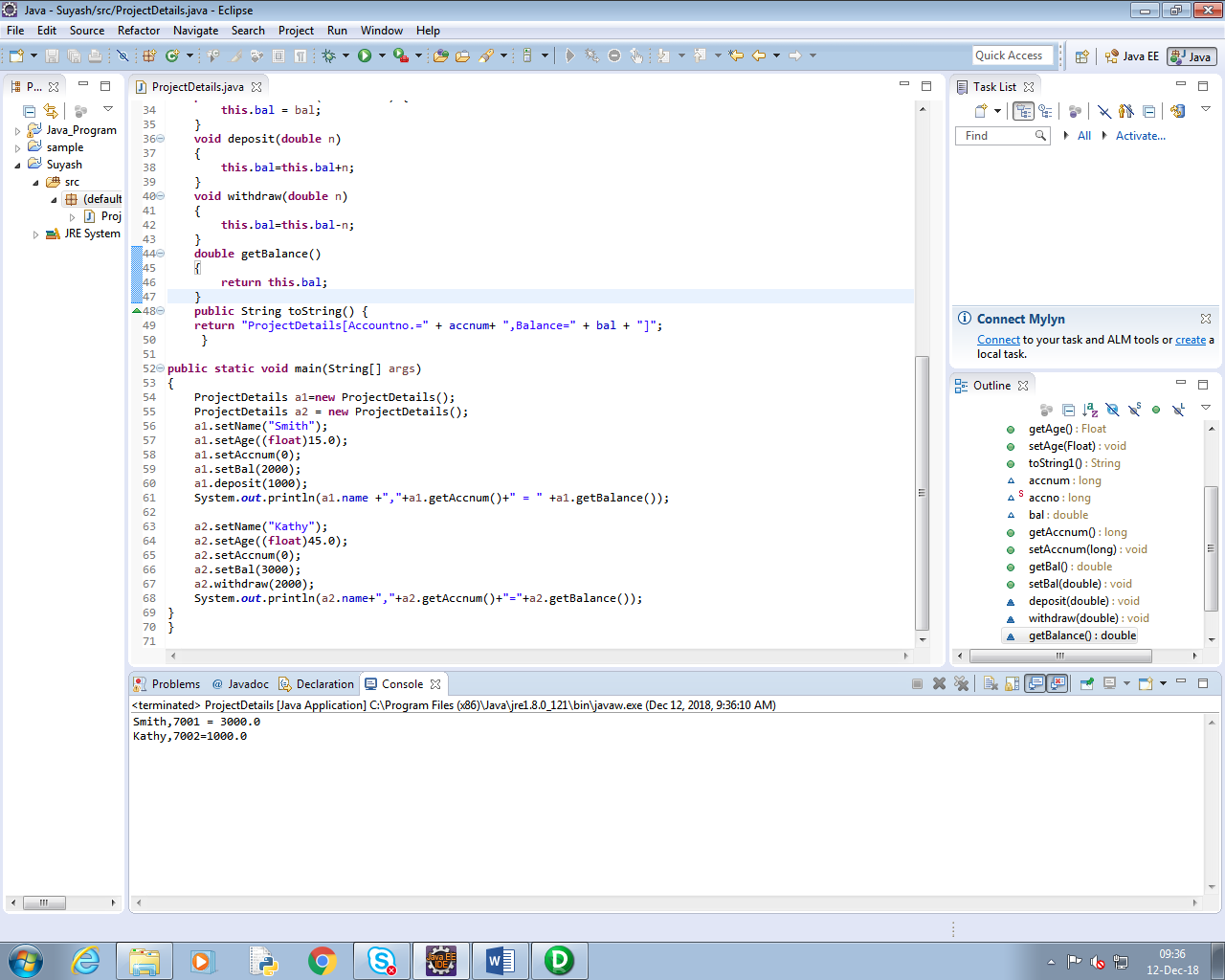
a2.setBal(3000);

a2.withdraw(2000);

System.***out***.println(a2.name+","+a2.getAccnum()+"="+a2.getBalance());

}

}



**Inherit two classes Savings Account and Current Account from account class. Implement the following in the respective classes.**

**a) Savings Account**

**a. Add a variable called minimum Balance and assign final modifier.**

**b. Override method called withdraw (This method should check for minimum balance and allow withdraw to happen)**

**b) Current Account**

**a. Add a variable called overdraft Limit**

**b. Override method called withdraw (checks whether overdraft limit is reached and returns a boolean value accordingly)**

**Solution:**

**public** **class** Accounts {

**long** accno;

**double** bal;

**public** **long** getAccno() {

**return** accno;

}

**public** **void** setAccno(**long** accno) {

**this**.accno = accno;

}

**public** **double** getBal() {

**return** bal;

}

**public** **void** setBal(**double** bal) {

**this**.bal = bal;

}

**boolean** withdraw()

{

**if**(bal==0)

**return** **true**;

**else**

**return** **false**;

}

}

**public** **class** CurrentAccount **extends** Accounts {

**final** **double** overdraft\_limit=2000;

@Override

**boolean** withdraw()

{

**if**(**this**.bal==overdraft\_limit)

{

**return** **true**;

}

**else**

{

**return** **false**;

}

}

**public** **static** **void** main(String[] args)

{

SavingAccount sa=**new** SavingAccount();

sa.setAccno(1208900);

sa.setBal(1500);

**boolean** a=sa.withdraw();

**if**(a==**true**)

System.***out***.println("Cannot remove from account");

**else**

System.***out***.println("Money removed from account");

CurrentAccount ca=**new** CurrentAccount();

ca.setAccno(874365873);

ca.setBal(6799);

**boolean** b=ca.withdraw();

**if**(b==**true**)

System.***out***.println("Cannot remove from account");

**else**

System.***out***.println("Money removed from account");

}

}

**public** **class** SavingAccount **extends** Accounts {

**final** **double** minbal=2000;

@Override

**boolean** withdraw()

{

**if**(**this**.bal<=minbal)

{

**return** **true**;

}

**else**

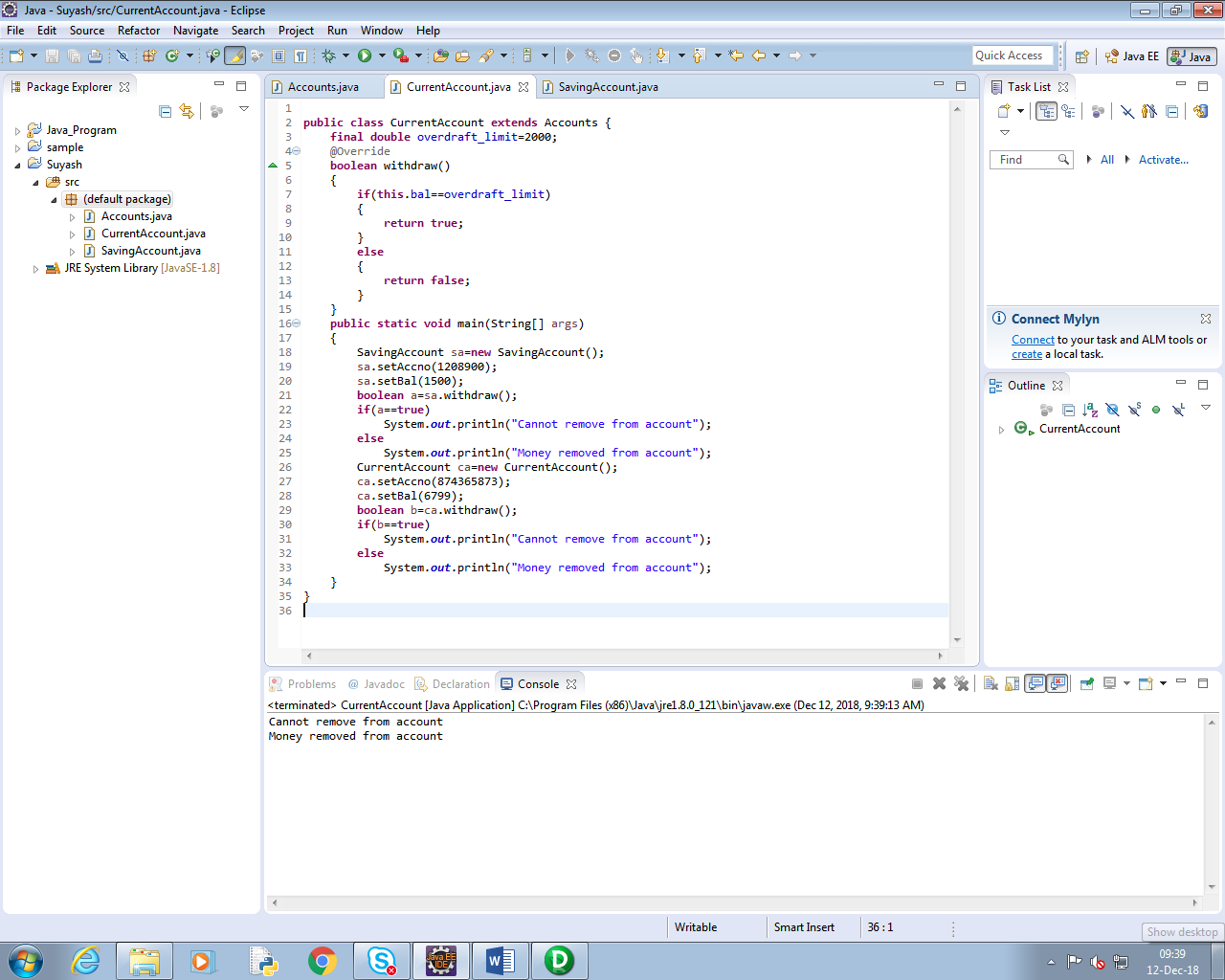
{

**return** **false**;

}

}

}



**Refer the case study 2 in page no: 5 and create an application for that requirement by creating packages and classes as given below:**

**a) com.cg.eis.bean**

**In this package, create “Employee” class with different attributes such as id, name, salary, designation, insuranceScheme.**

**b) com.cg.eis.service**

**This package will contain code for services offered in Employee Insurance System. The service class will have one EmployeeService Interface and its corresponding implementation class.**

**c) com.cg.eis.pl**

**This package will contain code for getting input from user, produce expected output to the user and invoke services offered by the system.**

**The services offered by this application currently are:**

**i) Get employee details from user.**

**ii) Find the insurance scheme for an employee based on salary and designation.**

**iii) Display all the details of an employee.**

**Use overrides annotation for the overridden methods available in a derived class of an interface of all the assignments.**

**Refer the problem statement 4.1. Modify account class as abstract class and declare withdraw method.**

**Solution:**

**public** **class** Employee {

**int** id;

String name;

**public** **double** salary;

**public** String designation;

String insurance\_scheme;

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **double** getSalary() {

**return** salary;

}

**public** **void** setSalary(**double** salary) {

**this**.salary = salary;

}

**public** String getDesignation() {

**return** designation;

}

**public** **void** setDesignation(String designation) {

**this**.designation = designation;

}

**public** String getInsurance\_scheme() {

**return** insurance\_scheme;

}

**public** **void** setInsurance\_scheme(String insurance\_scheme) {

**this**.insurance\_scheme = insurance\_scheme;

}

}

package com.cg.eis.pl;

import java.util.Scanner;

import com.cg.eis.bean.\*;

import com.cg.eis.service.\*;

public class Details {

public static void main(String[] args){

Scanner sc =new Scanner(System.in);

Employee e =new Employee();

e.setName(sc.next());

e.setId(sc.nextInt());

//String designation=sc.next();

e.setDesignation(sc.next());

//double sal=sc.nextDouble();

e.setSalary(sc.nextDouble());

//e.setInsurance\_scheme(sc.next());

Service s=new Service();

s.insuranceScheme(e);

System.out.println("Name: "+e.getName());

System.out.println("Id: "+e.getId());

System.out.println("Designation: "+e.getDesignation());

System.out.println("Salary: "+e.getSalary());

System.out.println("Insurance\_Scheme: "+e.getInsurance\_scheme());

sc.close();

}}

**package** com.cg.eis.service;

**import** com.cg.eis.bean.\*;

**public** **interface** EmployeeService {

**public** **void** insuranceScheme(Employee e);

}

package com.cg.eis.service;

//import com.cg.eis.exception;

import com.cg.eis.exception.EmployeeException;

import com.cg.eis.bean.\*;

public class Service implements EmployeeService {

//Employee e = new Employee();

//@override

static void validate(double salary)throws EmployeeException{

if(salary<3000)

throw new EmployeeException("Salary is below 3000"); }

//override

public void insuranceScheme(Employee e) {

// TODO Auto-generated method stub

try{

validate(e.salary);

}

catch(Exception m){

System.out.println("Exception occured: "+m);

}

if (e.salary>5000 && e.salary< 20000 && e.designation.equals("System Associate"))

{

e.setInsurance\_scheme("Scheme C");

}

else if(e.salary>20000 && e.salary< 40000 && e.designation.equals("Programmer"))

{

e.setInsurance\_scheme("Scheme B");

}

else if(e.salary>=40000 && e.designation.equals("Manager"))

{

e.setInsurance\_scheme("Scheme A");

}

else

{

e.setInsurance\_scheme("No Scheme");

}

}

}

**Modify the Lab assignment 2.3 to validate the full name of an employee. Create and throw a user defined exception if firstName and lastName is blank**

**Solution:**

import java.lang.\*;

import java.io.\*;

import java.util.Scanner;

public class NameBlank {

private String firstname;

private String lastname;

protected char gender;

NameBlank(String firstname, String lastname, char gender)

{

this.firstname =firstname;

this.lastname= lastname;

this.gender=gender;

}

static void validate(String name)throws InvalidAgeException{

if(name==null)

throw new InvalidAgeException("not valid"); }

void display()

{

try{

validate(lastname);

validate(firstname);

}

catch(Exception m){

System.out.println("Exception occured: "+m);

}

System.out.println("First Name: "+this.firstname);

System.out.println("Last Name: "+this.lastname);

System.out.println("Gender: "+this.gender);

}

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

/\* String firstname=sc.next();

String lastname=sc.next();

char gender=sc.next().charAt(0);\*/

NameBlank e=new NameBlank(null,null ,'f');

e.display();

sc.close();

}

}

**Validate the age of a person in Lab assignment 4.2 and display proper message by using user defined exception. Age of a person should be above 15.**

**Solution:**

**import** java.util.Scanner;

**public** **class** AgeValidation {

**private** String firstname;

**private** String lastname;

**protected** **int** age;

AgeValidation(String firstname, String lastname, **int** age)

{

**this**.firstname =firstname;

**this**.lastname= lastname;

**this**.age=age;

}

**static** **void** validate(**int** age)**throws** AgeException{

**if**(age<=15)

**throw** **new** AgeException("Age of a person should be above 15"); }

**void** display()

{

**try**{

*validate*(age);

}

**catch**(Exception m){

System.***out***.println("Exception occured: "+m);

}

System.***out***.println("First Name: "+**this**.firstname);

System.***out***.println("Last Name: "+**this**.lastname);

System.***out***.println("Gender: "+**this**.age);

}

**public** **static** **void** main(String[] args)

{

Scanner sc=**new** Scanner(System.***in***);

/\* String firstname=sc.next();

String lastname=sc.next();

char gender=sc.next().charAt(0);\*/

AgeValidation e=**new** AgeValidation("Suyash","Bharambe" ,23);

e.display();

sc.close();

}

}

@SuppressWarnings("serial")

**public** **class** AgeException **extends** Exception {

AgeException(String s){

**super**(s);

}

}

