Name of the Course : Java for beginners: Step–by–step hands-on guide to Java

Level : Difficult

Tool Stack : Encapsulation, basic java programming and access specifires

Problem Statement :

SNMR College of Engineering and Technology wants to create an application to store their students details as well as the details of hostellers.

In case of any changes to be made to the attributes, admin can update the details like room number and phone number of the hosteler.

Develop a program to implement this scenario.

Description : Create a public class Student with protected attributes :

int studentId

String name

int departmentId

String gender

String phone

Create a public class Hosteller with private attributes

String hostelName

int roomNumber

Make this class inherit the Student class, as it holds all the properties of Student. Use appropriate public getters and setters for both the classes. Write a class Main with the main function. In Main class get the input of the hosteller using the method :

public static Hosteller getHostellerDetails().

Invoke this method from the main method and then modify the room number and phone number, if needed.

**Code:**

**import** java.util.Scanner;

**public** **class** Hosteller **extends** Student {

**private** String hostelName;

**private** **int** roomNumber;

**public** String getHostelName() {

**return** hostelName;

}

**public** **void** setHostelName(String hostelName) {

**this**.hostelName = hostelName;

}

**public** **int** getRoomNumber() {

**return** roomNumber;

}

**public** **void** setRoomNumber(**int** roomNumber) {

**this**.roomNumber = roomNumber;

}

}

**import** java.util.Scanner;

**public** **class** Main {

**public** **static** Hosteller getHostellerDetails() {

Hosteller h=**new** Hosteller();

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

System.***out***.println("Student Id:");

h.setStudentId(sc.nextInt());

System.***out***.println("Student Name:");

h.setName(sc.next());

System.***out***.println("Department Id:");

h.setDepartmentId(sc.nextInt());

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

h.setGender(sc.next());

System.***out***.println("Phone Number");

h.setPhone(sc.next());

System.***out***.println("Hostel name:");

h.setHostelName(sc.next());

System.***out***.println("Room number");

h.setRoomNumber(sc.nextInt());

System.***out***.println("Want Modify Room Number(Y/N)");

String modify=sc.next();

**if**(modify.equals("Y")) {

System.***out***.println("New Room Number");

h.setRoomNumber(sc.nextInt());

}

System.***out***.println("Want Modify Phone Number(Y/N)");

String phone=sc.next();

**if**(phone.equals("Y")) {

System.***out***.println("New Phone Number");

h.setPhone(sc.next());

}

System.***out***.println("The Student Details:");

System.***out***.println(h.getStudentId()+" "+h.getName()+" "+h.getDepartmentId()+" "

+h.getGender()+" "+h.getPhone()+" "+h.getHostelName()+" "+h.getRoomNumber());

**return** h;

}

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

Hosteller s=Main.*getHostellerDetails*();

}

}

**public** **class** Student {

**protected** **int** studentId;

**protected** String name;

**protected** **int** departmentId;

**protected** String gender;

**protected** String phone;

**public** **int** getStudentId() {

**return** studentId;

}

**public** **void** setStudentId(**int** studentId) {

**this**.studentId = studentId;

}

**public** String getName() {

**return** name;

}

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

**this**.name = name;

}

**public** **int** getDepartmentId() {

**return** departmentId;

}

**public** **void** setDepartmentId(**int** departmentId) {

**this**.departmentId = departmentId;

}

**public** String getGender() {

**return** gender;

}

**public** **void** setGender(String gender) {

**this**.gender = gender;

}

**public** String getPhone() {

**return** phone;

}

**public** **void** setPhone(String phone) {

**this**.phone = phone;

}

}

Pom.xml File

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>Contact\_Details\_of\_Hosteller</groupId>

<artifactId>Contact\_Details\_of\_Hosteller</artifactId>

<version>0.0.1-SNAPSHOT</version>

<build>

<sourceDirectory>src</sourceDirectory>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.0</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</build>

</project>

Junit Testing

**import** **static** org.junit.Assert.*assertTrue*;

**import** **static** org.junit.jupiter.api.Assertions.\*;

**import** org.junit.jupiter.api.Test;

**import** difficult\_Hosteller.Hosteller;

**import** difficult\_Hosteller.Main;

**class** MainTest {

@Test

**void** testGetHostellerDetails() {

String modify="Y",phone="Y";

Hosteller h=**new** Hosteller();

// h=Main.getHostellerDetails();

h.setStudentId(1);

h.setName("Rajnikant");

h.setDepartmentId(101);

h.setGender("Male");

h.setPhone("7895246366");

h.setHostelName("SNMR");

h.setRoomNumber(108);

**if**(modify.equals("Y")) {

h.setRoomNumber(110);

}

**if**(modify.equals("Y")) {

h.setPhone("7709304588");

}

h=Main.*getHostellerDetails*();

*assertEquals*(**true**, h **instanceof** Hosteller );

}

}

Test Data1

Student Id:

1

Student Name:

Rajnikant

Department Id:

101

Gender

Male

Phone Number

7895246366

Hostel name:

SNMR

Room number

108

Want Modify Room Number(Y/N)

Y

New Room Number

110

Want Modify Phone Number(Y/N)

Y

New Phone Number

7709304588

The Student Details:

1 Rajnikant 101 Male 7709304588 SNMR 110

Test Data2

Student Id

1

Student Name

John

Department Id

101

Gender

Male

Phone Number

9876543210

Hostel Name

YMCA

Room Number

10

Want Modify Room Number(Y/N)

Y

New Room Number

11

Want Modify Phone Number(Y/N)

Y

New Phone Number

9876543121

The Student Details

1 John 101 Male 9876543121 YMCA 11

Learning outcome: Participant could able to learn how to use inheritance, encapsulation and protected access specifires.