**0//ตัวอย่างที่ 1//Implementation for Composition**

class A {

B

A

private int a;

public A() {a=0;} //constructor 1

public A(int x){a=x;} //constructor 2

public int getData() {

return a;

}

public void setData(int x) {

a=x;

}

} //end class A

class B{

private int b;

private A objA;

public B() {b = 0; objA = new A();} //constructor 1

public B(int valB) { b = valB; objA = new A();} //constructor 2

public B(int valB, int valA) {b = valB; objA = new A(valA);} //constructor 3

public void setB(int valB) {

b=valB;

}

public void setA(int valA) {

objA.setData(valA);

}

public void display() {

System.out.println("b = " + b + " a in objA = " + objA.getData());

}

} //end class B

public class JavaApp {

public static void main(String[] args) {

B objB1 = new B();

objB1.setB(1);

objB1.setA(101);

ผลการทำงาน

b = 1 a in objA = 101

b = 2 a in objA = 201

b = 3 a in objA = 301

B objpB2=new B(2);

objpB2.setA(201);

B objB3 = new B(3,301);

objB1.display();

objpB2.display();

objB3.display();

}

}

**//ตัวอย่างที่ 2 //Implementation for Composition and Aggregation**

Faculty

Building

Room

class Room {

private String rname; //ชื่อห้อง

public Room(){};

public Room( String n){ rname = n;};

public String getName() {return rname;} ;

public void setName( String n ){ rname=n; };

} //end class Room

class Building {

private int buildingId; //รหัสตึก

private String buildingName; //ชื่อตึก

private Room [] room= new Room[10]; //ห้องในตึก

private int nRoom; //จำนวนห้อง

public Building(){nRoom = 0;}; //constructor1, automatically call Room()

//so Room() must be declared

public Building(int id, String n){ //constructor2

buildingId = id;

buildingName =n;

nRoom=0;

}

public void setId(int id) {buildingId=id;}

public void setName(String n) {buildingName = n;}

public String getName() {return buildingName;}

public void addRoom( String rname) {

if ((nRoom >= 0) && (nRoom < 10)){

room[nRoom] = new Room();

room[nRoom].setName(rname);

nRoom++;

}

}

public void setRoom (int i, String rname) {

if (i < 10) room[i].setName(rname);

}

public void display() {

System.out.println("Building : " + buildingId + ' ' + buildingName);

for (int i=0;i<nRoom;i++)

System.out.println("room " + (i+1) + " " + room[i].getName());

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

}

} //end class Building

class Faculty {

private int fid; //รหัสคณะ

private String fname; //ชื่อคณะ

private Building[] bdl = new Building[10]; //ตึกในคณะ

private int nBdl; //จำนวนตึก

Faculty(int id, String nameFac){ //constructor

fid=id;

fname=nameFac;

nBdl=0;

}

void addBuilding(Building bdobj){

if (nBdl < 10) bdl[nBdl++] = bdobj;

}

void display(){

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

System.out.println("in Faculty Reports : " + fid + ' ' + fname);

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

for (int i=0;i<nBdl;i++)

bdl[i].display();

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

}

} //end class Faculty

public class JavaApp {

public static void main(String[] args) {

Building b1= new Building();

b1.setId(1);

b1.setName("Computer Science");

b1.addRoom("CSB101");

b1.addRoom("CSB201");

b1.addRoom("CSB209");

b1.display();

Building b2= new Building(2,"Biology");

Faculty f1;

f1 = new Faculty(5,"SCIENCE");

f1.addBuilding(b1);

f1.addBuilding(b2);

f1.display();

b1.setRoom(0,"CSB100"); //edit room name from CSB101 to CSB100

b2.addRoom("BB111");

b2.addRoom("BB222");

b2.addRoom("BB333");

f1.display();

}

}

ผลการทำงาน

Building : 1 Computer Science

room 1 CSB101

room 2 CSB201

room 3 CSB209

\*\*\*\*\*\*\*\*\*

===================================

in Faculty Reports : 5 SCIENCE

===================================

Building : 1 Computer Science

room 1 CSB101

room 2 CSB201

room 3 CSB209

\*\*\*\*\*\*\*\*\*

Building : 2 Biology

\*\*\*\*\*\*\*\*\*

===================================

===================================

in Faculty Reports : 5 SCIENCE

===================================

Building : 1 Computer Science

room 1 CSB100

room 2 CSB201

room 3 CSB209

\*\*\*\*\*\*\*\*\*

Building : 2 Biology

room 1 BB111

room 2 BB222

room 3 BB333

\*\*\*\*\*\*\*\*\*

===================================

**//ตัวอย่างที่ 3//Implementation of Association : Course-Room Example**

class Course {

private int code; //รหัสกระบวนวิชา

private String courseName; //ชื่อกระบวนวิชา

private Room objRoom; //ห้องที่ใช้

public Course(){}; //constructor 1

public Course (int c, String n ) { //constructor 2

code=c;

courseName=n;

Room

Course

objRoom= new Room();

objRoom = null;

}

public String getCourseName() {return courseName;}

void bookRoom(Room r){ //จองห้อง

if (objRoom != null)

objRoom.unbookCourse();

objRoom = r;

}

void unbookRoom(){ //ยกเลิกการจองห้อง

objRoom = null;

}

void display() { //prototype

System.out.print(objRoom != null? objRoom.getName(): "\*\*\*\*\*\*");

System.out.println(" " + code + ' ' + courseName);

}

} //end class Course

class Room{

private String rname; //ชื่อห้อง

private Course objCourse ; //วิชาที่มาใช้ห้อง

public Room( String n){ //constructor

rname = n;

objCourse = new Course();

objCourse = null;

}

Room() { } //constructor

String getName(){return rname;}

void setName( String n) { rname = n;}

void bookCourse(Course c){

if (objCourse != null)

objCourse.unbookRoom();

objCourse = c;

}

void unbookCourse () { objCourse = null;};

void display(){

System.out.println(rname + " " + (objCourse != null ? objCourse.getCourseName () :

"available") );

}

} //end class Room

public class JavaApp {

public static void booking(Course c, Room r){

c.bookRoom(r);

r.bookCourse(c);

}

public static void main(String[] args) {

Room r100=new Room("CSB100");

Room r308=new Room("CSB308");

Room r207=new Room("CSB207");

Course c101=new Course(101,"Intro to Computer" );

Course c211=new Course(211,"OO Programming");

Course c333=new Course(333,"Internet Application Development");

c101.display(); //not assigned room

c211.display(); //not assigned room

c333.display(); //not assigned room

r100.display(); //not assigned course

r207.display(); //not assigned course

r308.display(); //not assigned course

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

booking(c101, r100);

booking(c211, r207);

c101.display();

c211.display();

c333.display(); //still not assigned room yet

r100.display();

r207.display();

r308.display(); //still not assigned course yet

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

booking(c333, r207); //r207 changes course from c211 to c333

booking(c333, r308); //c333 changes room from r207 to r308

c101.display();

c211.display();

c333.display();

r100.display();

r207.display();

r308.display();

}

}

ผลการทำงาน

\*\*\*\*\*\* 101 Intro to Computer

\*\*\*\*\*\* 211 OO Programming

\*\*\*\*\*\* 333 Internet Application Development

CSB100 available

CSB207 available

CSB308 available

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

CSB100 101 Intro to Computer

CSB207 211 OO Programming

\*\*\*\*\*\* 333 Internet Application Development

CSB100 Intro to Computer

CSB207 OO Programming

CSB308 available

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

CSB100 101 Intro to Computer

\*\*\*\*\*\* 211 OO Programming

CSB308 333 Internet Application Development

CSB100 Intro to Computer

CSB207 available

CSB308 Internet Application Development