

**Name:** Muhammad Sanan Umar

**Registration No:** SP25-BCS-061(A)

**Object Oriented Programming**

**ASSIGNMENT 2**

public class Main{

public static void main(String args[]){

CityHousing ch=new CityHousing("Lahore",2);

ch.showcityhousing();

//Find plot by id

Plot p=ch.findplotbyId("1-001");

System.out.println(p);

}

}public class Block{

private String blockname;

private Plot[][]plots;

private Park[] parks;

private CommercialMarket market;

//private int streetcount;

private static final int[] Streetlengths={10,11,12,13,14};

Block(String blockname){

this.blockname=blockname;

buildplots();

}

public void buildplots(){

int[]streetarr={10,11,12,13,14};

plots=new Plot[streetarr.length][];

for(int i=0;i<streetarr.length;i++){

plots[i]=new Plot[streetarr[i]];

for(int j=0;j<streetarr[i];j++){

String id=(i+1)+"-"+String.format("%03d",(j+1));

Plottype type;

Shapetype shape;

switch(i) {

case 0:

type = Plottype.RES\_5\_MARLA; shape = Shapetype.RECTANGLE;

break;

case 1:

type = Plottype.RES\_10\_MARLA; shape = Shapetype.RECTANGLE;

break;

case 2:

type = Plottype.RES\_1\_KANAL; shape = Shapetype.TRAPEZOID;

break;

case 3:

type = Plottype.COMM\_SHOP; shape = Shapetype.RECTANGLE;

break;

default:

type = Plottype.COMM\_OFFICE; shape = Shapetype.RECTANGLE;

}

if((j+1)%5==0)

type=Plottype.PARKING;

if(i<=2&&(j+1)%4==0)

plots[i][j]=new CornerPlot(id,type,shape,5,4,2,6,1);

else

plots[i][j]=new Plot(id,type,shape,5,4,2,6);

}

}

this.parks=new Park[2];

for(int i=0;i<parks.length;i++){

parks[i]=new Park("Park-"+(i+1),Shapetype.RECTANGLE,3,4,2,6);

}

this.market=new CommercialMarket("Market-"+blockname,15);

}

public void showallplots(){

System.out.println();

System.out.println("All plots in block "+blockname);

System.out.println();

for(int i=0;i<plots.length;i++){

for(int j=0;j<plots[i].length;j++) {

System.out.println(plots[i][j]);

}

}

}

public void showallparks(){

System.out.println();

System.out.println("All parks in block "+blockname);

System.out.println();

for(int i=0;i<parks.length;i++){

System.out.println();

System.out.println(parks[i]);

}

}

public void showmarket(){

System.out.println();

System.out.println("All shops in block "+blockname);

System.out.println();

System.out.println();

market.showallshops();

}

public Plot findplotbyId(String id){

for(int i=0;i<plots.length;i++){

for(int j=0;j<plots[i].length;j++){

Plot p1=plots[i][j];

if(p1.getid().equals(id)){

return p1;

}

}

}

return null;

}

public boolean bookplot(String id){

Plot p=findplotbyId(id);

if(p!=null){

return p.book();

}

else

return false;

}

public boolean cancelplot(String id){

Plot p=findplotbyId(id);

if(p!=null){

return p.cancel();

}

else

return false;

}

/\* public double calculateBlockValue() {

double sum = 0;

for (int i = 0; i < plots.length; i++) {

for (int j = 0; j < plots[i].length; j++) {

sum += plots[i][j].getprice();

}

}

return sum;

}\*/

}public class CityHousing{

private String cityname;

private HousingSociety[] societies;

CityHousing(String cityname,int n){

this.cityname=cityname;

societies=new HousingSociety[n];

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

societies[i]=new HousingSociety("LDA AVENUE "+(i+1),3);

}

}

public void showcityhousing(){

for(int i=0;i<societies.length;i++){

societies[i].showallblocks();

}

}

public Plot findplotbyId(String id){

for(int i=0;i<societies.length;i++){

Plot p=societies[i].findplotbyId(id);

if(p!=null){

return p;

}

}

return null;

}

}public class CommercialMarket{

private String id;

private Shop[] shops;

CommercialMarket(String id,int n){

this.id=id;

this.shops=new Shop[n];

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

shops[i]=new Shop(id+"- Shop:"+(i+1),Shapetype.TRAPEZOID,Plottype.COMM\_SHOP);

}

}

public void showallshops(){

System.out.println("Market:"+id);

for(int i=0;i<shops.length;i++){

System.out.println(shops[i]);

}

}

}public class CornerPlot extends Plot{

private double extrawidth;

private final double premiumrate=0.08;

private double extraprice;

private double baseprice;

CornerPlot(String id,Plottype type,Shapetype shape,double width,double depth,double front,double back,double extrawidth){

super(id,type,shape,width,depth,front,back);

this.extrawidth=extrawidth;

applyCornerPrice();

this.baseprice=type.getprice();

}

public void applyCornerPrice() {

extraprice=getprice()\*premiumrate\*extrawidth;;

double newprice=extraprice+getprice();

setprice(newprice);

//double newPrice = getprice() + (getprice() \* premiumrate);

//setprice(newPrice); // Update parent class price

}

/\*public String toString(){

return String.format("%s %.2f %.2f Final price=%.2f Rs",super.toString(),premiumrate \* 100,extraprice,getprice());

}\*/

@Override

public String toString() {

return String.format("%s |Premium rate: %.2f |Base price: %.2f ",super.toString(),premiumrate\*100,this.baseprice);

}

}public class HousingSociety{

private String name;

private Block[]blocks;

HousingSociety(String name,int n){

this.name=name;

blocks=new Block[n];

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

String blockname=name+" Block "+(char)('A'+i);

blocks[i]=new Block(blockname);

}

}

public void showallblocks(){

System.out.println("Housing society "+name);

for(int i=0;i<blocks.length;i++){

blocks[i].showallplots();

blocks[i].showallparks();

blocks[i].showmarket();

System.out.println();

}

}

public Plot findplotbyId(String id){

for(int i=0;i<blocks.length;i++){

Plot p=blocks[i].findplotbyId(id);

if(p!=null){

return p;

}

}

return null;

}

}public class Park{

private String id;

private Shapetype shape;

private double width;

private double depth;

private double front;

private double back;

Park(String id,Shapetype shape,double width,double depth,double front,double back){

this.id=id;

this.shape=shape;

this.width=width;

this.depth=depth;

this.back=back;

this.front=front;

}

public String toString(){

return String.format("Park: %s , Shape: %s , Dimensions:W:%.2f,D:%.2f,F:%.2f, B:%.2f",id,shape,width,depth,front,back);

}

} public class Plot{

private String id;

private Plottype type;

private Shapetype shape;

private double area;

private double price;

private boolean isavailable;

private double width;

private double depth;

private double front;

private double back;

public Plot(String id,Plottype type,Shapetype shape,double width,double depth,double front,double back){

this.id=id;

this.type=type;

this.shape=shape;

this.width=width;

this.depth=depth;

this.front=front;

this.back=back;

this.isavailable=true;

this.price=type.getprice();

this.area=calarea();

}

public void setprice(double price){

this.price=price;

}

public double calarea(){

switch(shape){

case RECTANGLE:

return area=width\*depth;

case TRAPEZOID:

return area = ((front + back) / 2) \* depth;

case L\_SHAPE:

return area = (width \* depth) + (front \* back);

default:

return 0;

}

}

//Getters

public boolean isavailable(){

return isavailable;

}

public double getarea(){

return area;

}

public double getprice(){

return price;

}

public String getid(){

return id;

}

public Plottype gettype(){

return type;

}

//Methods

public boolean book(){

if(isavailable){

isavailable=false;

return true;

}

else

return false;

}

public boolean cancel(){

if(!isavailable){

isavailable=true;

return true;

}

else

return false;

}

public double getwidth(){

return width;

}

public double getdepth(){

return depth;

}

public double getfront(){

return front;

}

public double getback(){

return back;

}

//toString

@Override

public String toString(){

String status=isavailable?"Available":"Booked";

return String.format("Id: %s |Type: %s |Shape: %s |Area: %.2f |Price: %.2f |Status: %s",id,type,shape,this.area,this.price,status);

}

}public enum Plottype{

RES\_5\_MARLA(4000000),

RES\_10\_MARLA(7500000),

RES\_1\_KANAL(14000000),

COMM\_SHOP(3000000),

COMM\_OFFICE(5000000),

PARKING(200000);

private final double price;

Plottype(double price){

this.price=price;

}

public double getprice(){

return price;

}

}public enum Shapetype{

RECTANGLE,TRAPEZOID,L\_SHAPE;

}public class Shop{

private String id;

private double price;

private Shapetype shape;

private Plottype type;

private boolean available;

Shop(String id,Shapetype shape,Plottype type){

this.id=id;

this.price=price;

this.available=true;

this.shape=shape;

this.type=type;

this.price=type.getprice();

}

public boolean book(){

if(available){

available=false;

return true;

}

else

return false;

}

public boolean cancel(){

if(!available){

available=true;

return true;

}

else

return false;

}

public boolean getavailable(){

return available;

}

public String toString(){

String status=available?"available":"booked";

return String.format("Shop %s | Status: %s | Price: %.2f |Shape: %s |Shoptype: %s",id,status,this.price,shape,type);

//return id + " | " + " | " +status+"Price: "+this.price;

}

}

**OUTPUTS:**

