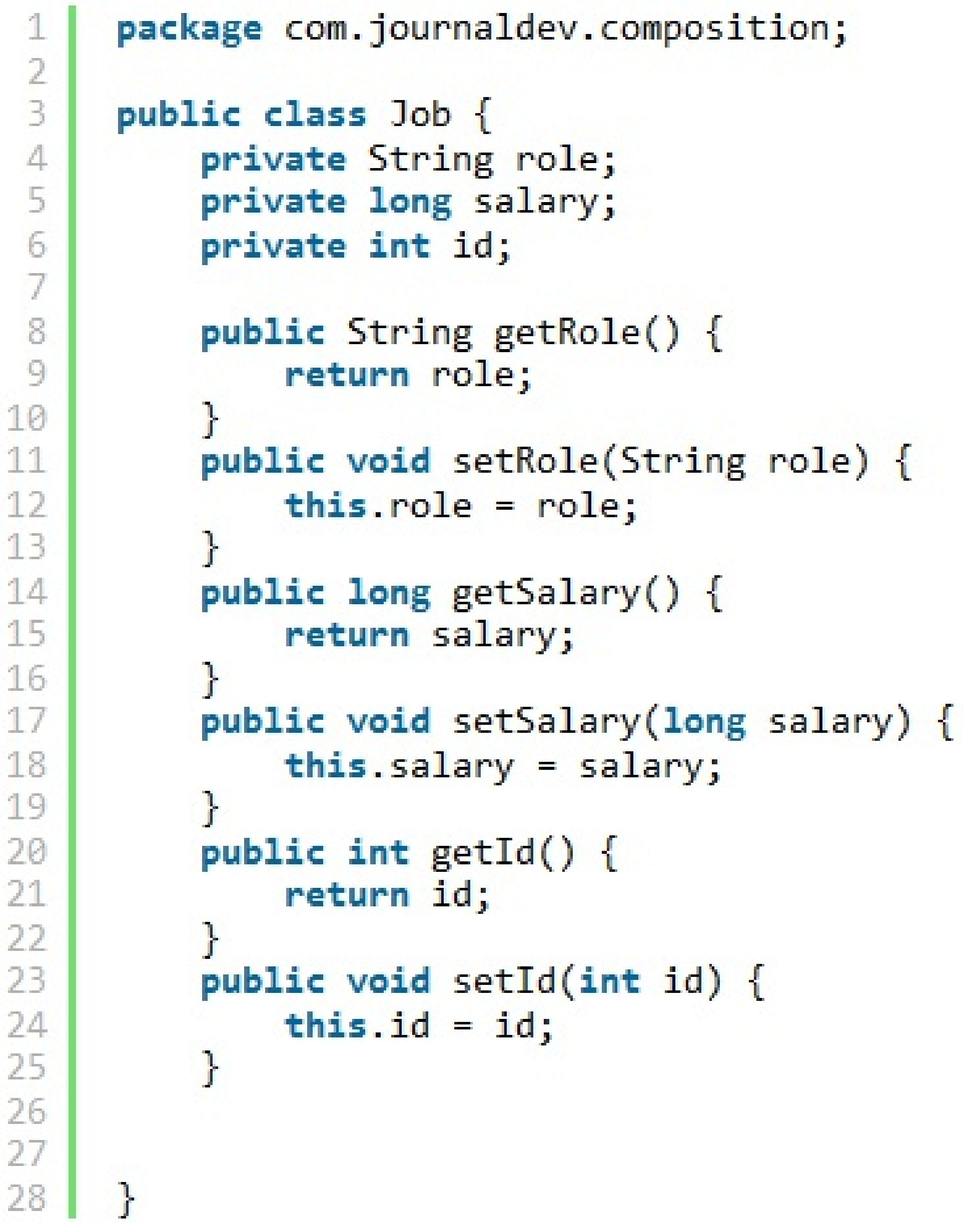
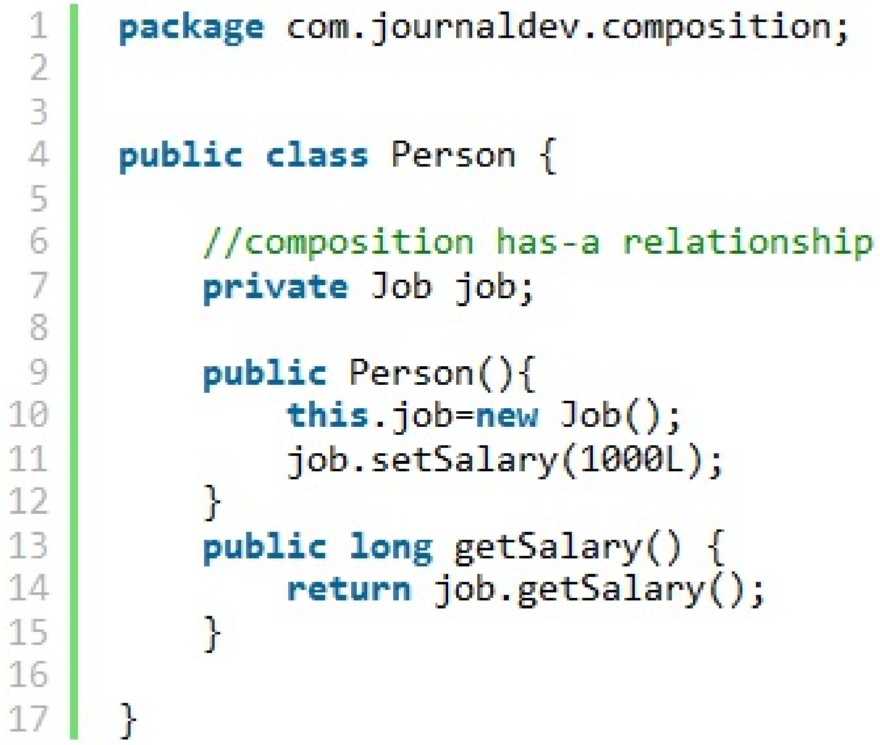
Job.java



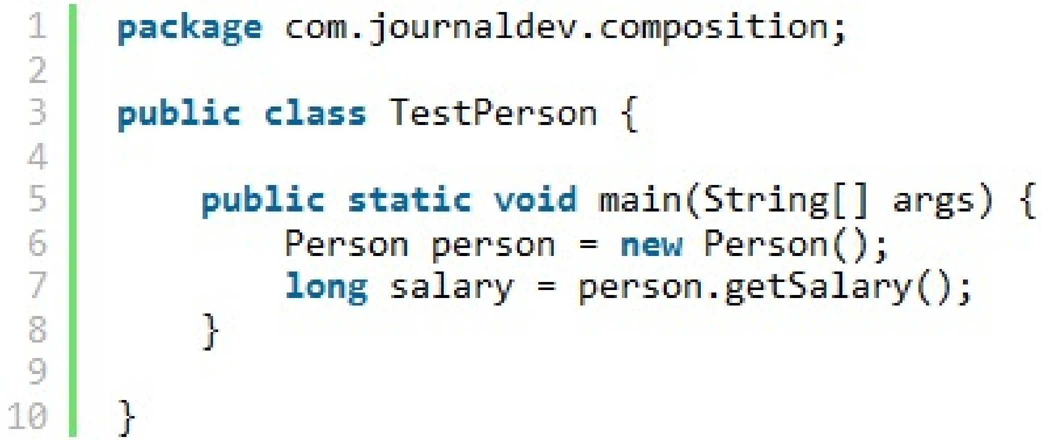
```java
package com.journaldev.composition;

public class Job {
    private String role;
    private long salary;
    private int id;

    public String getRole() {
        return role;
    }
    public void setRole(String role) {
        this.role = role;
    }
    public long getSalary() {
        return salary;
    }
    public void setSalary(long salary) {
        this.salary = salary;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }


}
```

Person.java

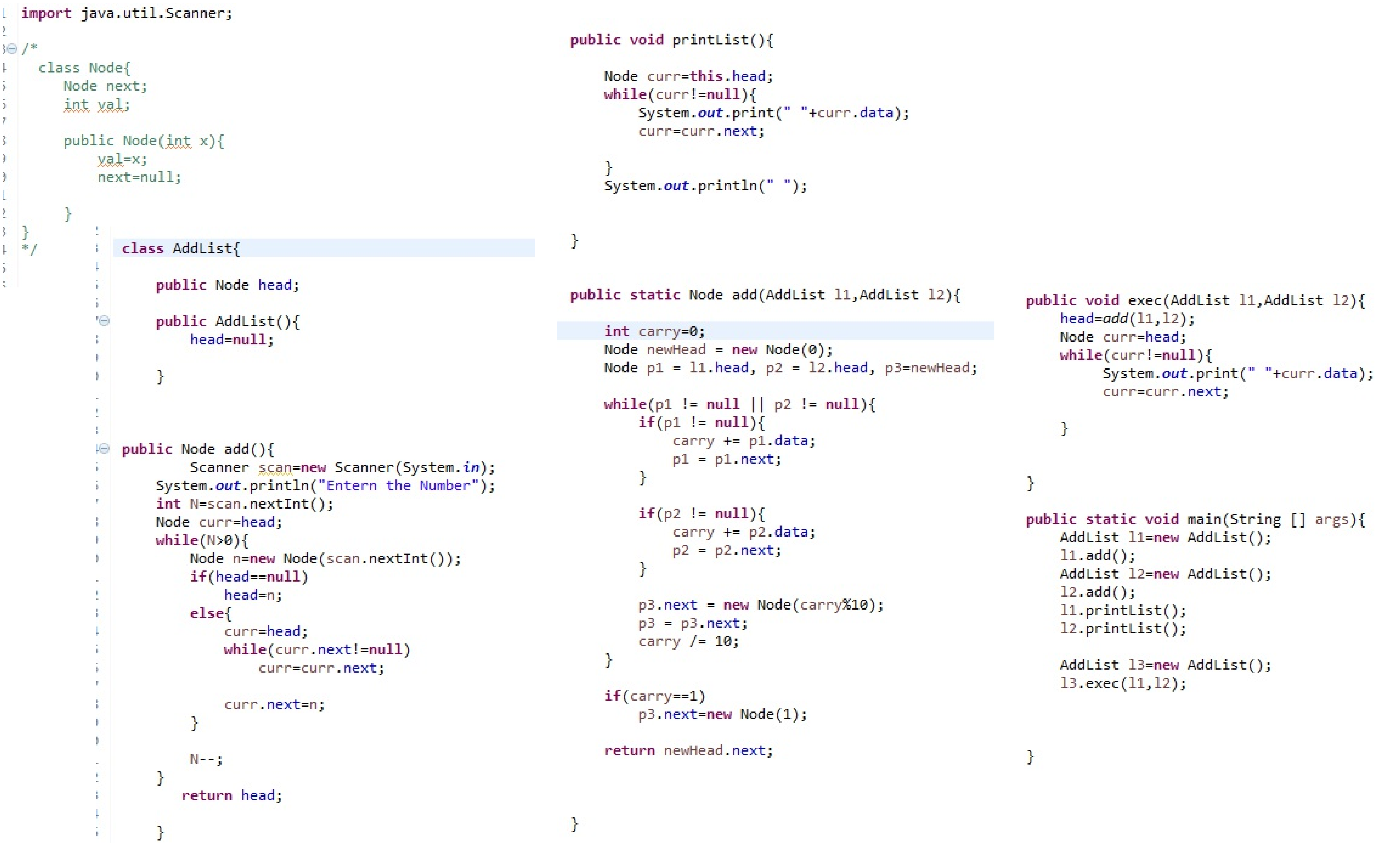


```java
package com.journaldev.composition;


public class Person {

    //composition has-a relationship
    private Job job;

    public Person(){
        this.job=new Job();
        job.setSalary(1000L);
    }
    public long getSalary() {
        return job.getSalary();
    }

}
```

TestPerson.java



```java
package com.journaldev.composition;

public class TestPerson {

    public static void main(String[] args) {
        Person person = new Person();
        long salary = person.getSalary();
    }

}
```

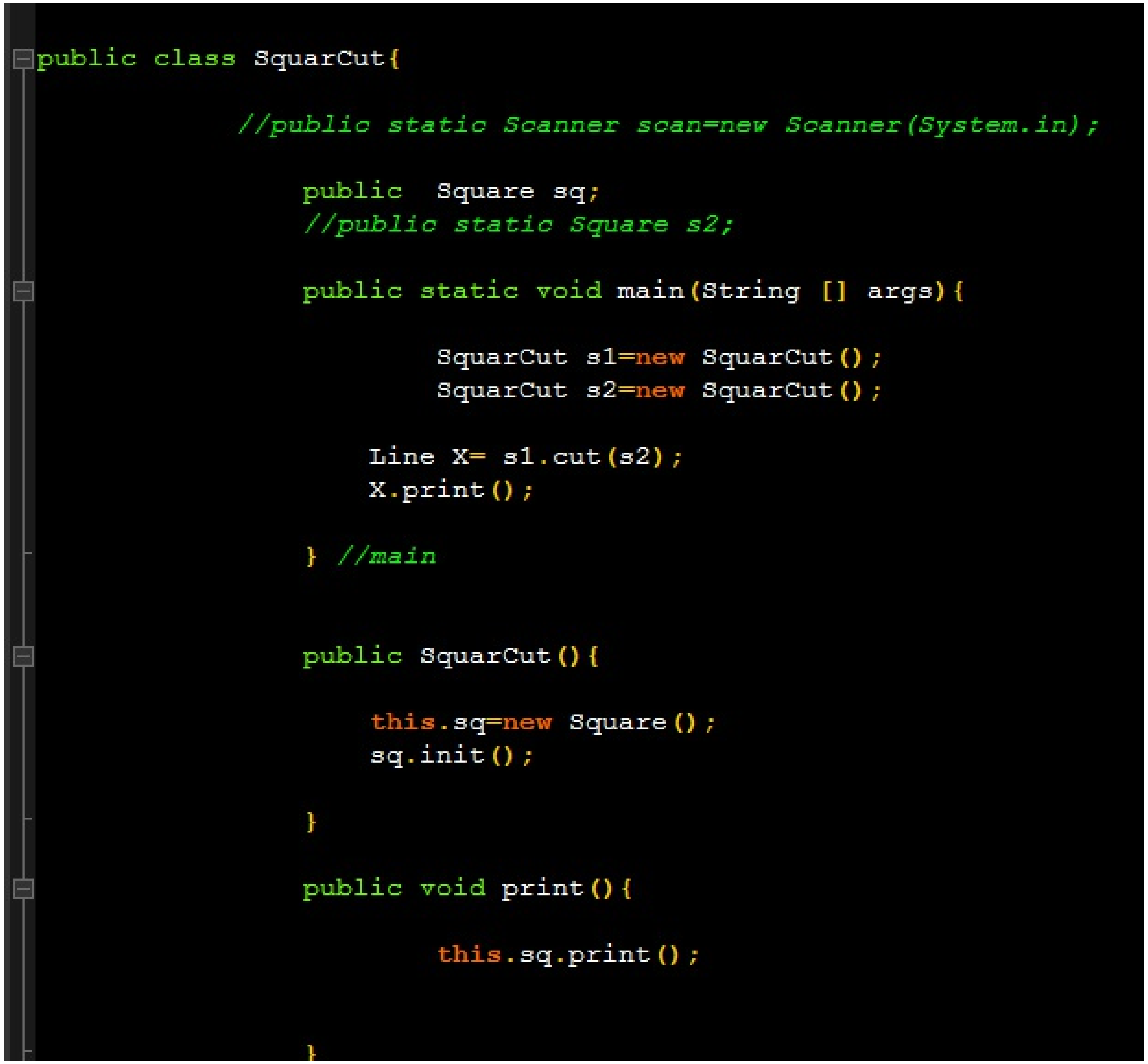


```
import java.util.Scanner;

/*
  class Node{
     Node next;
     int val;

     public Node(int x){
         val=x;
         next=null;

     }
 }
*/
```

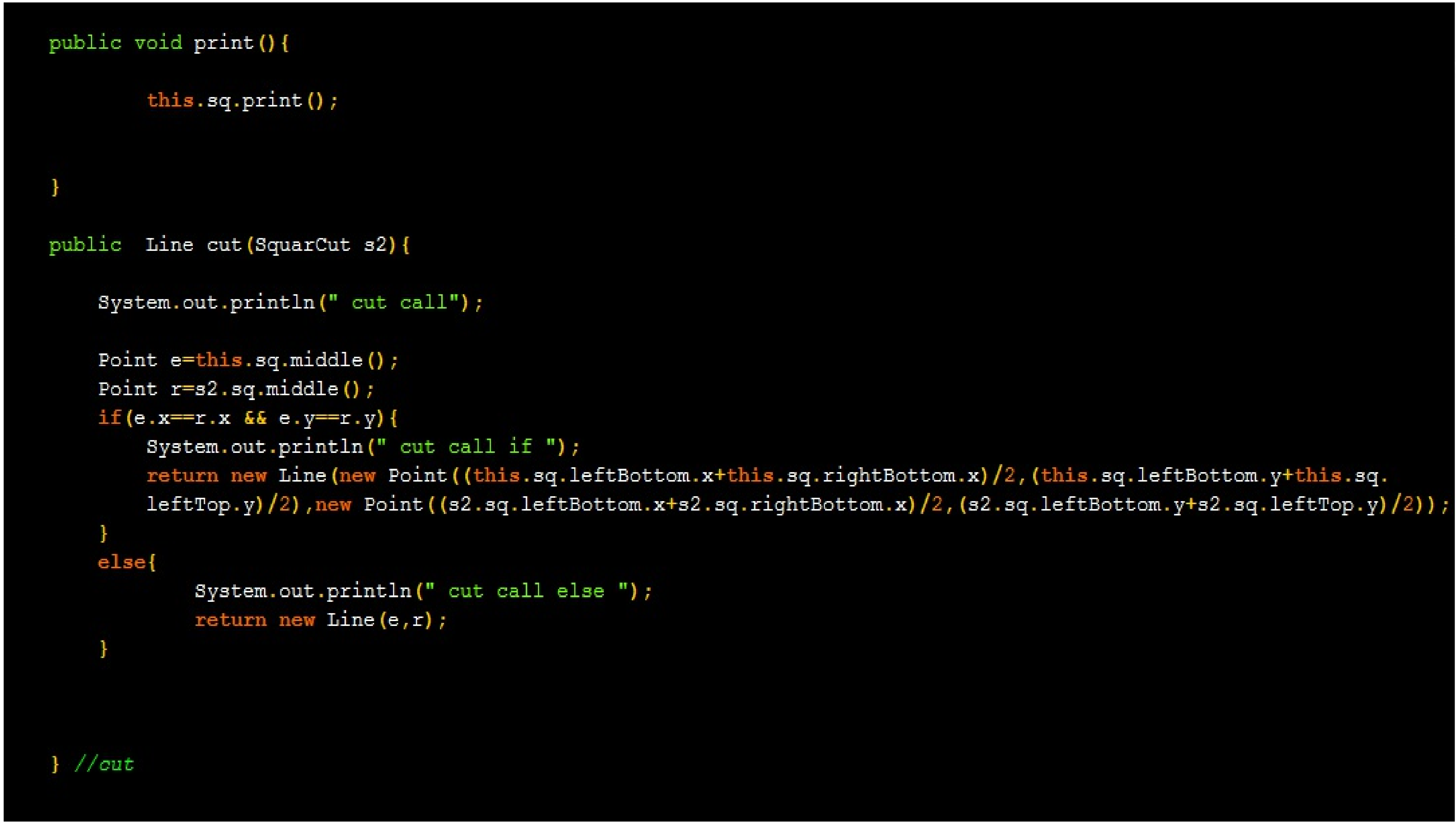
```
class AddList{

    public Node head;

    public AddList(){
        head=null;

    }


public Node add(){
        Scanner scan=new Scanner(System.in);
    System.out.println("Entern the Number");
    int N=scan.nextInt();
    Node curr=head;
    while(N>0){
        Node n=new Node(scan.nextInt());
        if(head==null)
            head=n;
        else{
            curr=head;
            while(curr.next!=null)
                curr=curr.next;

            curr.next=n;
        }

        N--;
    }
      return head;

    }
```

```
public void printList(){

    Node curr=this.head;
    while(curr!=null){
        System.out.print(" "+curr.data);
        curr=curr.next;

    }
    System.out.println(" ");


}


public static Node add(AddList l1,AddList l2){

    int carry=0;
    Node newHead = new Node(0);
    Node p1 = l1.head, p2 = l2.head, p3=newHead;

    while(p1 != null || p2 != null){
        if(p1 != null){
            carry += p1.data;
            p1 = p1.next;
        }

        if(p2 != null){
            carry += p2.data;
            p2 = p2.next;
        }

        p3.next = new Node(carry%10);
        p3 = p3.next;
        carry /= 10;
    }

    if(carry==1)
        p3.next=new Node(1);

    return newHead.next;



}
```

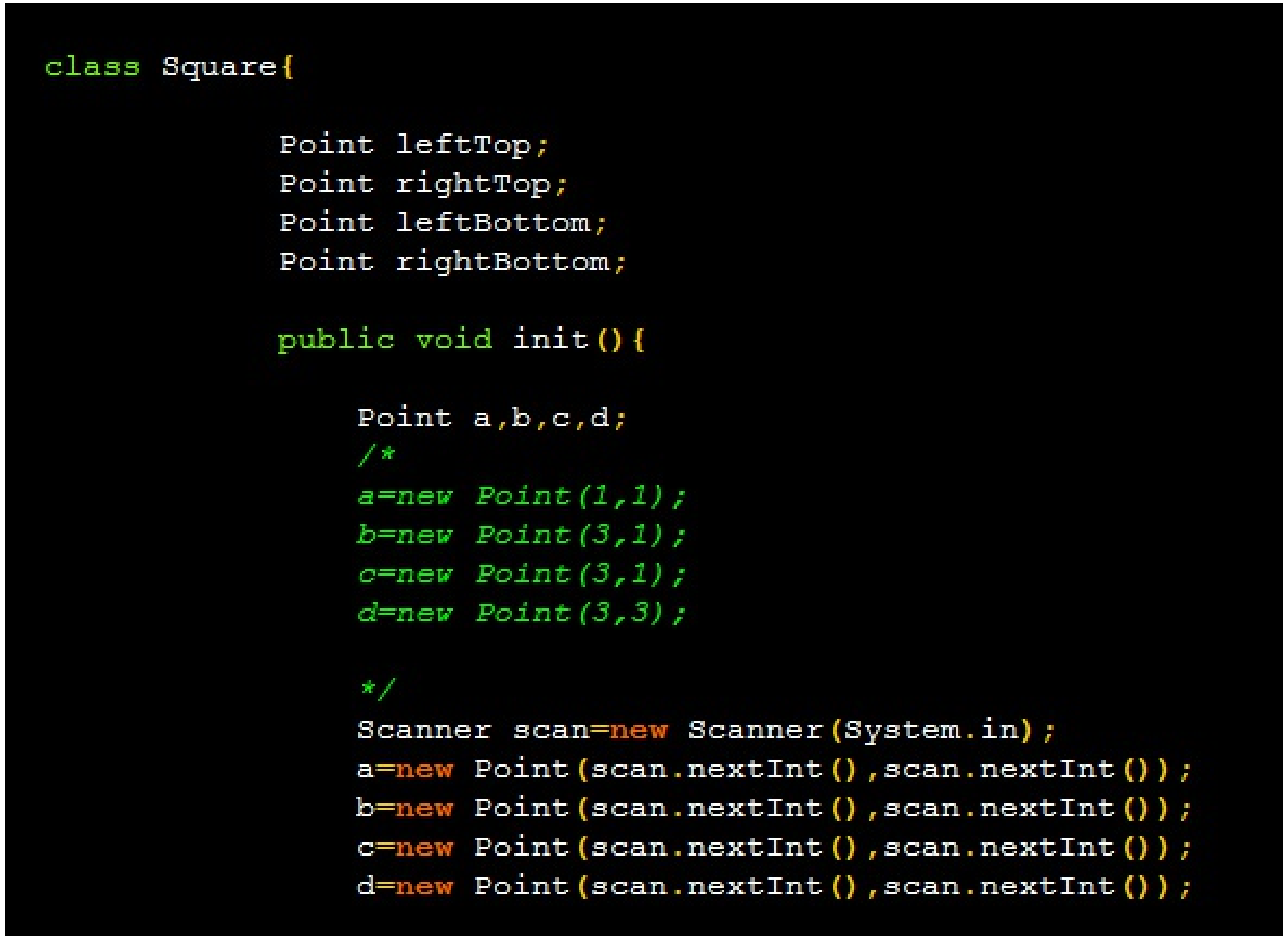
```
public void exec(AddList l1,AddList l2){
    head=add(l1,l2);
    Node curr=head;
    while(curr!=null){
         System.out.print(" "+curr.data);
         curr=curr.next;

    }


}

public static void main(String [] args){
    AddList l1=new AddList();
    l1.add();
    AddList l2=new AddList();
    l2.add();
    l1.printList();
    l2.printList();

    AddList l3=new AddList();
    l3.exec(l1,l2);



}
```



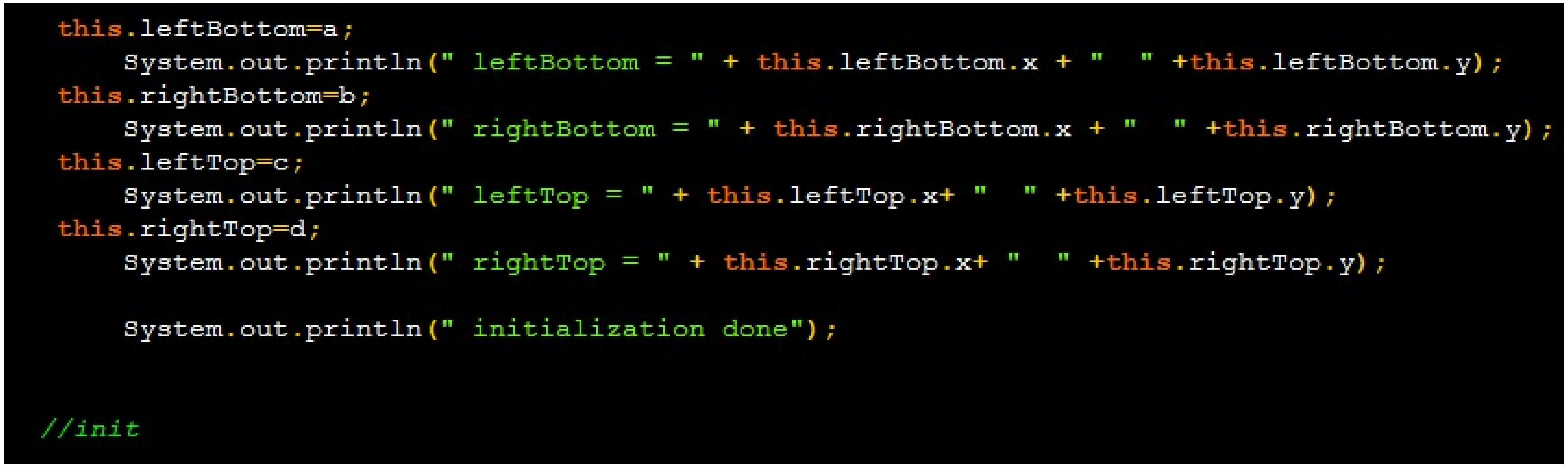
```
public class SquarCut{

            //public static Scanner scan=new Scanner(System.in);

                public  Square sq;
                //public static Square s2;

                public static void main(String [] args){

                        SquarCut s1=new SquarCut();
                        SquarCut s2=new SquarCut();

                    Line X= s1.cut(s2);
                    X.print();

                } //main


                public SquarCut(){

                    this.sq=new Square();
                    sq.init();

                }

                public void print(){

                        this.sq.print();


                }
```



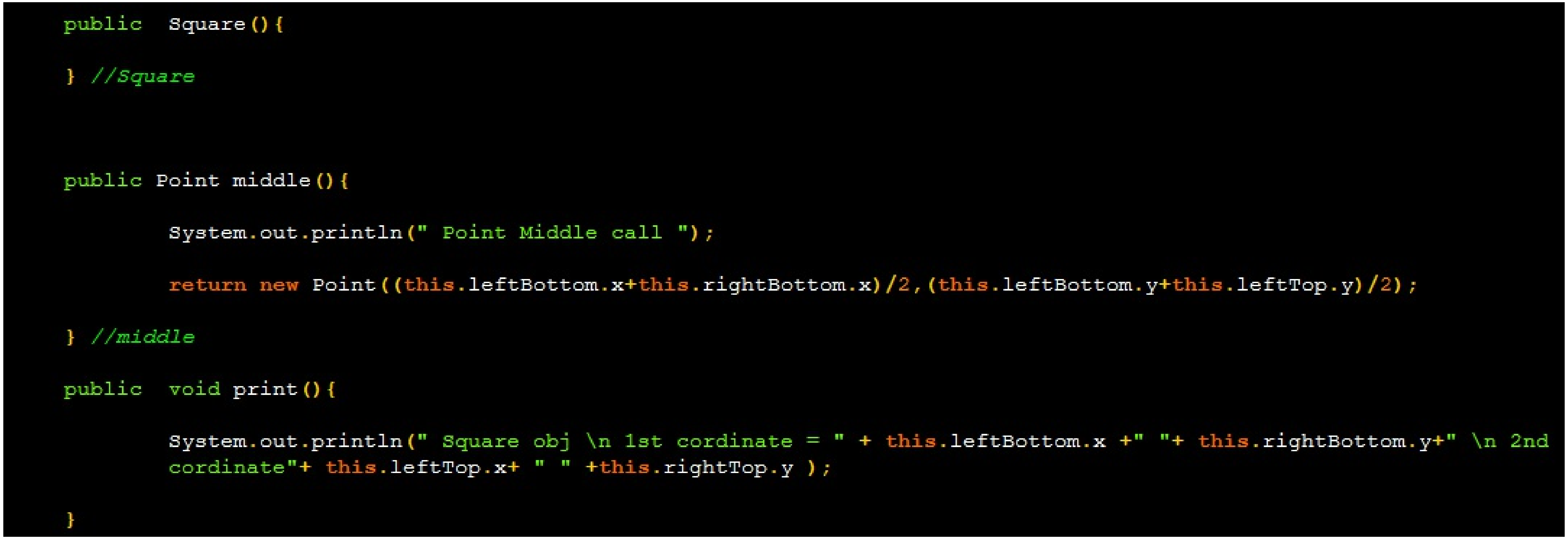
```
public void print(){

        this.sq.print();


}

public  Line cut(SquarCut s2){

    System.out.println(" cut call");

    Point e=this.sq.middle();
    Point r=s2.sq.middle();
    if(e.x==r.x && e.y==r.y){
        System.out.println(" cut call if ");
        return new Line(new Point((this.sq.leftBottom.x+this.sq.rightBottom.x)/2,(this.sq.leftBottom.y+this.sq.
        leftTop.y)/2),new Point((s2.sq.leftBottom.x+s2.sq.rightBottom.x)/2,(s2.sq.leftBottom.y+s2.sq.leftTop.y)/2));
    }
    else{
            System.out.println(" cut call else ");
            return new Line(e,r);
    }



} //cut
```



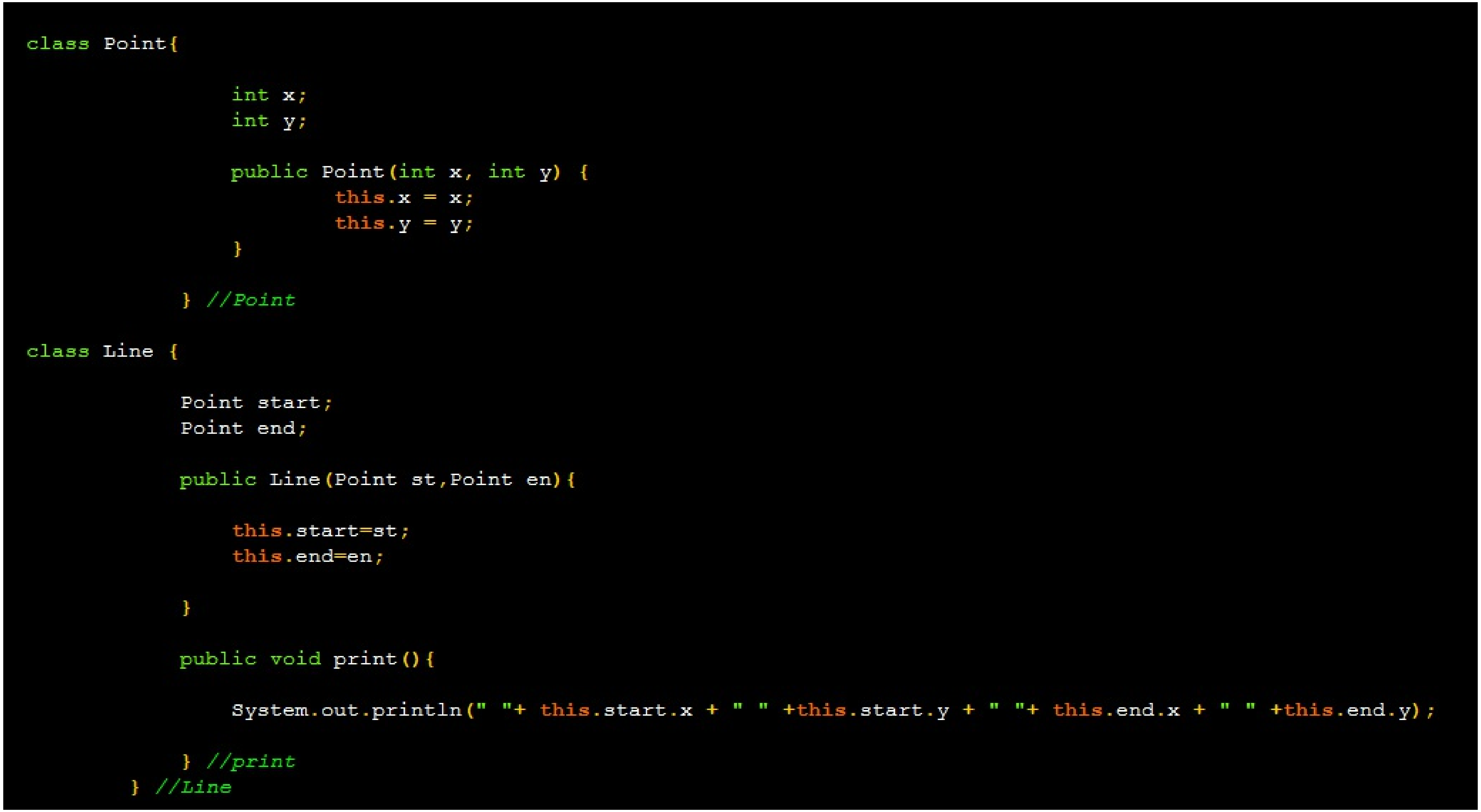
```java
class Square{

        Point leftTop;
        Point rightTop;
        Point leftBottom;
        Point rightBottom;

        public void init(){

            Point a,b,c,d;
            /*
            a=new Point(1,1);
            b=new Point(3,1);
            c=new Point(3,1);
            d=new Point(3,3);

            */
            Scanner scan=new Scanner(System.in);
            a=new Point(scan.nextInt(),scan.nextInt());
            b=new Point(scan.nextInt(),scan.nextInt());
            c=new Point(scan.nextInt(),scan.nextInt());
            d=new Point(scan.nextInt(),scan.nextInt());
```



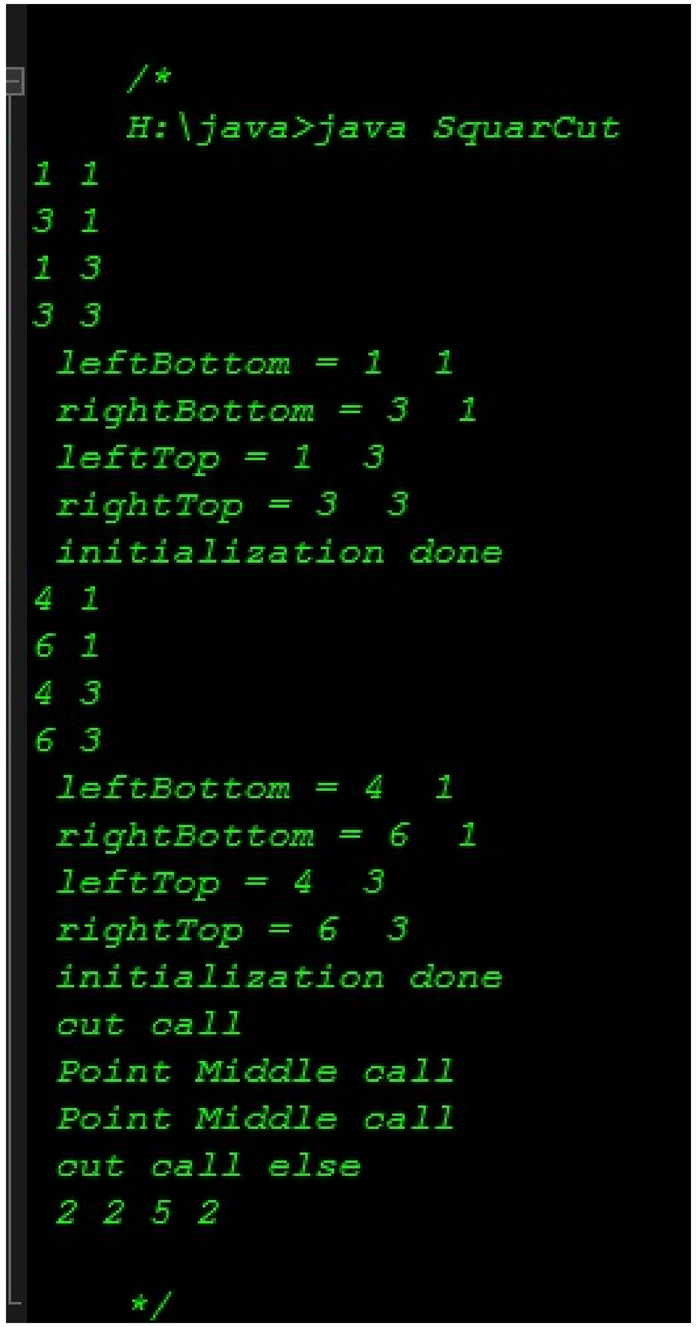
```java
   this.leftBottom=a;
       System.out.println(" leftBottom = " + this.leftBottom.x + "   " +this.leftBottom.y);
   this.rightBottom=b;
       System.out.println(" rightBottom = " + this.rightBottom.x + "   " +this.rightBottom.y);
   this.leftTop=c;
       System.out.println(" leftTop = " + this.leftTop.x+ "   " +this.leftTop.y);
   this.rightTop=d;
       System.out.println(" rightTop = " + this.rightTop.x+ "   " +this.rightTop.y);

       System.out.println(" initialization done");


  //init
```



```
    public  Square(){

    } //Square



    public Point middle(){

            System.out.println(" Point Middle call ");

            return new Point((this.leftBottom.x+this.rightBottom.x)/2,(this.leftBottom.y+this.leftTop.y)/2);

    } //middle

    public  void print(){

            System.out.println(" Square obj \n 1st cordinate = " + this.leftBottom.x +" "+ this.rightBottom.y+" \n 2nd
            cordinate"+ this.leftTop.x+ " " +this.rightTop.y );

    }
```



```
class Point{

                int x;
                int y;

                public Point(int x, int y) {
                        this.x = x;
                        this.y = y;
                }

            } //Point

class Line {

            Point start;
            Point end;

            public Line(Point st,Point en){

                this.start=st;
                this.end=en;

            }

            public void print(){

                System.out.println(" "+ this.start.x + " " +this.start.y + " "+ this.end.x + " " +this.end.y);

            } //print
        } //Line
```



```
    /*
    H:\java>java SquarCut
1 1
3 1
1 3
3 3
 leftBottom = 1  1
 rightBottom = 3  1
 leftTop = 1  3
 rightTop = 3  3
 initialization done
4 1
6 1
4 3
6 3
 leftBottom = 4  1
 rightBottom = 6  1
 leftTop = 4  3
 rightTop = 6  3
 initialization done
 cut call
 Point Middle call
 Point Middle call
 cut call else
 2 2 5 2

    */
```