ANDROID CIRCLE

LEVEL : JAVA AHMED RADWAN

WEEK 2

1. What are access modifiers ? and Types of Access Modifiers.

- Access Modifiers in java are special keywords by them we can determine the accessibility of an class, method or an attribute.

- Types of Access Modifiers : Public , Private , Protected , Default.

- Public : It can be accessed (class , method or attribute) from everywhere (class or package).

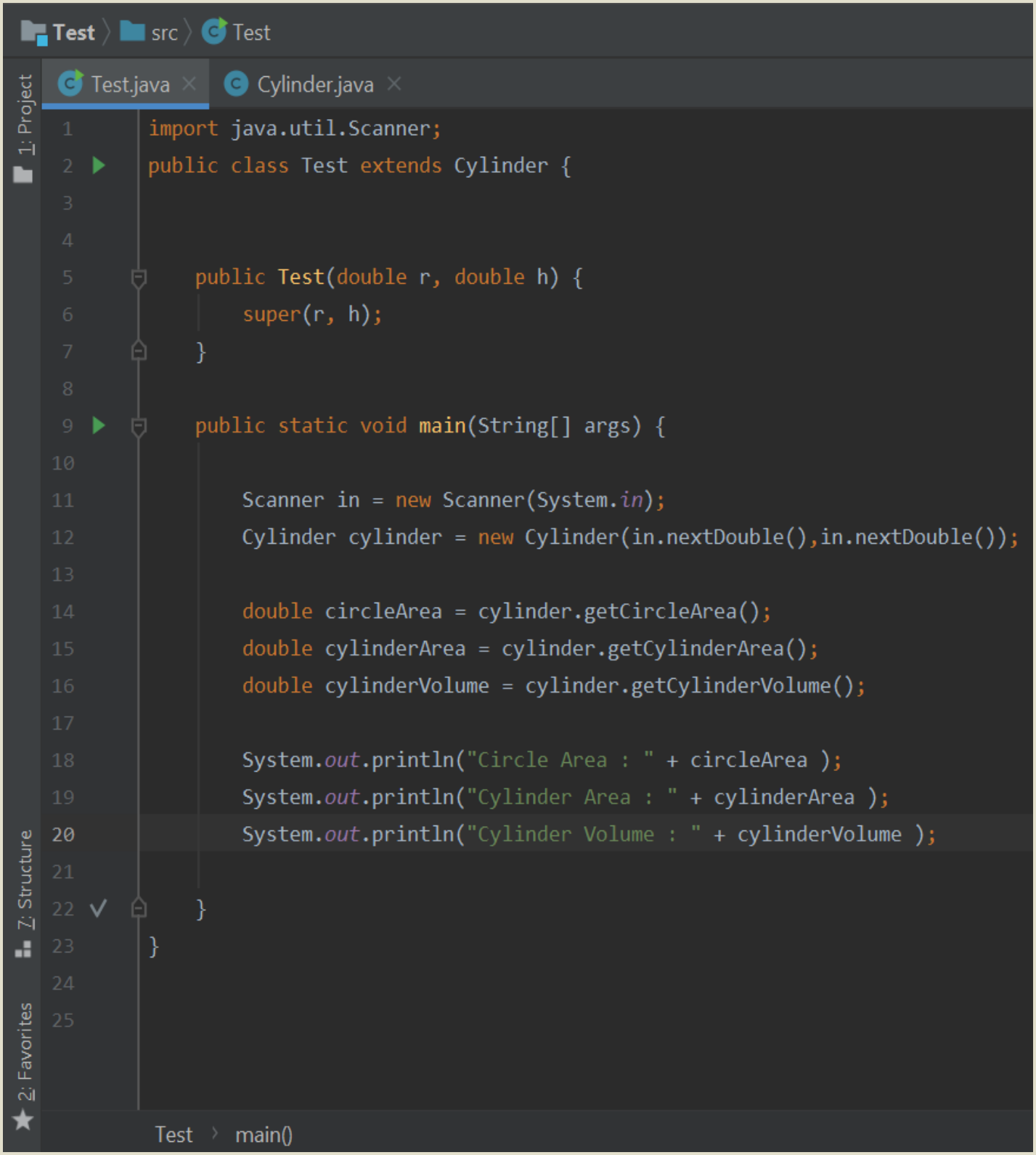
- **Private : It can only be accessed (class , method or attribute) within its class.**

- **Protected : : It can be accessed (class , method or attribute) within its package or another package using inheritance (child class).**

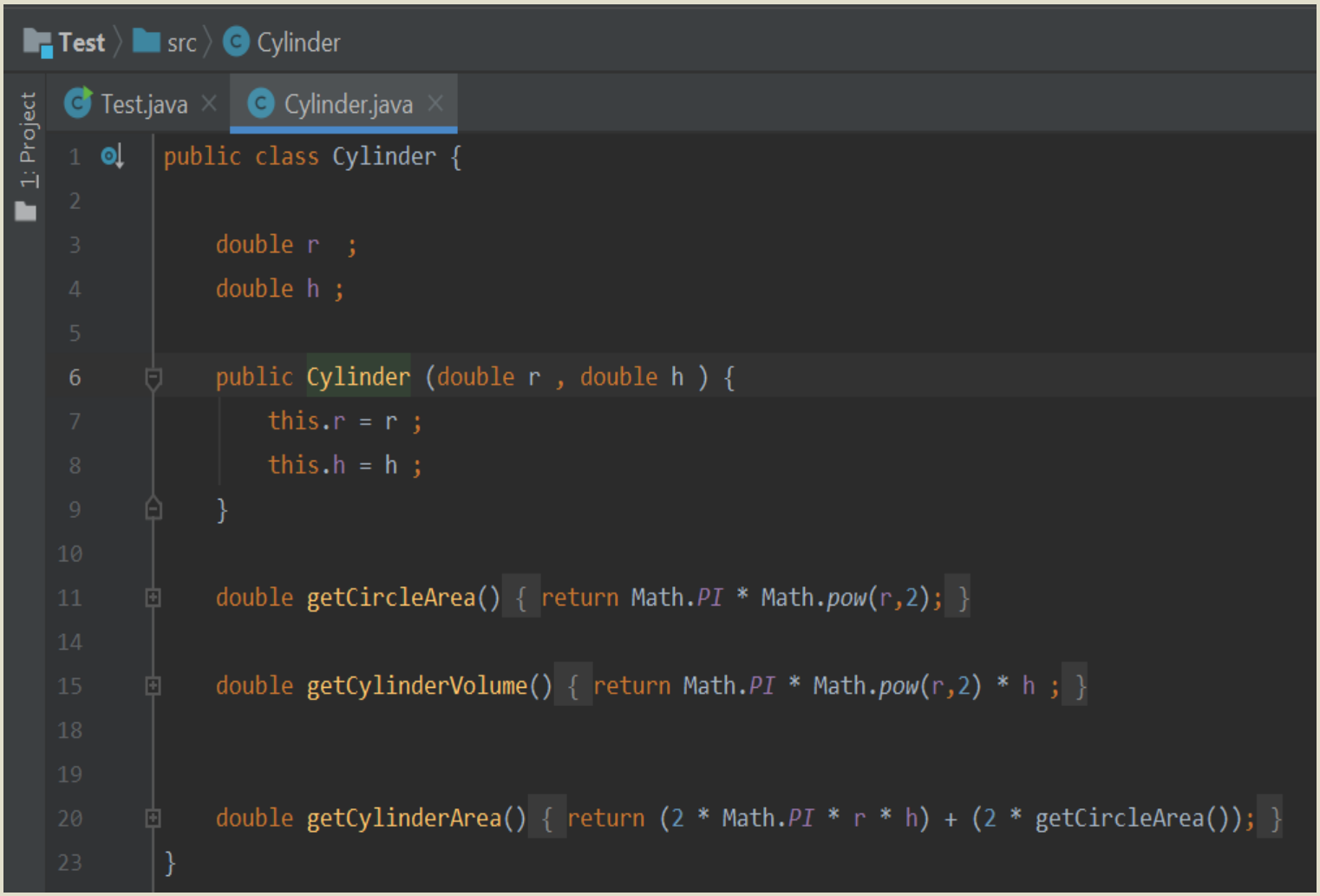
- **Default : It is the default access modifier that given to the class , method or attribute if you did not give it any access modifiers, It can only be accessed (class , method or attribute) within its package.**

---

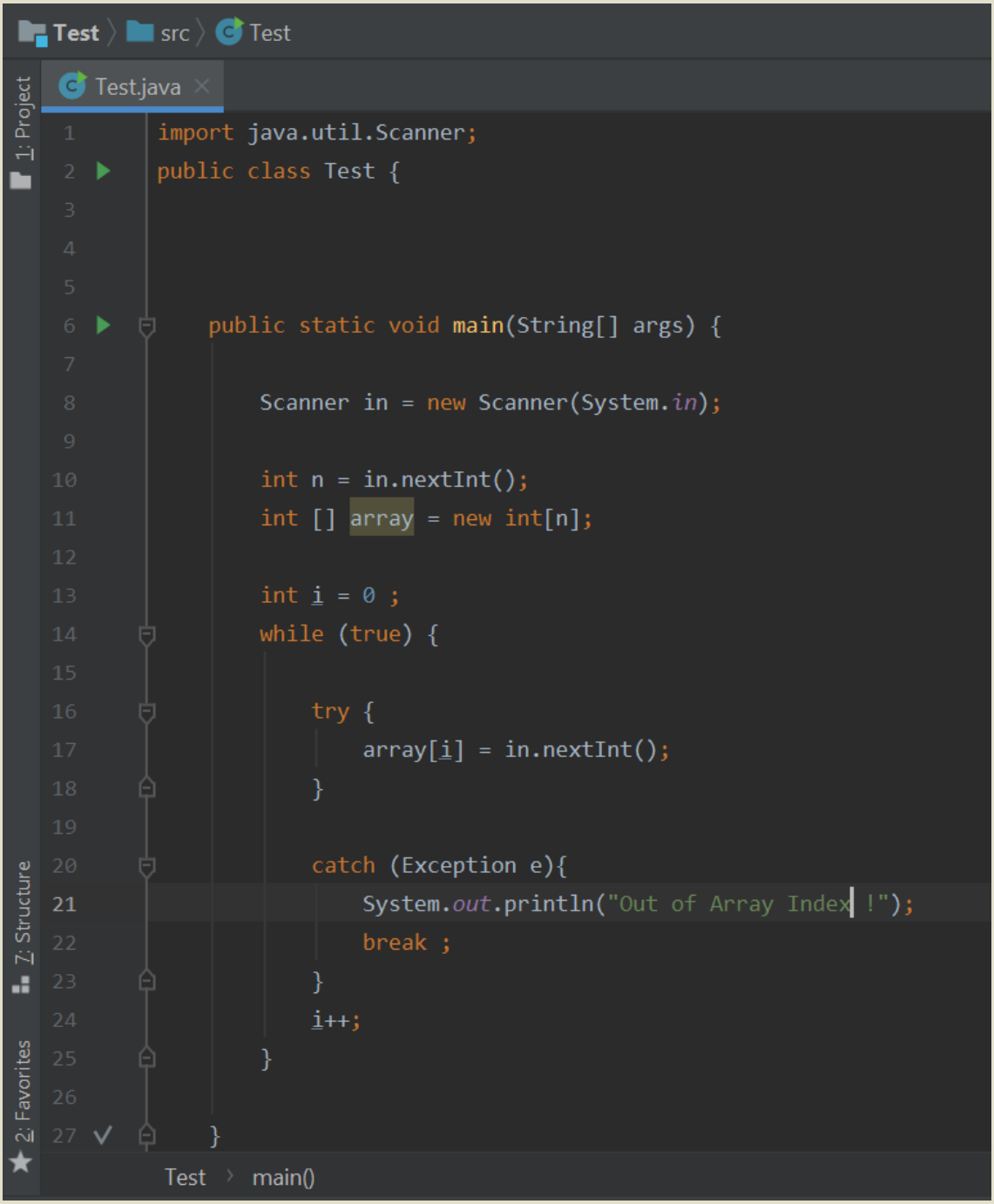
2. Program code




```java
import java.util.Scanner;
public class Test extends Cylinder {


    public Test(double r, double h) {
        super(r, h);
    }

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);
        Cylinder cylinder = new Cylinder(in.nextDouble(),in.nextDouble());

        double circleArea = cylinder.getCircleArea();
        double cylinderArea = cylinder.getCylinderArea();
        double cylinderVolume = cylinder.getCylinderVolume();

        System.out.println("Circle Area : " + circleArea );
        System.out.println("Cylinder Area : " + cylinderArea );
        System.out.println("Cylinder Volume : " + cylinderVolume );

    }
}
```


Test
src
Cylinder
1: Project
Test.java
Cylinder.java
public class Cylinder {
double r ;
double h ;
public Cylinder (double r , double h) {
this.r = r ;
this.h = h ;
}
double getCircleArea() { return Math.PI * Math.pow(r,2); }
double getCylinderVolume() { return Math.PI * Math.pow(r,2) * h ; }
double getCylinderArea() { return (2 * Math.PI * r * h) + (2 * getCircleArea()); }
}

3.Program Code




```java
import java.util.Scanner;
public class Test {



    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        int n = in.nextInt();
        int [] array = new int[n];

        int i = 0 ;
        while (true) {

            try {
                array[i] = in.nextInt();
            }

            catch (Exception e){
                System.out.println("Out of Array Index !");
                break ;
            }
            i++;
        }

    }
```