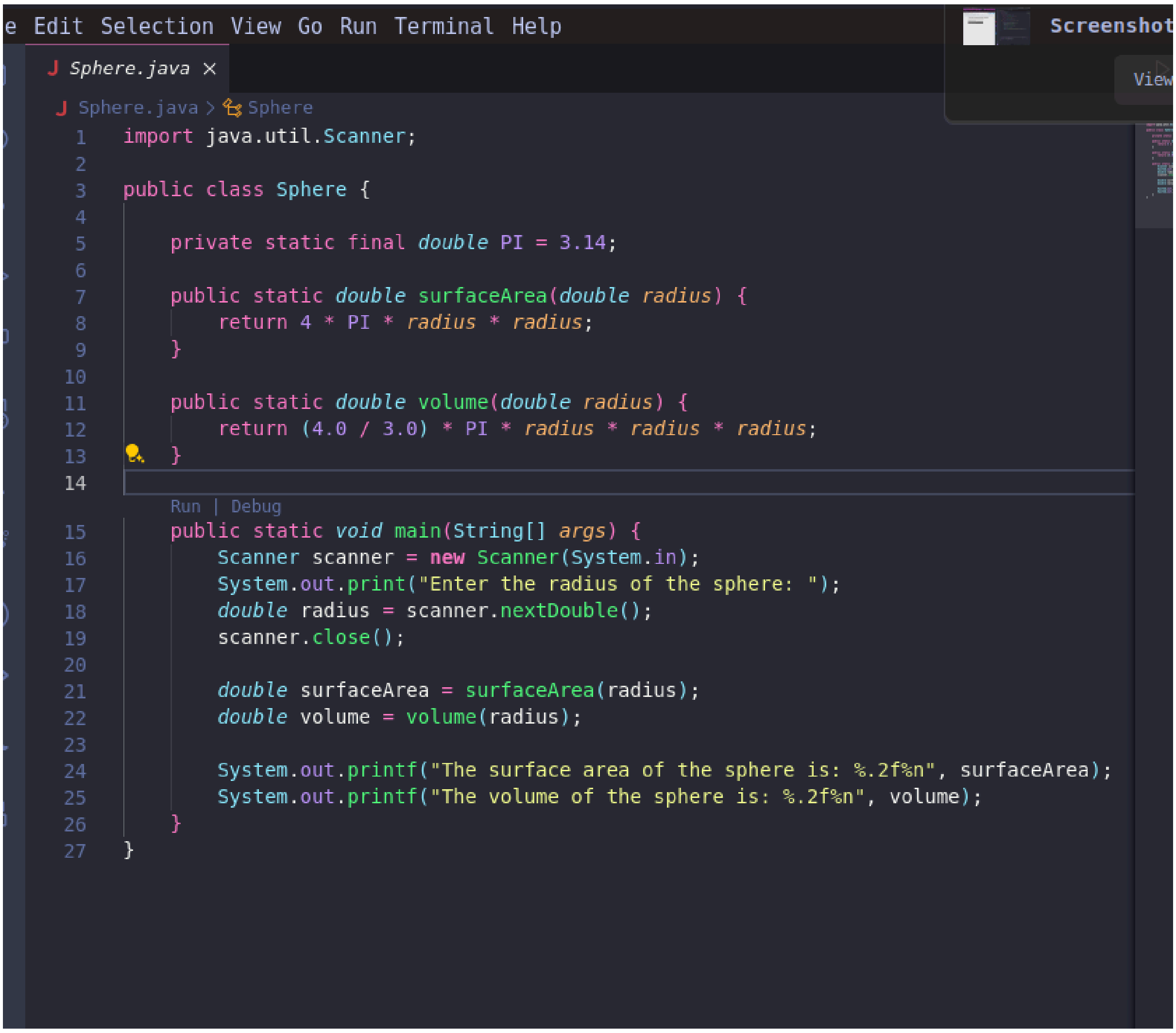
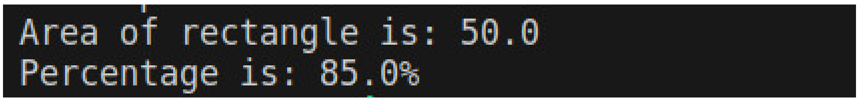
JAVA ASSIGNMENT _22610018_VighneshPotdar_S2Batch

1. Write a program to calculate area and volume of sphere using static variable and method create two static methods for area and volume calculation. (insert data from user).

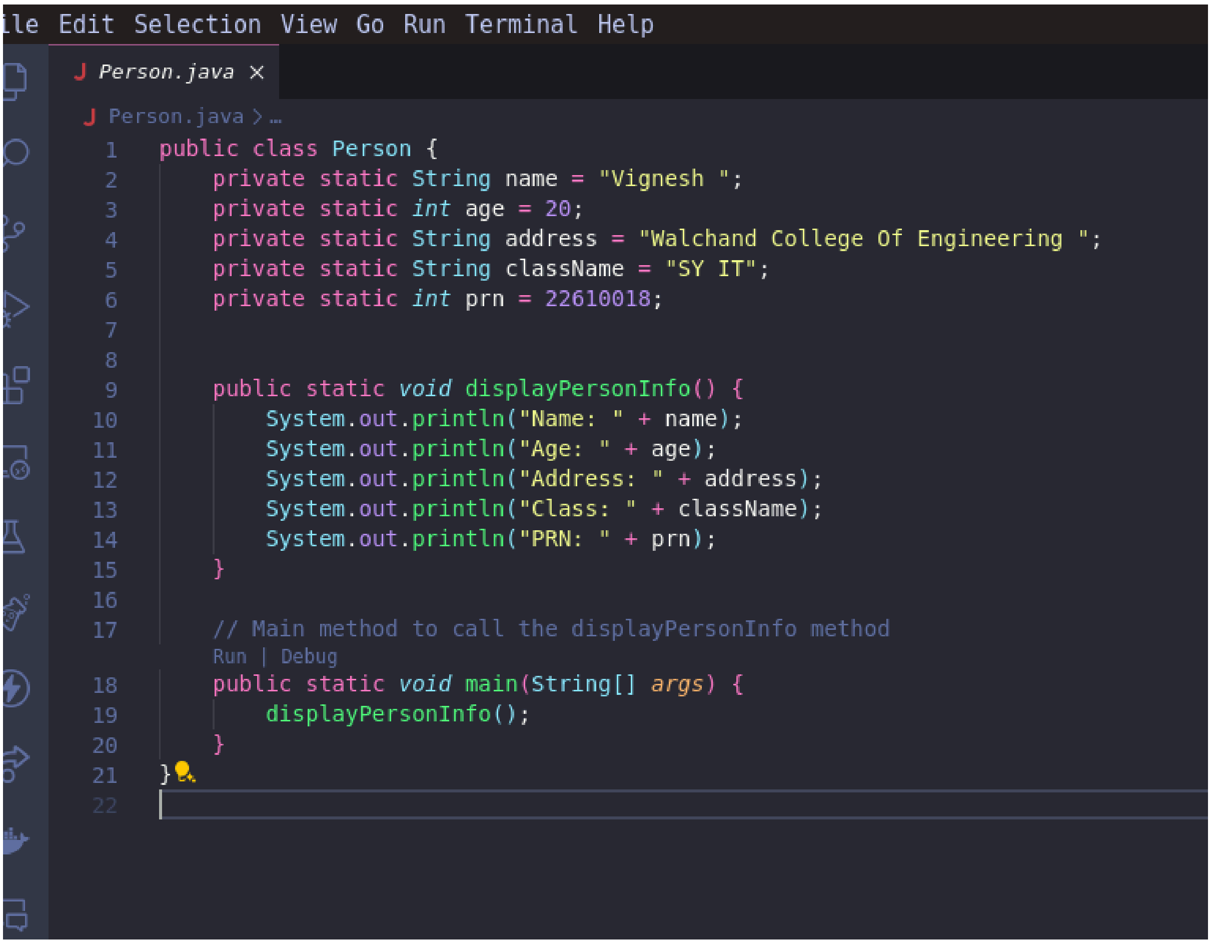


```java
import java.util.Scanner;

public class Sphere {

    private static final double PI = 3.14;

    public static double surfaceArea(double radius) {
        return 4 * PI * radius * radius;
    }

    public static double volume(double radius) {
        return (4.0 / 3.0) * PI * radius * radius * radius;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the radius of the sphere: ");
        double radius = scanner.nextDouble();
        scanner.close();

        double surfaceArea = surfaceArea(radius);
        double volume = volume(radius);

        System.out.printf("The surface area of the sphere is: %.2f%n", surfaceArea);
        System.out.printf("The volume of the sphere is: %.2f%n", volume);
    }
}
```

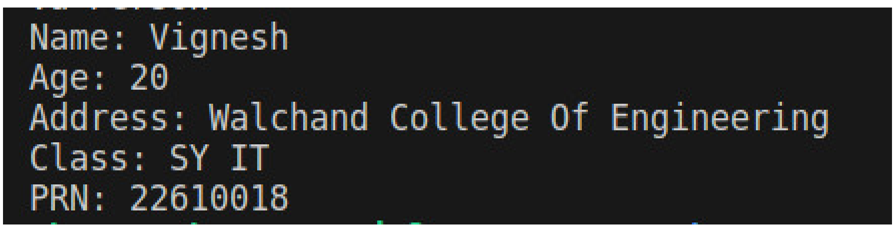


```
Area of rectangle is: 50.0
Percentage is: 85.0%
```

2. Display all your information (prn, name, age, address, class) on console without creating any object and writing any code in main method.

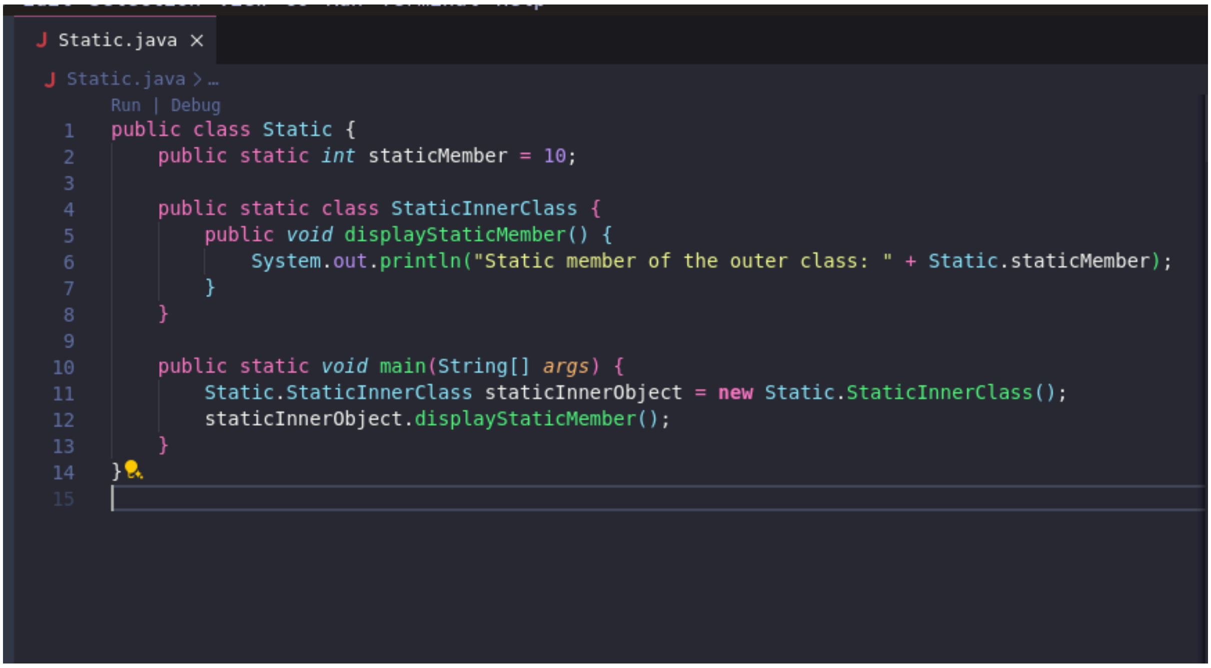


```java
public class Person {
    private static String name = "Vignesh ";
    private static int age = 20;
    private static String address = "Walchand College Of Engineering ";
    private static String className = "SY IT";
    private static int prn = 22610018;


    public static void displayPersonInfo() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Address: " + address);
        System.out.println("Class: " + className);
        System.out.println("PRN: " + prn);
    }

    // Main method to call the displayPersonInfo method
    public static void main(String[] args) {
        displayPersonInfo();
    }
}
```

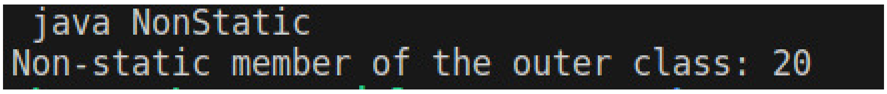


```
Name: Vignesh
Age: 20
Address: Walchand College Of Engineering
Class: SY IT
PRN: 22610018
```

3. Demonstrate how to use static inner class and non-static inner class to access static and non-static members of outer class.

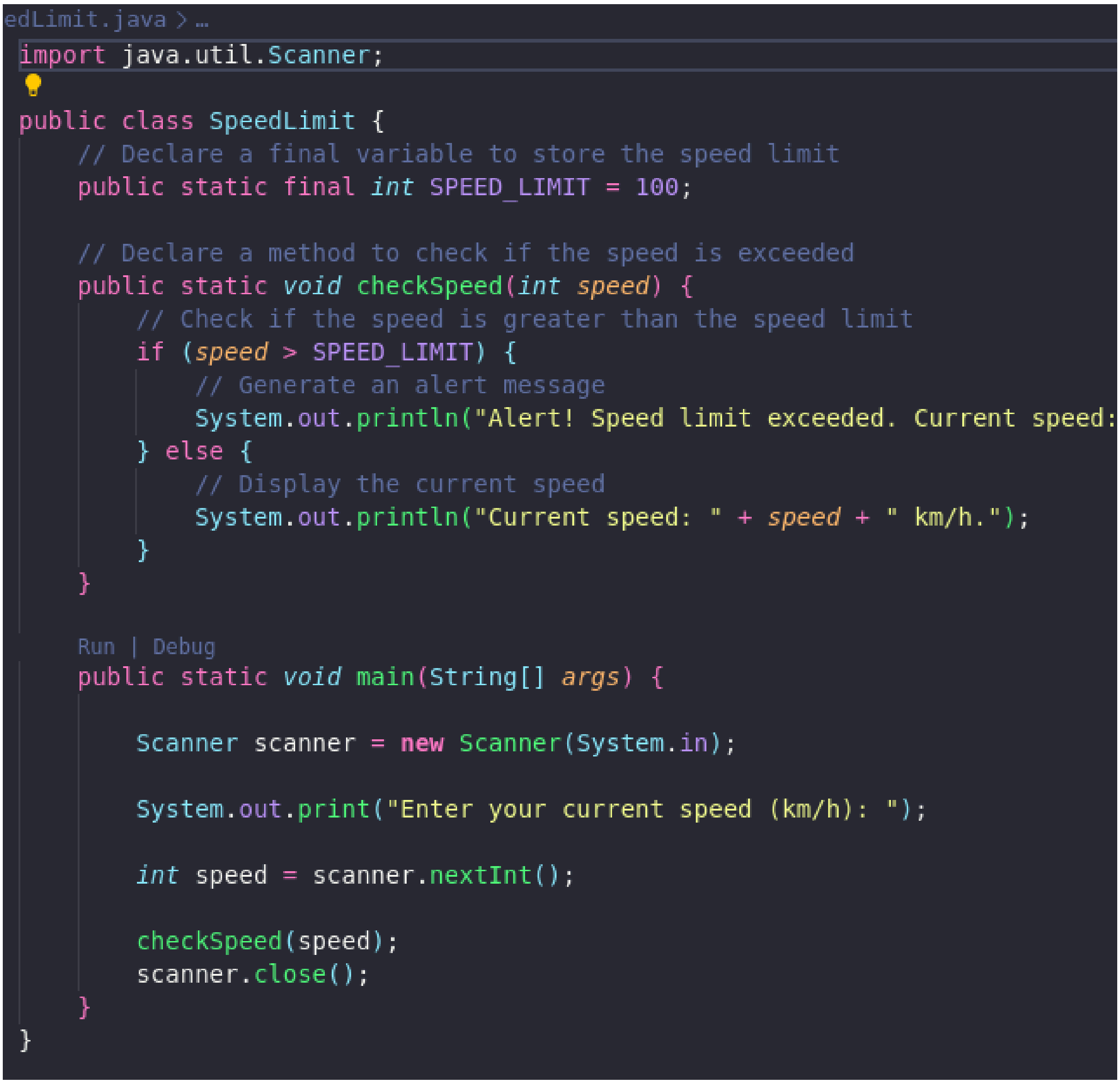


```java
public class Static {
    public static int staticMember = 10;

    public static class StaticInnerClass {
        public void displayStaticMember() {
            System.out.println("Static member of the outer class: " + Static.staticMember);
        }
    }

    public static void main(String[] args) {
        Static.StaticInnerClass staticInnerObject = new Static.StaticInnerClass();
        staticInnerObject.displayStaticMember();
    }
}
```

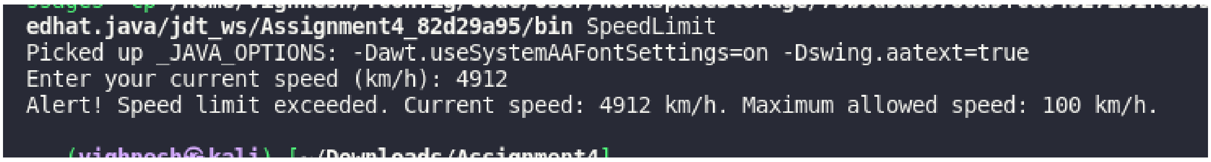


```
 java NonStatic
Non-static member of the outer class: 20
```

4. Write a program using final variable to check speed limit exceeds or not on highway. If speed is greater than 100. Then generate alert message.



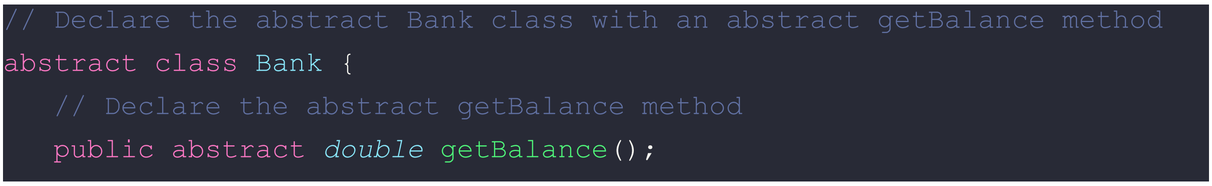
```java
import java.util.Scanner;

public class SpeedLimit {
    // Declare a final variable to store the speed limit
    public static final int SPEED_LIMIT = 100;

    // Declare a method to check if the speed is exceeded
    public static void checkSpeed(int speed) {
        // Check if the speed is greater than the speed limit
        if (speed > SPEED_LIMIT) {
            // Generate an alert message
            System.out.println("Alert! Speed limit exceeded. Current speed:
        } else {
            // Display the current speed
            System.out.println("Current speed: " + speed + " km/h.");
        }
    }

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter your current speed (km/h): ");

        int speed = scanner.nextInt();

        checkSpeed(speed);
        scanner.close();
    }
}
```



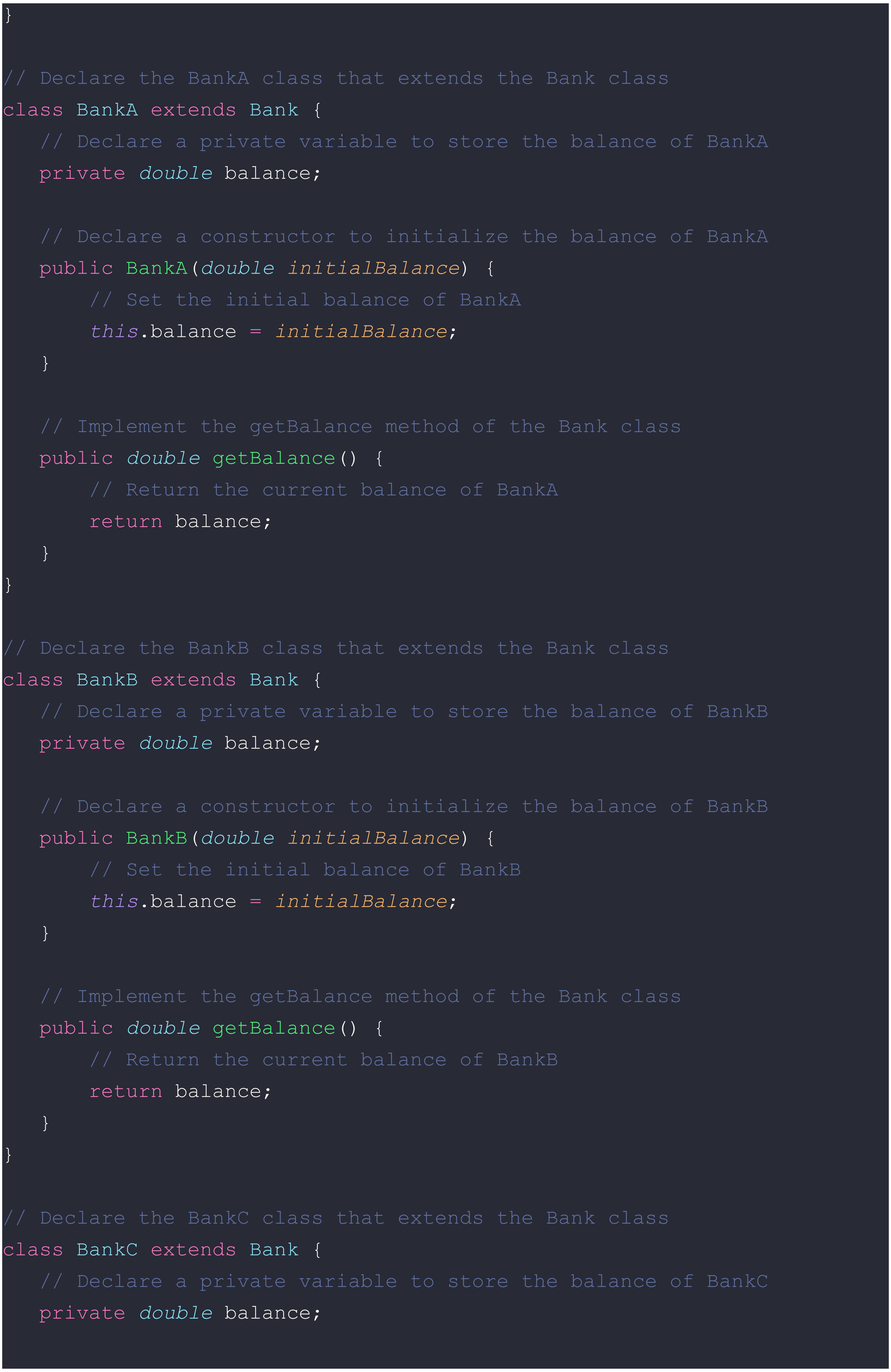
```
edhat.java/jdt_ws/Assignment4_82d29a95/bin SpeedLimit
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Enter your current speed (km/h): 4912
Alert! Speed limit exceeded. Current speed: 4912 km/h. Maximum allowed speed: 100 km/h.
```

Q5.

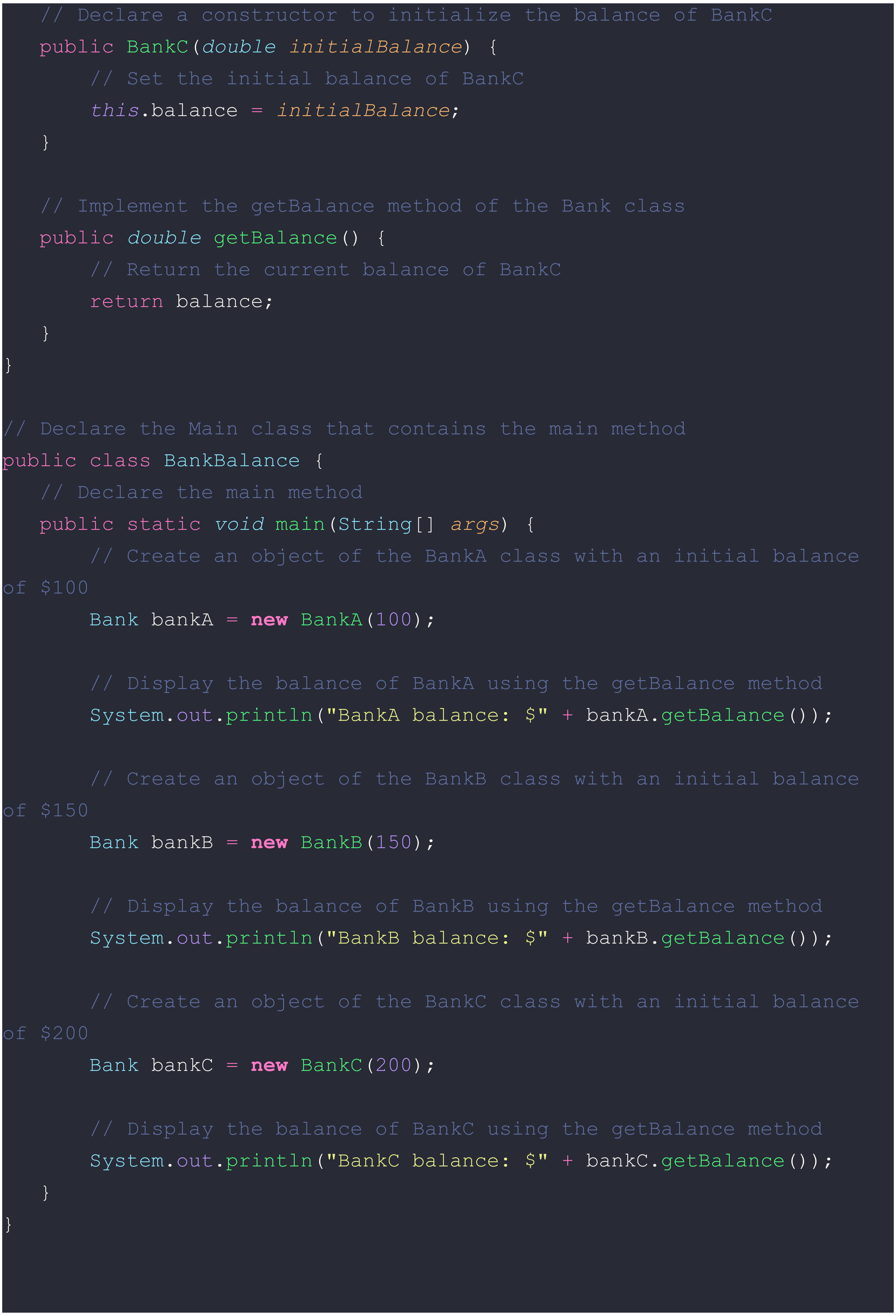
5. Create an abstract class 'Bank' with an abstract method 'getBalance'. $100, $150 and $200 are deposited in banks A, B and C respectively. 'BankA', 'BankB' and 'BankC' are subclasses of class 'Bank', each having a method named 'getBalance'. Call this method by creating an object of each of the three classes.



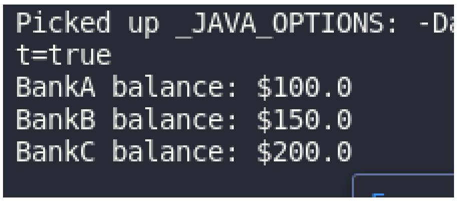
```java
// Declare the abstract Bank class with an abstract getBalance method
abstract class Bank {
   // Declare the abstract getBalance method
   public abstract double getBalance();
```



```
}

// Declare the BankA class that extends the Bank class
class BankA extends Bank {
   // Declare a private variable to store the balance of BankA
   private double balance;

   // Declare a constructor to initialize the balance of BankA
   public BankA(double initialBalance) {
       // Set the initial balance of BankA
       this.balance = initialBalance;
   }

   // Implement the getBalance method of the Bank class
   public double getBalance() {
       // Return the current balance of BankA
       return balance;
   }
}

// Declare the BankB class that extends the Bank class
class BankB extends Bank {
   // Declare a private variable to store the balance of BankB
   private double balance;

   // Declare a constructor to initialize the balance of BankB
   public BankB(double initialBalance) {
       // Set the initial balance of BankB
       this.balance = initialBalance;
   }

   // Implement the getBalance method of the Bank class
   public double getBalance() {
       // Return the current balance of BankB
       return balance;
   }
}

// Declare the BankC class that extends the Bank class
class BankC extends Bank {
   // Declare a private variable to store the balance of BankC
   private double balance;
```

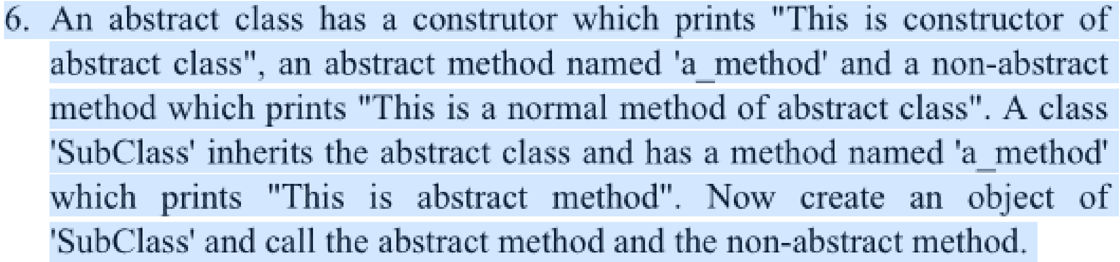


```java
    // Declare a constructor to initialize the balance of BankC
    public BankC(double initialBalance) {
        // Set the initial balance of BankC
        this.balance = initialBalance;
    }

    // Implement the getBalance method of the Bank class
    public double getBalance() {
        // Return the current balance of BankC
        return balance;
    }
}

// Declare the Main class that contains the main method
public class BankBalance {
    // Declare the main method
    public static void main(String[] args) {
        // Create an object of the BankA class with an initial balance
of $100
        Bank bankA = new BankA(100);

        // Display the balance of BankA using the getBalance method
        System.out.println("BankA balance: $" + bankA.getBalance());

        // Create an object of the BankB class with an initial balance
of $150
        Bank bankB = new BankB(150);

        // Display the balance of BankB using the getBalance method
        System.out.println("BankB balance: $" + bankB.getBalance());

        // Create an object of the BankC class with an initial balance
of $200
        Bank bankC = new BankC(200);

        // Display the balance of BankC using the getBalance method
        System.out.println("BankC balance: $" + bankC.getBalance());
    }
}
```



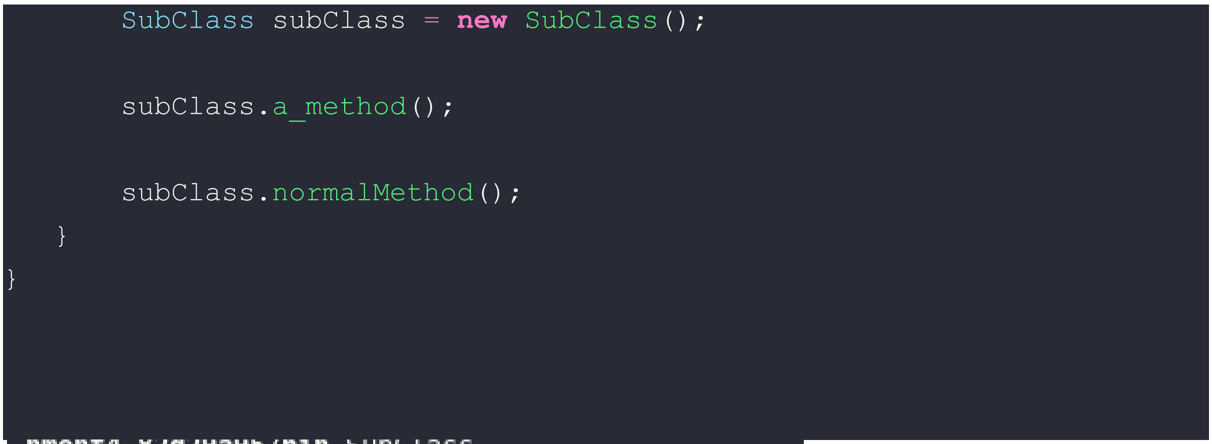
```
Picked up _JAVA_OPTIONS: -Da
t=true
BankA balance: $100.0
BankB balance: $150.0
BankC balance: $200.0
```

Q6.

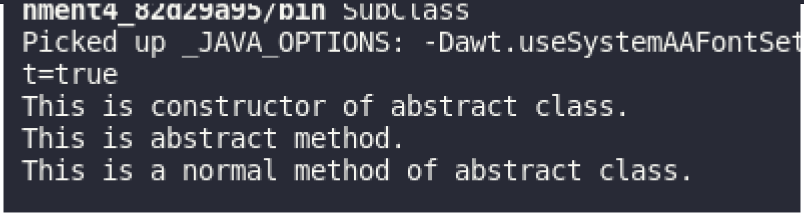


6. An abstract class has a construtor which prints "This is constructor of abstract class", an abstract method named 'a_method' and a non-abstract method which prints "This is a normal method of abstract class". A class 'SubClass' inherits the abstract class and has a method named 'a_method' which prints "This is abstract method". Now create an object of 'SubClass' and call the abstract method and the non-abstract method.

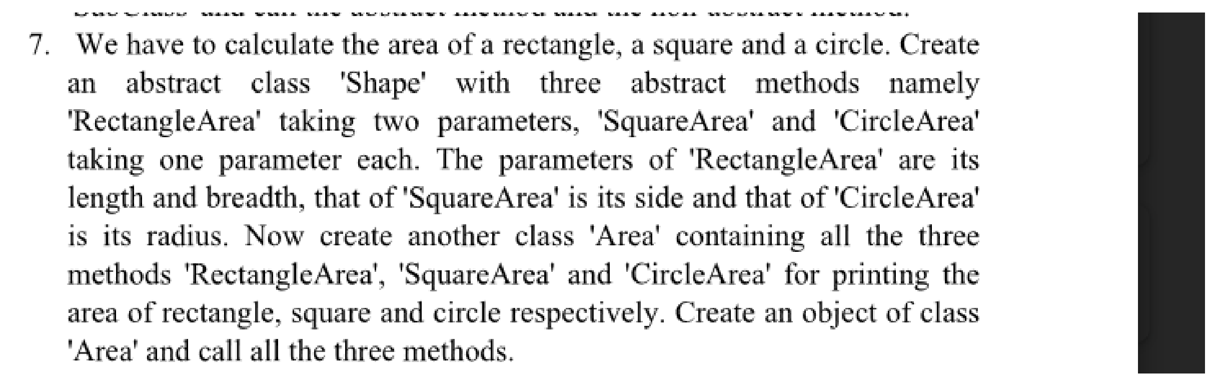
```java
abstract class AbsClass {
   // Declare the constructor of the AbsClass
   public AbsClass() {
       // Print a message when an object of the AbsClass is created
       System.out.println("This is constructor of abstract class.");
   }

   // Declare the abstract method named a_method
   public abstract void a_method();

   // Declare the non-abstract method named normalMethod
   public void normalMethod() {
       // Print a message when the normalMethod is called
       System.out.println("This is a normal method of abstract
class.");
   }
}


public class SubClass extends AbsClass {

   public void a_method() {

       System.out.println("This is abstract method.");
   }


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



```
        SubClass subClass = new SubClass();

        subClass.a_method();

        subClass.normalMethod();
    }
}
```

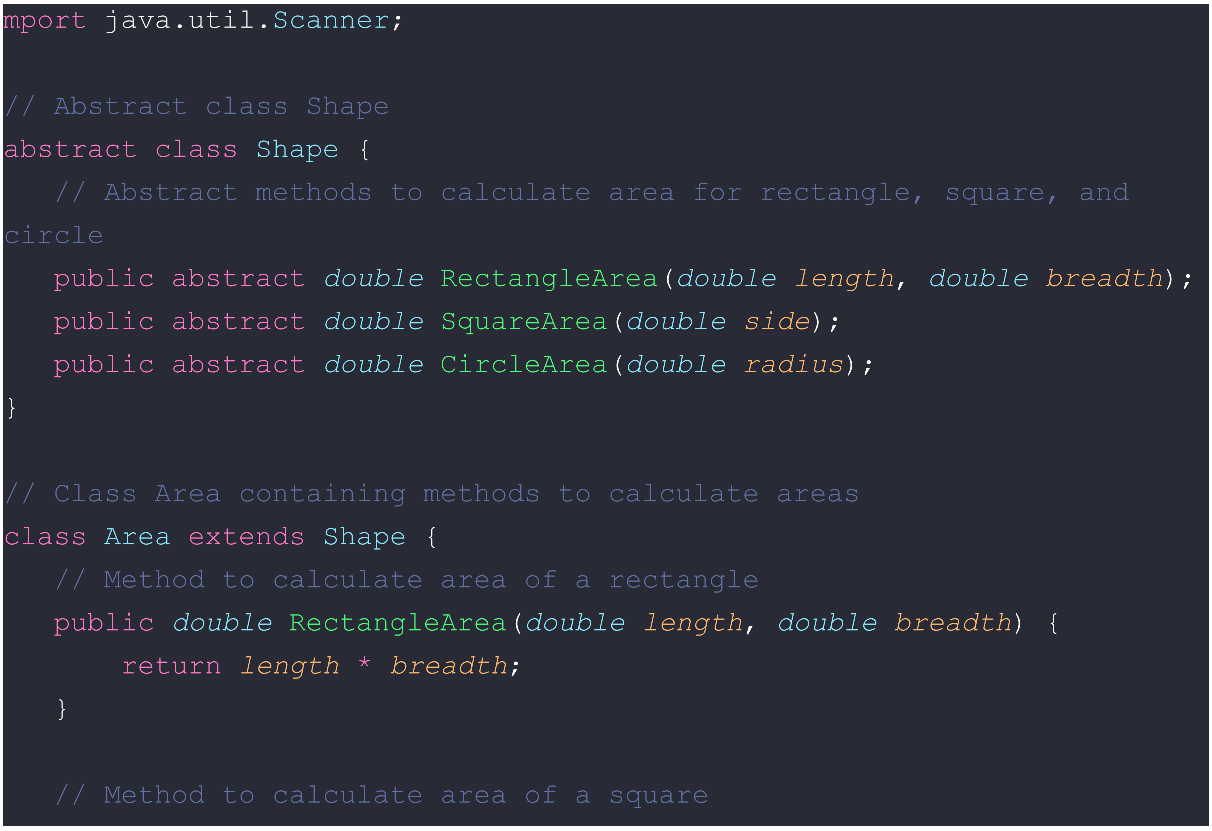


```
nment4_82d29a95/bin SubClass
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSet
t=true
This is constructor of abstract class.
This is abstract method.
This is a normal method of abstract class.
```

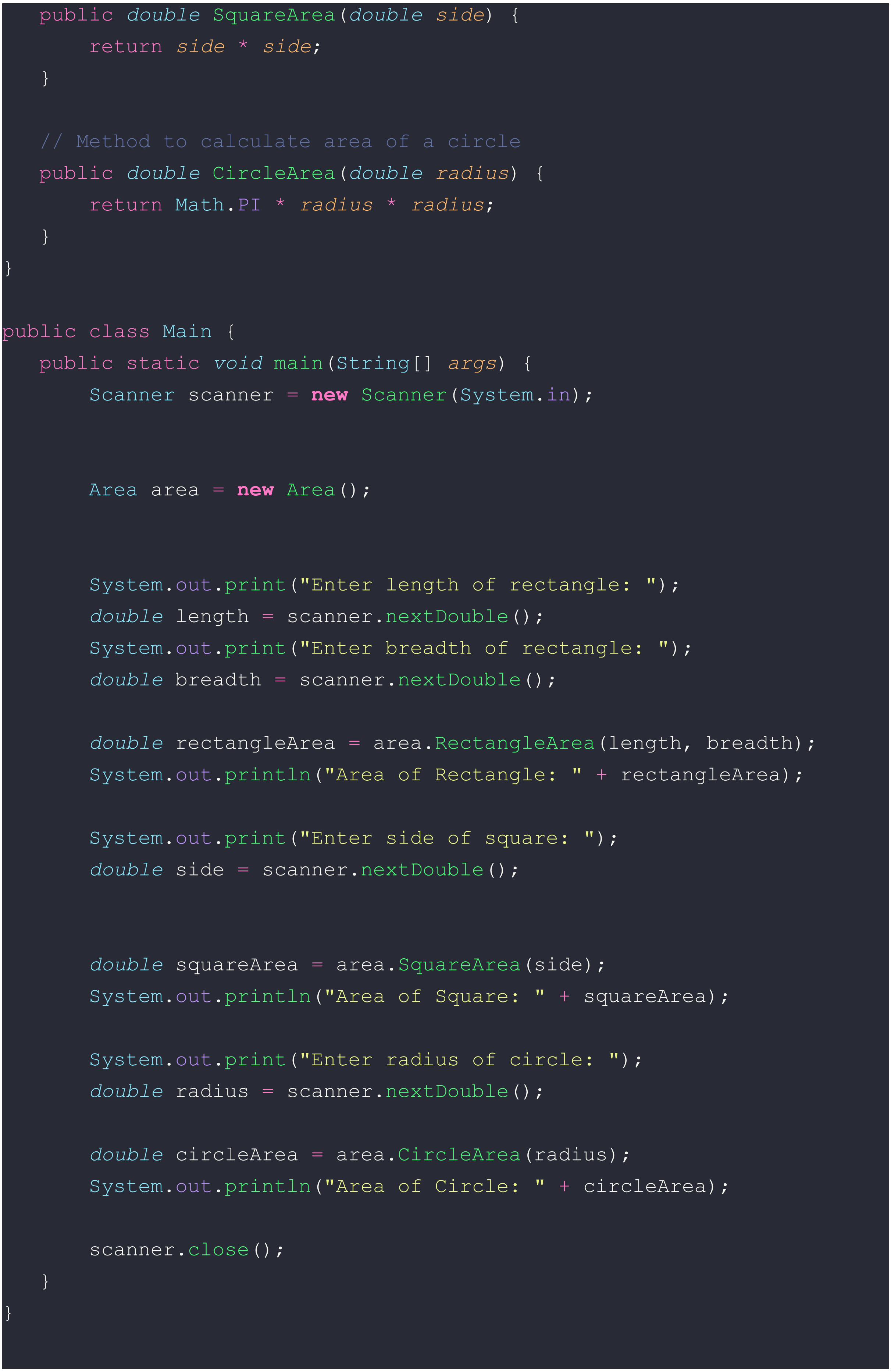


SubClass and call the abstract method and the non abstract method.

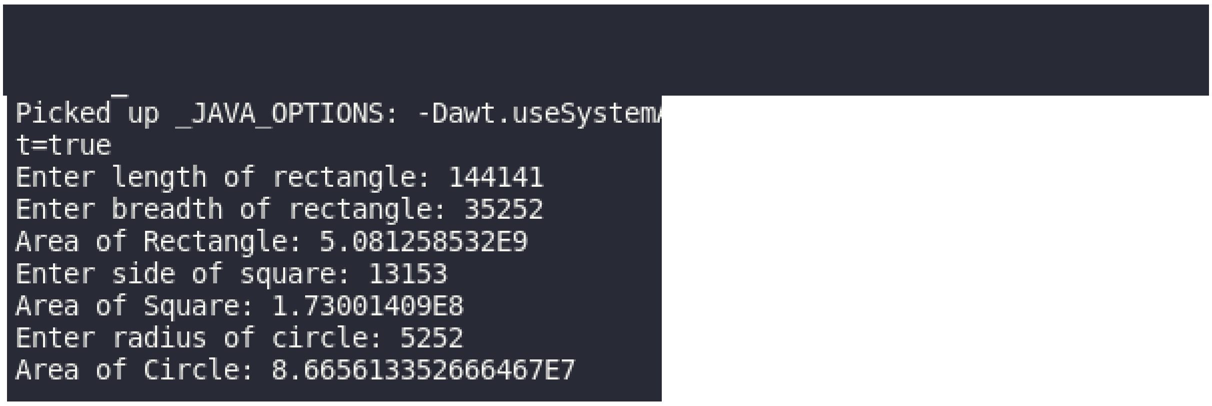
7. We have to calculate the area of a rectangle, a square and a circle. Create an abstract class 'Shape' with three abstract methods namely 'RectangleArea' taking two parameters, 'SquareArea' and 'CircleArea' taking one parameter each. The parameters of 'RectangleArea' are its length and breadth, that of 'SquareArea' is its side and that of 'CircleArea' is its radius. Now create another class 'Area' containing all the three methods 'RectangleArea', 'SquareArea' and 'CircleArea' for printing the area of rectangle, square and circle respectively. Create an object of class 'Area' and call all the three methods.



```
mport java.util.Scanner;

// Abstract class Shape
abstract class Shape {
    // Abstract methods to calculate area for rectangle, square, and
circle
    public abstract double RectangleArea(double length, double breadth);
    public abstract double SquareArea(double side);
    public abstract double CircleArea(double radius);
}

// Class Area containing methods to calculate areas
class Area extends Shape {
    // Method to calculate area of a rectangle
    public double RectangleArea(double length, double breadth) {
        return length * breadth;
    }

    // Method to calculate area of a square
```



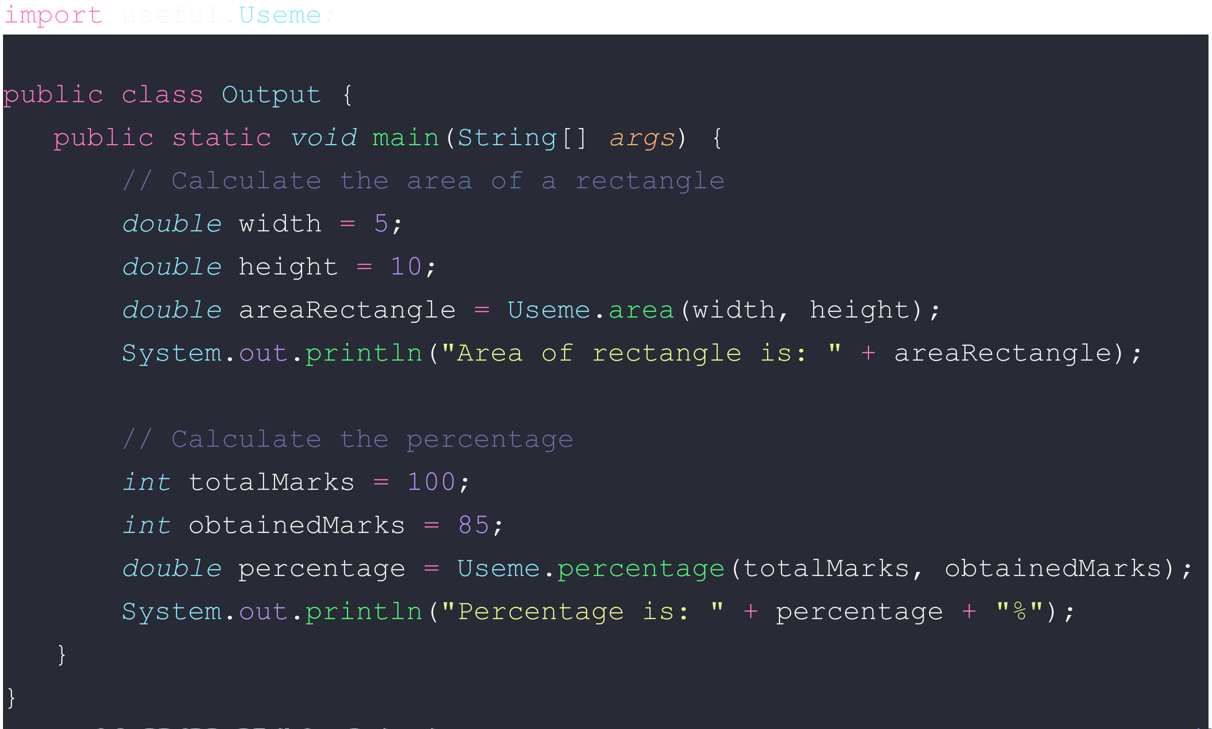
```java
    public double SquareArea(double side) {
        return side * side;
    }

    // Method to calculate area of a circle
    public double CircleArea(double radius) {
        return Math.PI * radius * radius;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);


        Area area = new Area();


        System.out.print("Enter length of rectangle: ");
        double length = scanner.nextDouble();
        System.out.print("Enter breadth of rectangle: ");
        double breadth = scanner.nextDouble();

        double rectangleArea = area.RectangleArea(length, breadth);
        System.out.println("Area of Rectangle: " + rectangleArea);

        System.out.print("Enter side of square: ");
        double side = scanner.nextDouble();


        double squareArea = area.SquareArea(side);
        System.out.println("Area of Square: " + squareArea);

        System.out.print("Enter radius of circle: ");
        double radius = scanner.nextDouble();

        double circleArea = area.CircleArea(radius);
        System.out.println("Area of Circle: " + circleArea);

        scanner.close();
    }
}
```



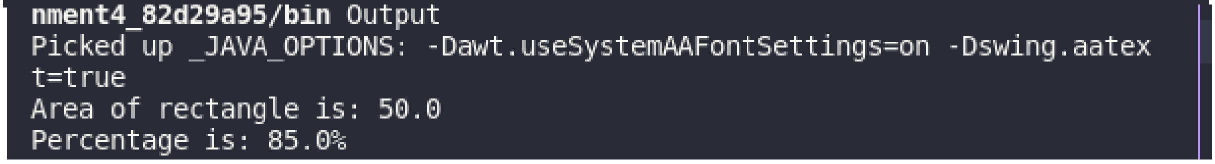
```
Picked up _JAVA_OPTIONS: -Dawt.useSystem
t=true
Enter length of rectangle: 144141
Enter breadth of rectangle: 35252
Area of Rectangle: 5.081258532E9
Enter side of square: 13153
Area of Square: 1.73001409E8
Enter radius of circle: 5252
Area of Circle: 8.665613352666467E7
```

8. Define a package named ‘useful’ with a class name ‘Useme’ having following methods:
   1. area() → To calculate area of given shape.
   2. percentage() → to calculate percentage given total marks and marks obtained.

   Develop a program to import above package and use both methods.



```java
import useful.Useme;

public class Output {
   public static void main(String[] args) {
       // Calculate the area of a rectangle
       double width = 5;
       double height = 10;
       double areaRectangle = Useme.area(width, height);
       System.out.println("Area of rectangle is: " + areaRectangle);

       // Calculate the percentage
       int totalMarks = 100;
       int obtainedMarks = 85;
       double percentage = Useme.percentage(totalMarks, obtainedMarks);
       System.out.println("Percentage is: " + percentage + "%");
   }
}
```



```
nment4_82d29a95/bin Output
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatex
t=true
Area of rectangle is: 50.0
Percentage is: 85.0%
```