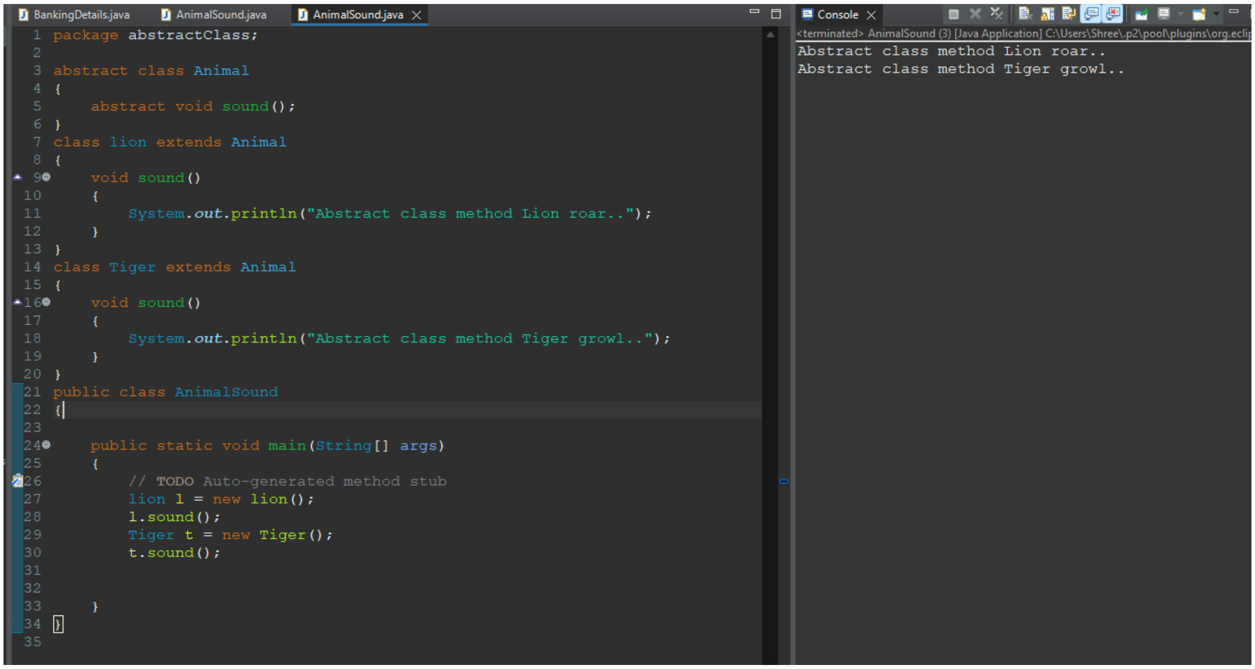
Assignment No:-27

Name:-Suryawanshi Sangramsingh Sambhaji

Batch: - Delta - DCA (Java) 2024 Date:-12/6/2024

Abstraction:

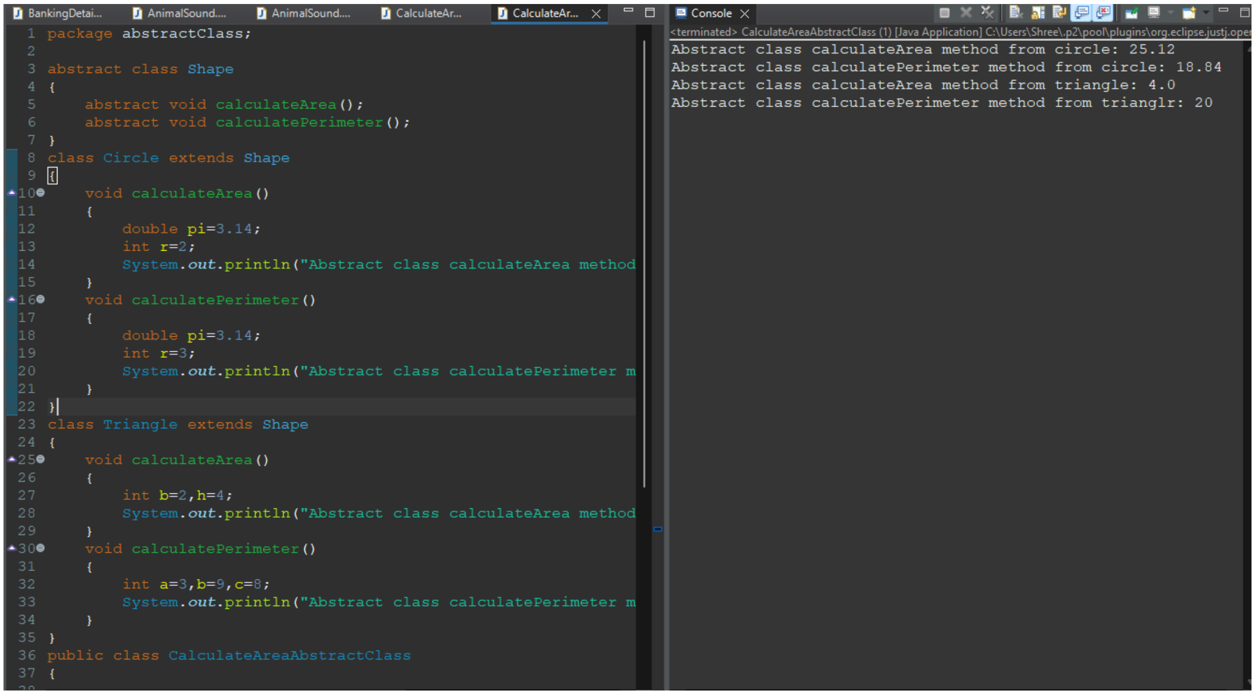
1. Write a Java program to create an abstract class Animal with an abstract method called sound(). Create subclasses Lion and Tiger that extend the Animal class and implement the sound() method to make a specific sound for each animal.



```java
package abstractClass;

abstract class Animal
{
    abstract void sound();
}
class lion extends Animal
{
    void sound()
    {
        System.out.println("Abstract class method Lion roar..");
    }
}
class Tiger extends Animal
{
    void sound()
    {
        System.out.println("Abstract class method Tiger growl..");
    }
}
public class AnimalSound
{

    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        lion l = new lion();
        l.sound();
        Tiger t = new Tiger();
        t.sound();


    }
}
```

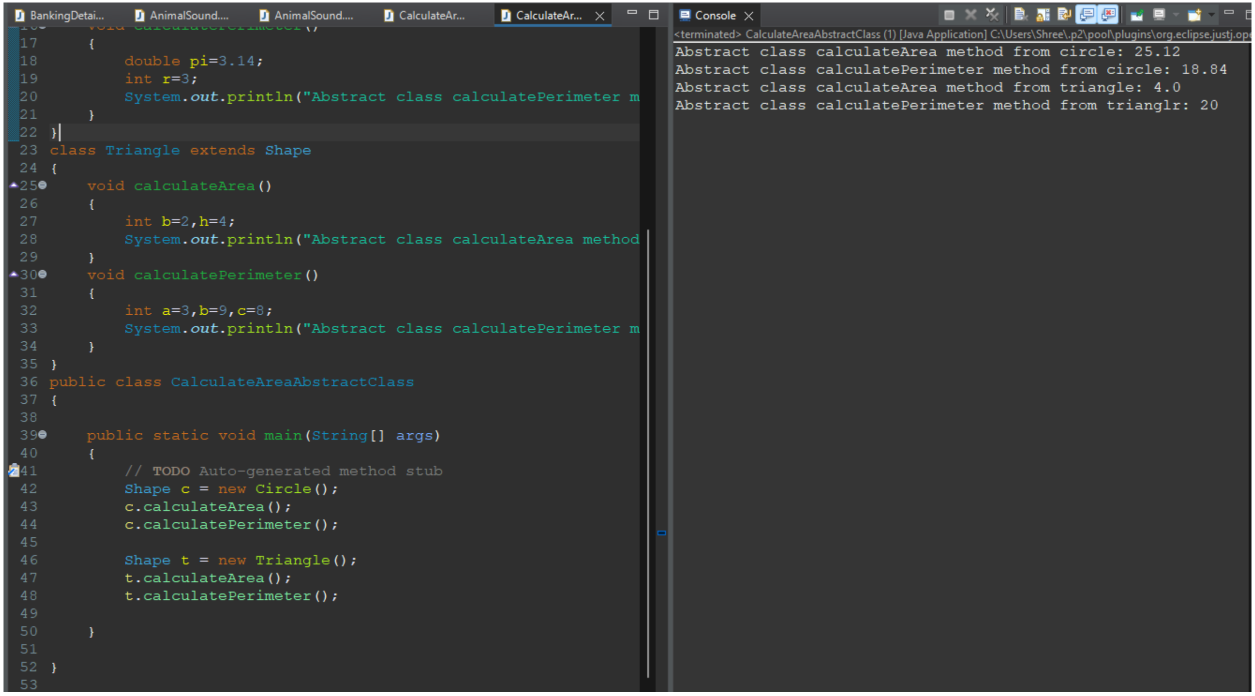
```
<terminated> AnimalSound (3) [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclip
Abstract class method Lion roar..
Abstract class method Tiger growl..
```

2. Write a Java program to create an abstract class Shape with abstract methods calculateArea() and calculatePerimeter(). Create subclasses Circle and Triangle that extend the Shape class and implement the respective methods to calculate the area and perimeter of each shape.



```java
package abstractClass;

abstract class Shape
{
    abstract void calculateArea();
    abstract void calculatePerimeter();
}
class Circle extends Shape
{
    void calculateArea()
    {
        double pi=3.14;
        int r=2;
        System.out.println("Abstract class calculateArea method
    }
    void calculatePerimeter()
    {
        double pi=3.14;
        int r=3;
        System.out.println("Abstract class calculatePerimeter m
    }
}
class Triangle extends Shape
{
    void calculateArea()
    {
        int b=2,h=4;
        System.out.println("Abstract class calculateArea method
    }
    void calculatePerimeter()
    {
        int a=3,b=9,c=8;
        System.out.println("Abstract class calculatePerimeter m
    }
}
public class CalculateAreaAbstractClass
{
```

```
<terminated> CalculateAreaAbstractClass (1) [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.oper
Abstract class calculateArea method from circle: 25.12
Abstract class calculatePerimeter method from circle: 18.84
Abstract class calculateArea method from triangle: 4.0
Abstract class calculatePerimeter method from trianglr: 20
```

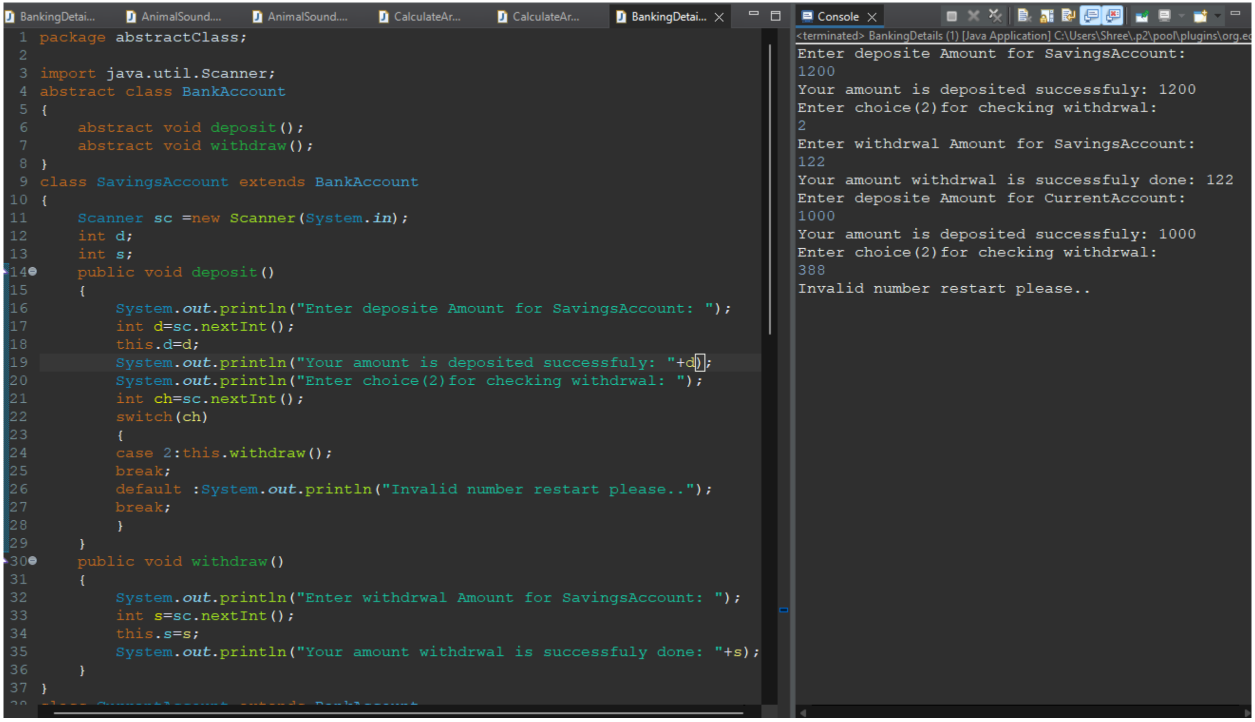


```java
    {
        double pi=3.14;
        int r=3;
        System.out.println("Abstract class calculatePerimeter m
    }
}
class Triangle extends Shape
{
    void calculateArea()
    {
        int b=2,h=4;
        System.out.println("Abstract class calculateArea method
    }
    void calculatePerimeter()
    {
        int a=3,b=9,c=8;
        System.out.println("Abstract class calculatePerimeter m
    }
}
public class CalculateAreaAbstractClass
{
```

```java

    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Shape c = new Circle();
        c.calculateArea();
        c.calculatePerimeter();

        Shape t = new Triangle();
        t.calculateArea();
        t.calculatePerimeter();

    }

}

```

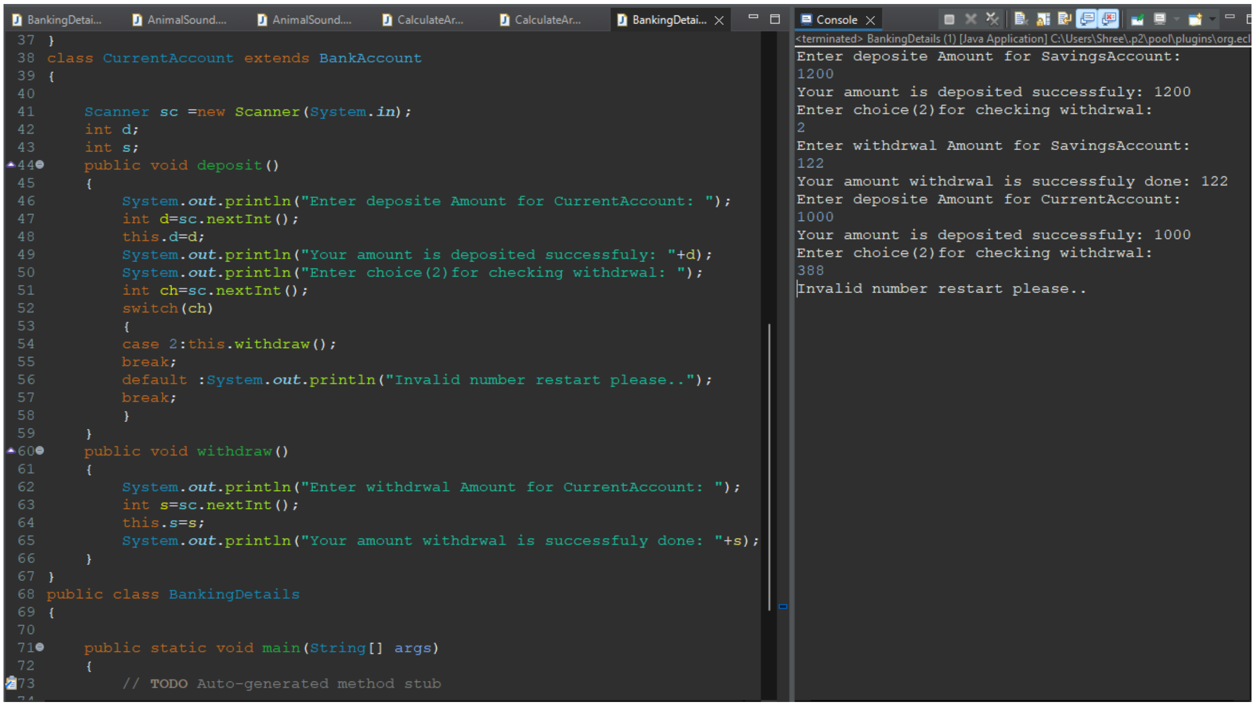
```
<terminated> CalculateAreaAbstractClass (1) [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.ope
Abstract class calculateArea method from circle: 25.12
Abstract class calculatePerimeter method from circle: 18.84
Abstract class calculateArea method from triangle: 4.0
Abstract class calculatePerimeter method from trianglr: 20
```

3. Write a Java program to create an abstract class BankAccount with abstract methods deposit() and withdraw(). Create subclasses: SavingsAccount and CurrentAccount that extend the BankAccount class and implement the respective methods to handle deposits and withdrawals for each account type.


BankingDetai...
AnimalSound....
AnimalSound....
CalculateAr...
CalculateAr...
BankingDetai...
Console
<terminated> BankingDetails (1) [Java Application] C:\Users\Shree\.p2\pool\plugins\org.ec

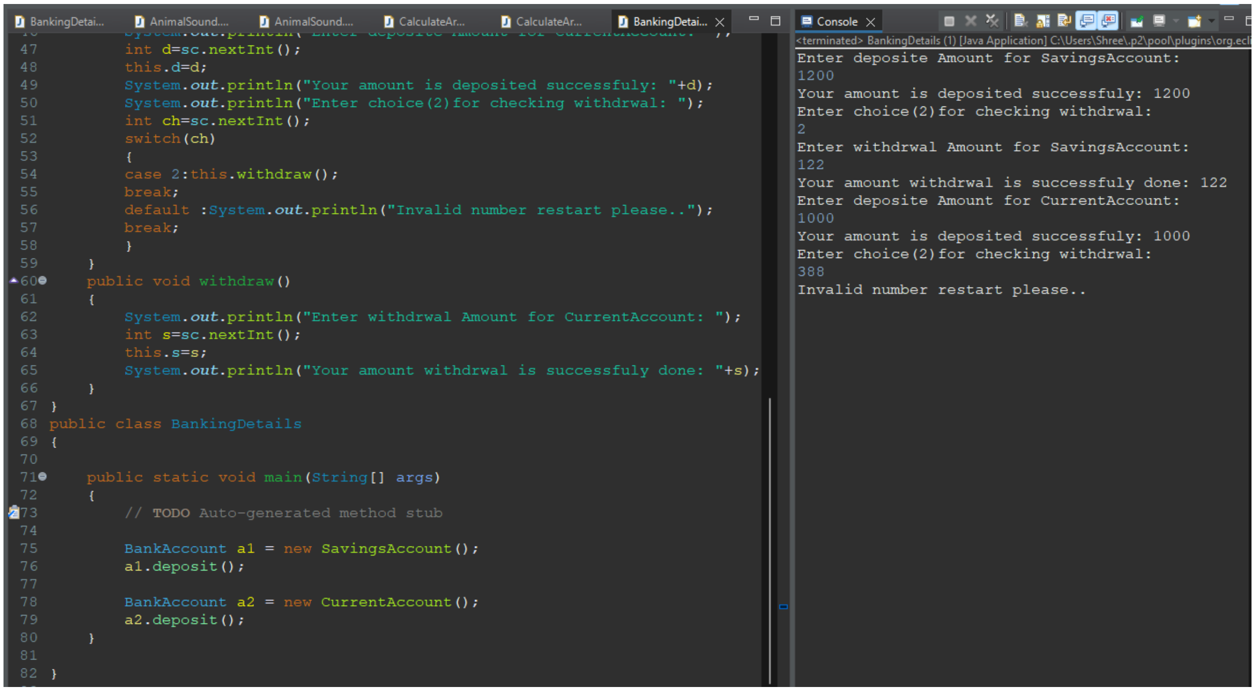
```
package abstractClass;

import java.util.Scanner;
abstract class BankAccount
{
    abstract void deposit();
    abstract void withdraw();
}
class SavingsAccount extends BankAccount
{
    Scanner sc =new Scanner(System.in);
    int d;
    int s;
    public void deposit()
    {
        System.out.println("Enter deposite Amount for SavingsAccount: ");
        int d=sc.nextInt();
        this.d=d;
        System.out.println("Your amount is deposited successfuly: "+d);
        System.out.println("Enter choice(2)for checking withdrwal: ");
        int ch=sc.nextInt();
        switch(ch)
        {
        case 2:this.withdraw();
        break;
        default :System.out.println("Invalid number restart please..");
        break;
        }
    }
    public void withdraw()
    {
        System.out.println("Enter withdrwal Amount for SavingsAccount: ");
        int s=sc.nextInt();
        this.s=s;
        System.out.println("Your amount withdrwal is successfuly done: "+s);
    }
}
```

```
Enter deposite Amount for SavingsAccount:
1200
Your amount is deposited successfuly: 1200
Enter choice(2)for checking withdrwal:
2
Enter withdrwal Amount for SavingsAccount:
122
Your amount withdrwal is successfuly done: 122
Enter deposite Amount for CurrentAccount:
1000
Your amount is deposited successfuly: 1000
Enter choice(2)for checking withdrwal:
388
Invalid number restart please..
```


BankingDetai...
AnimalSound....
AnimalSound....
CalculateAr...
CalculateAr...
BankingDetai...
Console
<terminated> BankingDetails (1) [Java Application] C:\Users\Shree\.p2\pool\plugins\org.ecl

```
}
class CurrentAccount extends BankAccount
{

    Scanner sc =new Scanner(System.in);
    int d;
    int s;
    public void deposit()
    {
        System.out.println("Enter deposite Amount for CurrentAccount: ");
        int d=sc.nextInt();
        this.d=d;
        System.out.println("Your amount is deposited successfuly: "+d);
        System.out.println("Enter choice(2)for checking withdrwal: ");
        int ch=sc.nextInt();
        switch(ch)
        {
        case 2:this.withdraw();
        break;
        default :System.out.println("Invalid number restart please..");
        break;
        }
    }
    public void withdraw()
    {
        System.out.println("Enter withdrwal Amount for CurrentAccount: ");
        int s=sc.nextInt();
        this.s=s;
        System.out.println("Your amount withdrwal is successfuly done: "+s);
    }
}
public class BankingDetails
{

    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
```

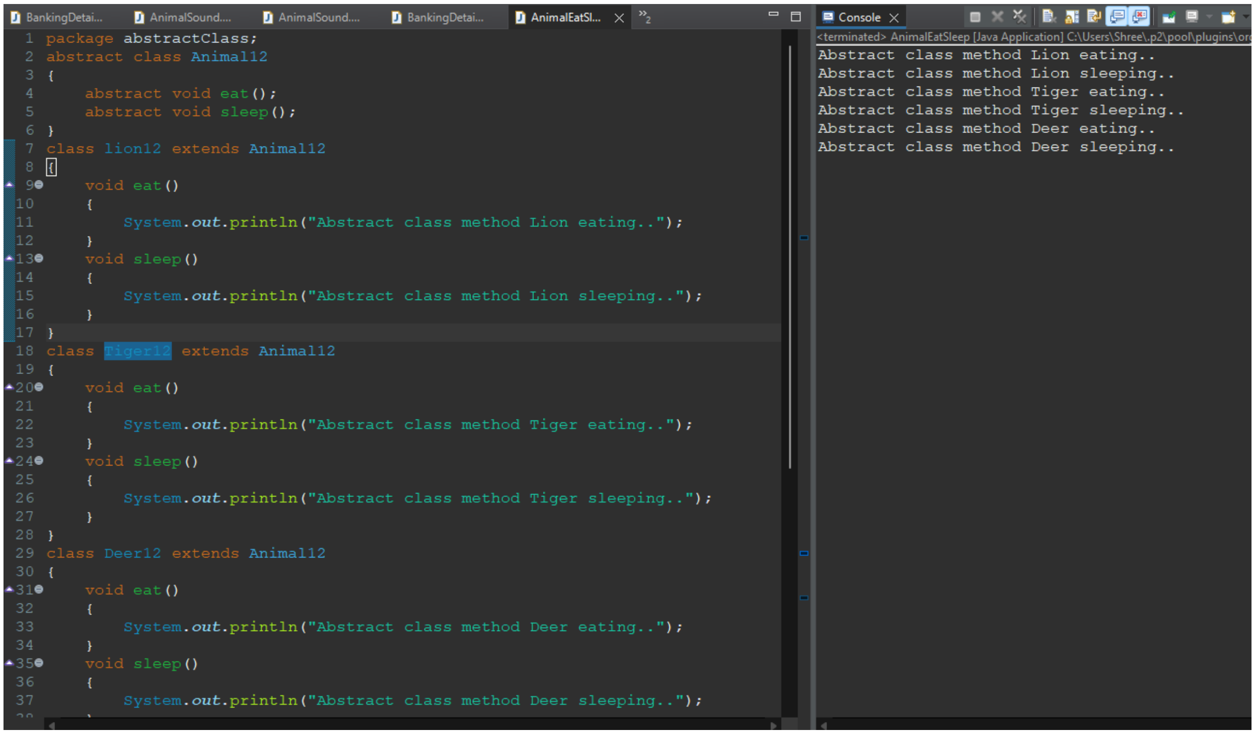
```
Enter deposite Amount for SavingsAccount:
1200
Your amount is deposited successfuly: 1200
Enter choice(2)for checking withdrwal:
2
Enter withdrwal Amount for SavingsAccount:
122
Your amount withdrwal is successfuly done: 122
Enter deposite Amount for CurrentAccount:
1000
Your amount is deposited successfuly: 1000
Enter choice(2)for checking withdrwal:
388
Invalid number restart please..
```



```java
        System.out.println("Enter deposite Amount for CurrentAccount: ");
        int d=sc.nextInt();
        this.d=d;
        System.out.println("Your amount is deposited successfuly: "+d);
        System.out.println("Enter choice(2)for checking withdrwal: ");
        int ch=sc.nextInt();
        switch(ch)
        {
        case 2:this.withdraw();
        break;
        default :System.out.println("Invalid number restart please..");
        break;
        }
    }
    public void withdraw()
    {
        System.out.println("Enter withdrwal Amount for CurrentAccount: ");
        int s=sc.nextInt();
        this.s=s;
        System.out.println("Your amount withdrwal is successfuly done: "+s);
    }
}
public class BankingDetails
{

    public static void main(String[] args)
    {
        // TODO Auto-generated method stub

        BankAccount a1 = new SavingsAccount();
        a1.deposit();

        BankAccount a2 = new CurrentAccount();
        a2.deposit();
    }

}
```

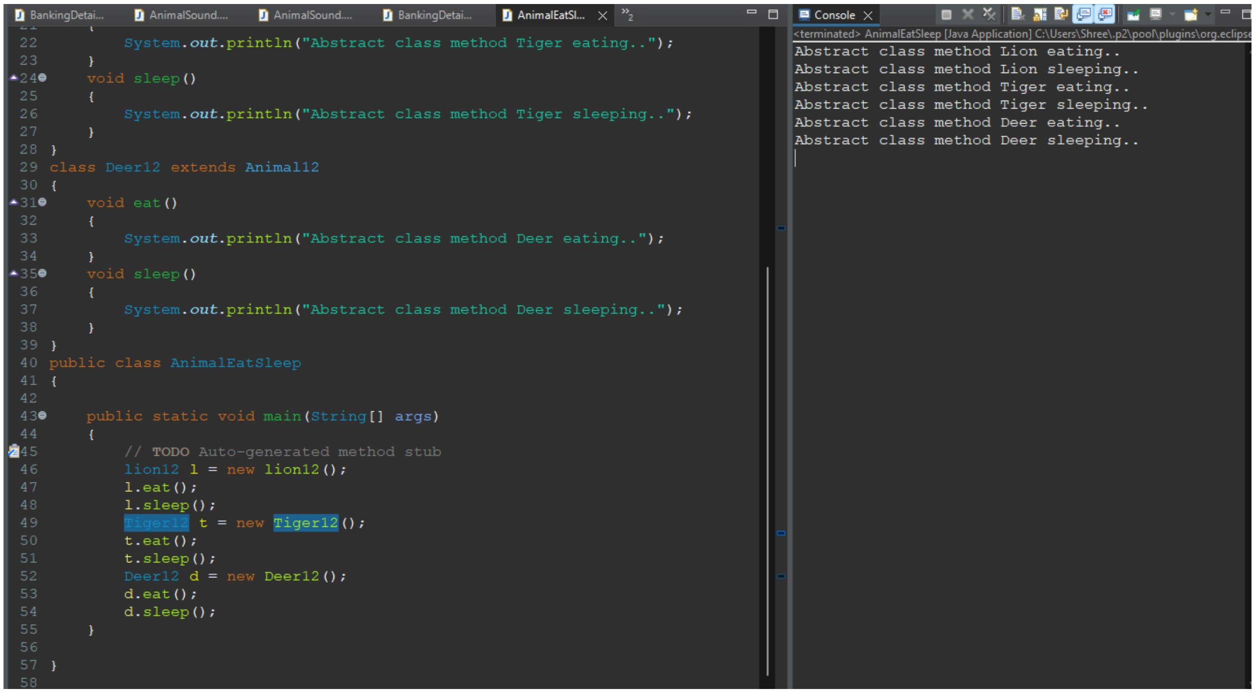
```
<terminated> BankingDetails (1) [Java Application] C:\Users\Shree\.p2\pool\plugins\org.ecli
Enter deposite Amount for SavingsAccount:
1200
Your amount is deposited successfuly: 1200
Enter choice(2)for checking withdrwal:
2
Enter withdrwal Amount for SavingsAccount:
122
Your amount withdrwal is successfuly done: 122
Enter deposite Amount for CurrentAccount:
1000
Your amount is deposited successfuly: 1000
Enter choice(2)for checking withdrwal:
388
Invalid number restart please..
```

4. Write a Java program to create an abstract class Animal with abstract methods eat() and sleep(). Create subclasses Lion, Tiger, and Deer that extend the Animal class and implement the eat() and sleep() methods differently based on their specific behavior.



```java
package abstractClass;
abstract class Animal12
{
    abstract void eat();
    abstract void sleep();
}
class lion12 extends Animal12
{
    void eat()
    {
        System.out.println("Abstract class method Lion eating..");
    }
    void sleep()
    {
        System.out.println("Abstract class method Lion sleeping..");
    }
}
class Tiger12 extends Animal12
{
    void eat()
    {
        System.out.println("Abstract class method Tiger eating..");
    }
    void sleep()
    {
        System.out.println("Abstract class method Tiger sleeping..");
    }
}
class Deer12 extends Animal12
{
    void eat()
    {
        System.out.println("Abstract class method Deer eating..");
    }
    void sleep()
    {
        System.out.println("Abstract class method Deer sleeping..");
```

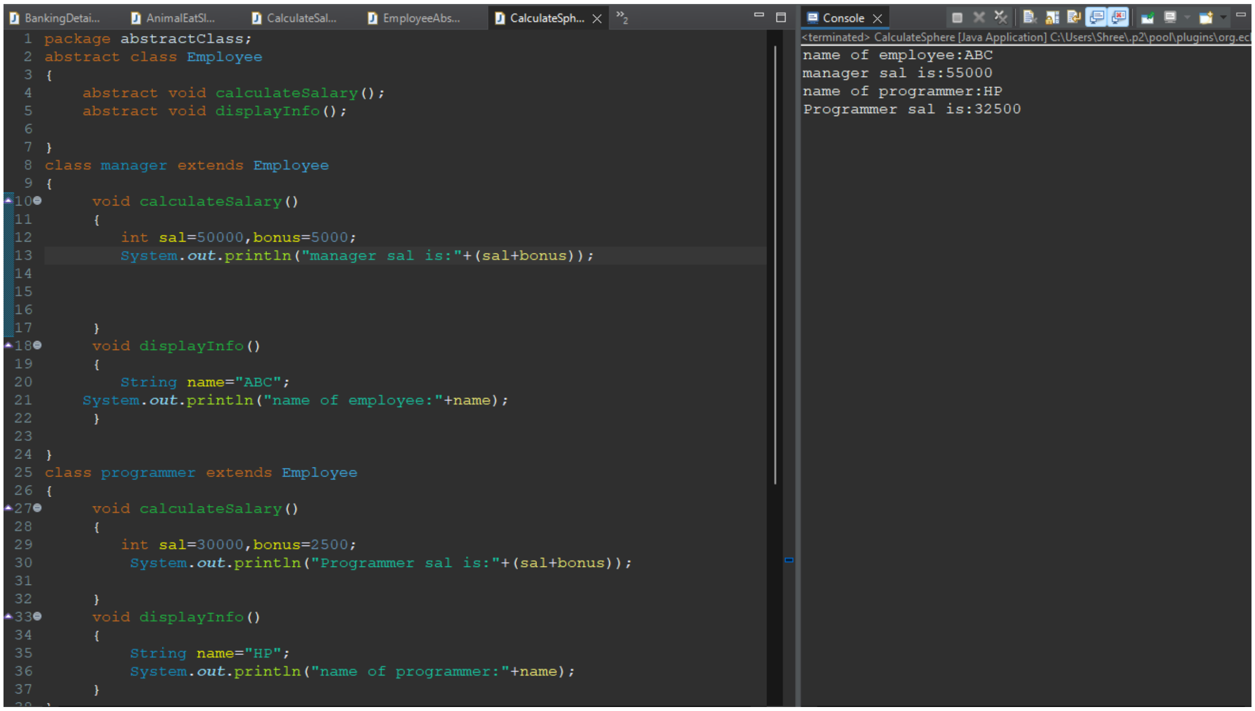
```
<terminated> AnimalEatSleep [Java Application] C:\Users\Shree\.p2\pool\plugins\or
Abstract class method Lion eating..
Abstract class method Lion sleeping..
Abstract class method Tiger eating..
Abstract class method Tiger sleeping..
Abstract class method Deer eating..
Abstract class method Deer sleeping..
```



```
        System.out.println("Abstract class method Tiger eating..");
    }
    void sleep()
    {
        System.out.println("Abstract class method Tiger sleeping..");
    }
}
class Deer12 extends Animal12
{
    void eat()
    {
        System.out.println("Abstract class method Deer eating..");
    }
    void sleep()
    {
        System.out.println("Abstract class method Deer sleeping..");
    }
}
public class AnimalEatSleep
{

    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        lion12 l = new lion12();
        l.eat();
        l.sleep();
        Tiger12 t = new Tiger12();
        t.eat();
        t.sleep();
        Deer12 d = new Deer12();
        d.eat();
        d.sleep();
    }

}
```

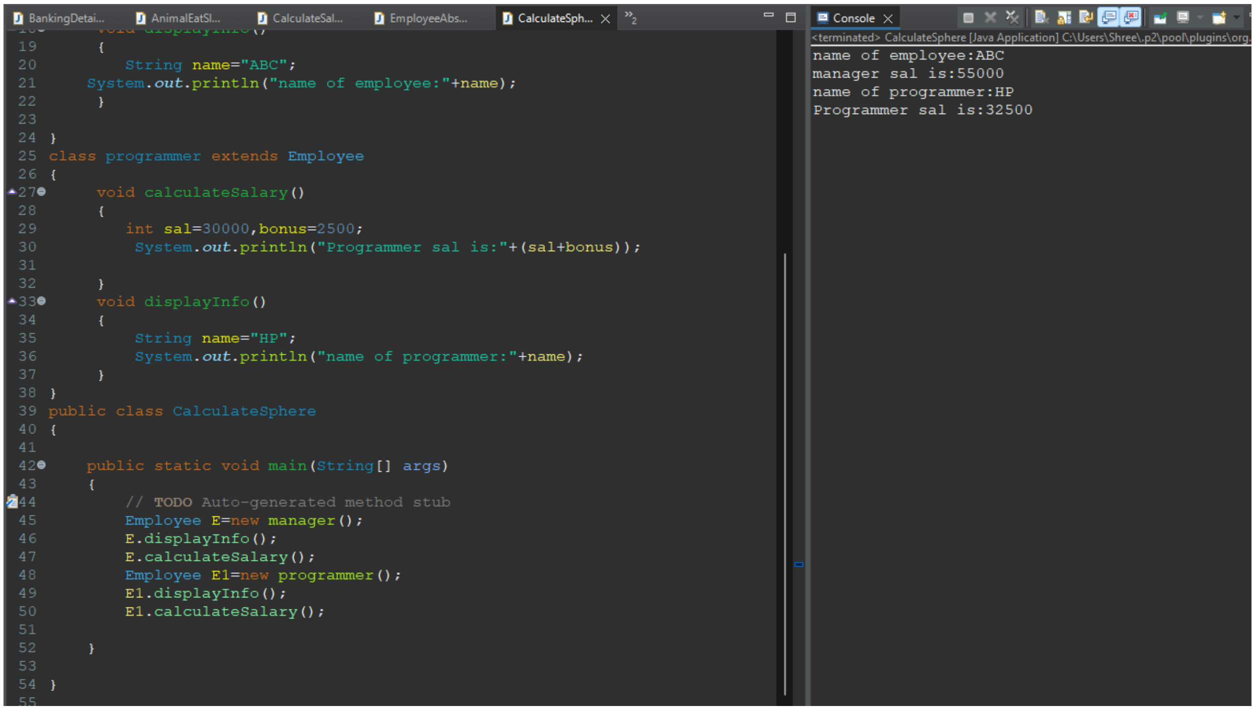
```
<terminated> AnimalEatSleep [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse
Abstract class method Lion eating..
Abstract class method Lion sleeping..
Abstract class method Tiger eating..
Abstract class method Tiger sleeping..
Abstract class method Deer eating..
Abstract class method Deer sleeping..
```

5. Write a Java program to create an abstract class Employee with abstract methods calculateSalary() and displayInfo(). Create subclasses Manager and Programmer that extend the Employee class and implement the respective methods to calculate salary and display information for each role.



```
package abstractClass;
abstract class Employee
{
    abstract void calculateSalary();
    abstract void displayInfo();

}
class manager extends Employee
{
     void calculateSalary()
     {
        int sal=50000,bonus=5000;
        System.out.println("manager sal is:"+(sal+bonus));



     }
     void displayInfo()
     {
        String name="ABC";
    System.out.println("name of employee:"+name);
     }

}
class programmer extends Employee
{
     void calculateSalary()
     {
        int sal=30000,bonus=2500;
         System.out.println("Programmer sal is:"+(sal+bonus));

     }
     void displayInfo()
     {
         String name="HP";
         System.out.println("name of programmer:"+name);
     }
```

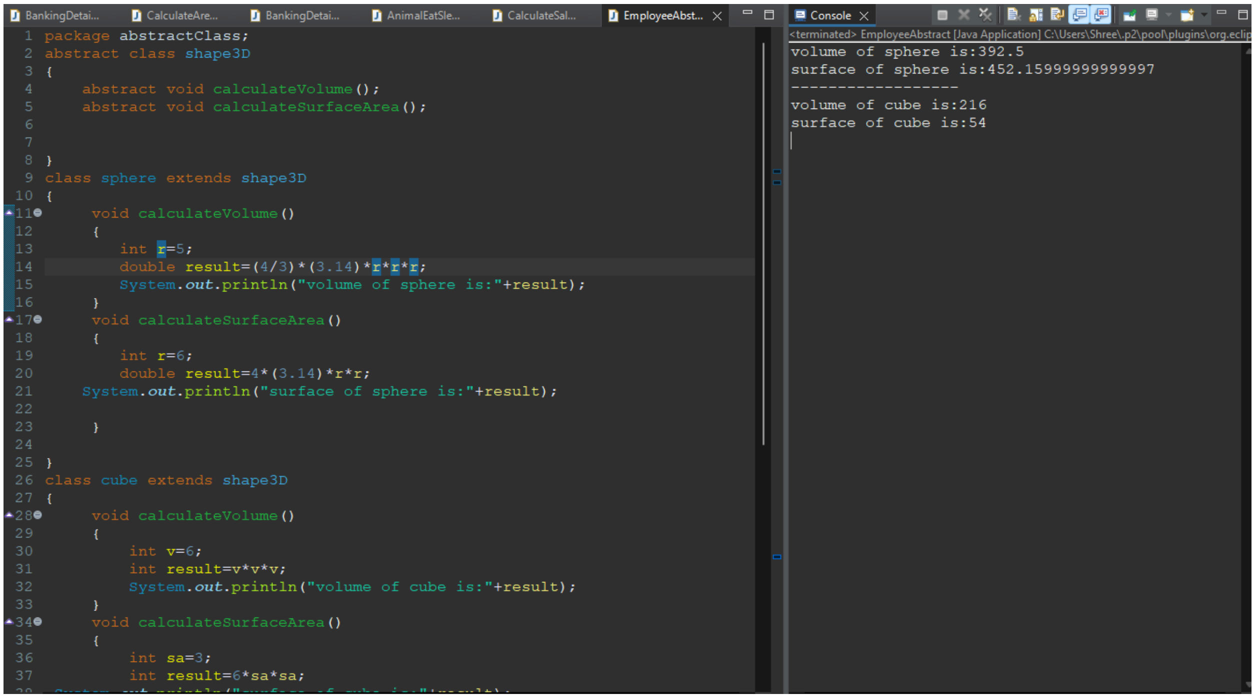
```
<terminated> CalculateSphere [Java Application] C:\Users\Shree\.p2\pool\plugins\org.ec
name of employee:ABC
manager sal is:55000
name of programmer:HP
Programmer sal is:32500
```



```java
    void displayInfo()
     {
        String name="ABC";
    System.out.println("name of employee:"+name);
     }

}
class programmer extends Employee
{
     void calculateSalary()
     {
        int sal=30000,bonus=2500;
         System.out.println("Programmer sal is:"+(sal+bonus));

     }
     void displayInfo()
     {
         String name="HP";
         System.out.println("name of programmer:"+name);
     }
}
public class CalculateSphere
{

    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Employee E=new manager();
        E.displayInfo();
        E.calculateSalary();
        Employee E1=new programmer();
        E1.displayInfo();
        E1.calculateSalary();

    }

}
```

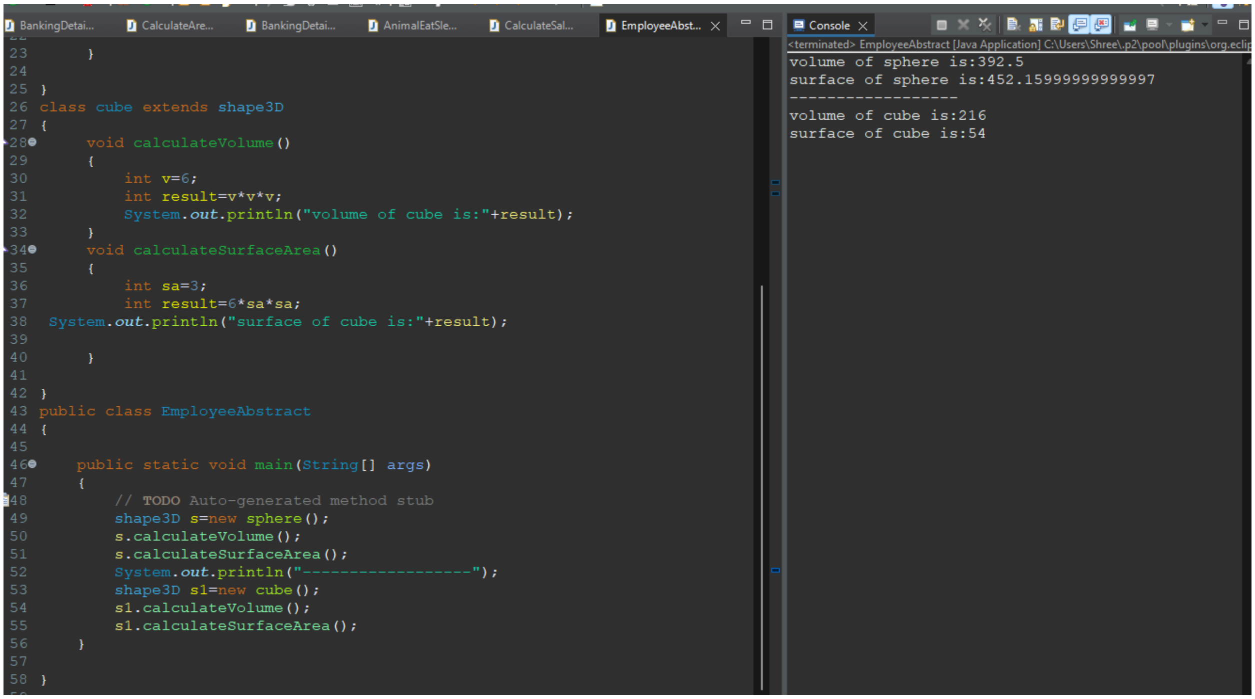
```
<terminated> CalculateSphere [Java Application] C:\Users\Shree\.p2\pool\plugins\org.
name of employee:ABC
manager sal is:55000
name of programmer:HP
Programmer sal is:32500
```

6. Write a Java program to create an abstract class Shape3D with abstract methods calculateVolume() and calculateSurfaceArea(). Create subclasses Sphere and Cube that extend the Shape3D class and implement the respective methods to calculate the volume and surface area of each shape.



```java
package abstractClass;
abstract class shape3D
{
    abstract void calculateVolume();
    abstract void calculateSurfaceArea();


}
class sphere extends shape3D
{
     void calculateVolume()
     {
        int r=5;
        double result=(4/3)*(3.14)*r*r*r;
        System.out.println("volume of sphere is:"+result);
     }
     void calculateSurfaceArea()
     {
        int r=6;
        double result=4*(3.14)*r*r;
    System.out.println("surface of sphere is:"+result);

     }

}
class cube extends shape3D
{
     void calculateVolume()
     {
         int v=6;
         int result=v*v*v;
         System.out.println("volume of cube is:"+result);
     }
     void calculateSurfaceArea()
     {
         int sa=3;
         int result=6*sa*sa;
 System.out.println("surface of cube is:"+result);
```

```
<terminated> EmployeeAbstract [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclip
volume of sphere is:392.5
surface of sphere is:452.15999999999997
------------------
volume of cube is:216
surface of cube is:54
```

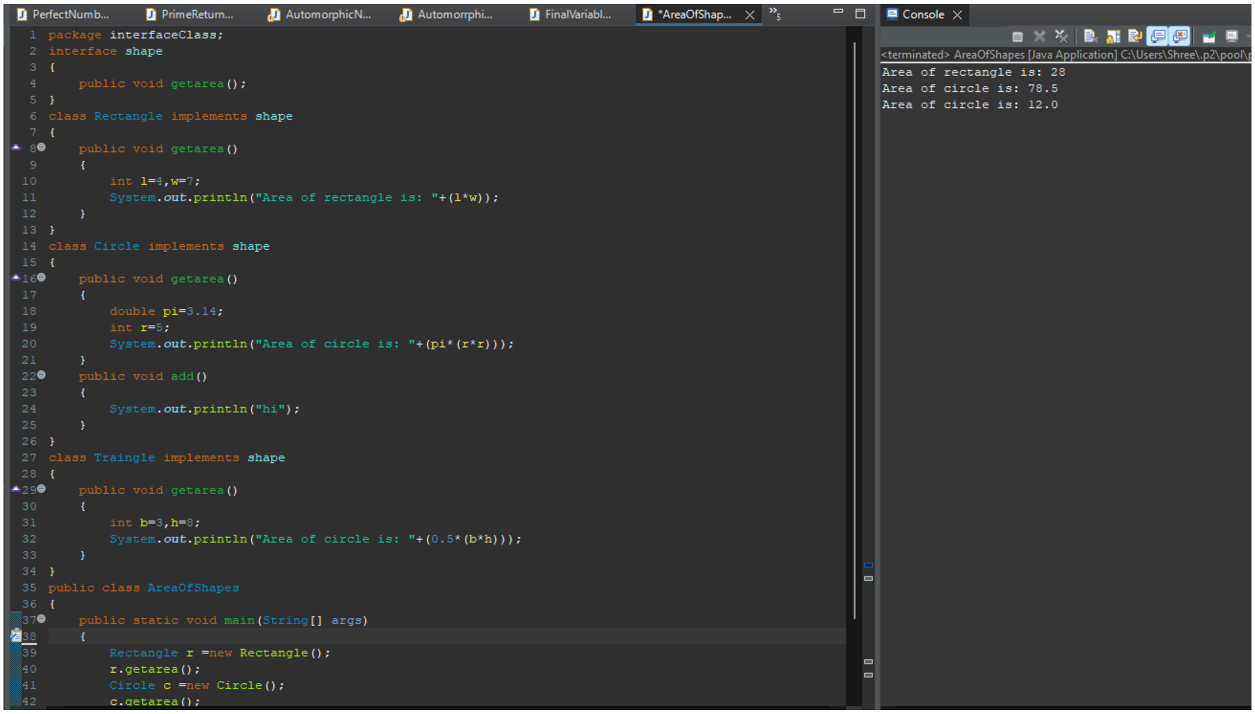


```
     }

}
class cube extends shape3D
{
     void calculateVolume()
     {
         int v=6;
         int result=v*v*v;
         System.out.println("volume of cube is:"+result);
     }
     void calculateSurfaceArea()
     {
         int sa=3;
         int result=6*sa*sa;
 System.out.println("surface of cube is:"+result);

     }

}
public class EmployeeAbstract
{

    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        shape3D s=new sphere();
        s.calculateVolume();
        s.calculateSurfaceArea();
        System.out.println("------------------");
        shape3D s1=new cube();
        s1.calculateVolume();
        s1.calculateSurfaceArea();
    }

}
```

```
<terminated> EmployeeAbstract [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclip
volume of sphere is:392.5
surface of sphere is:452.15999999999997
------------------
volume of cube is:216
surface of cube is:54
```

Interface:

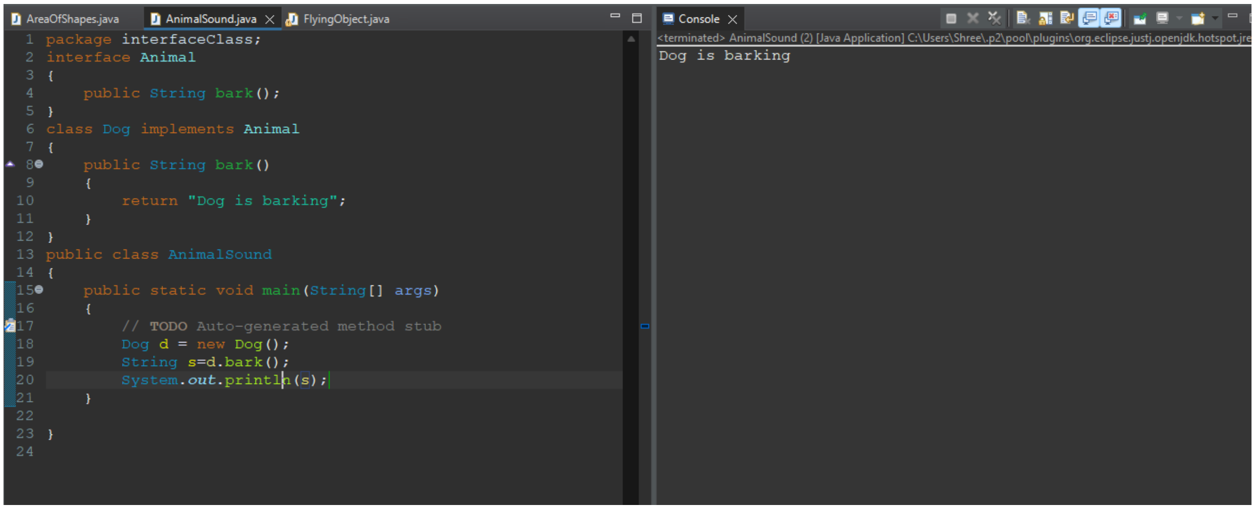
1. Write a Java program to create an interface Shape with the getArea() method. Create three classes Rectangle, Circle, and Triangle that implement the Shape interface. Implement the getArea() method for each of the three classes.



```java
package interfaceClass;
interface shape
{
    public void getarea();
}
class Rectangle implements shape
{
    public void getarea()
    {
        int l=4,w=7;
        System.out.println("Area of rectangle is: "+(l*w));
    }
}
class Circle implements shape
{
    public void getarea()
    {
        double pi=3.14;
        int r=5;
        System.out.println("Area of circle is: "+(pi*(r*r)));
    }
    public void add()
    {
        System.out.println("hi");
    }
}
class Traingle implements shape
{
    public void getarea()
    {
        int b=3,h=8;
        System.out.println("Area of circle is: "+(0.5*(b*h)));
    }
}
public class AreaOfShapes
{
    public static void main(String[] args)
    {
        Rectangle r =new Rectangle();
        r.getarea();
        Circle c =new Circle();
        c.getarea();
```

```
<terminated> AreaOfShapes [Java Application] C:\Users\Shree\.p2\pool\p
Area of rectangle is: 28
Area of circle is: 78.5
Area of circle is: 12.0
```

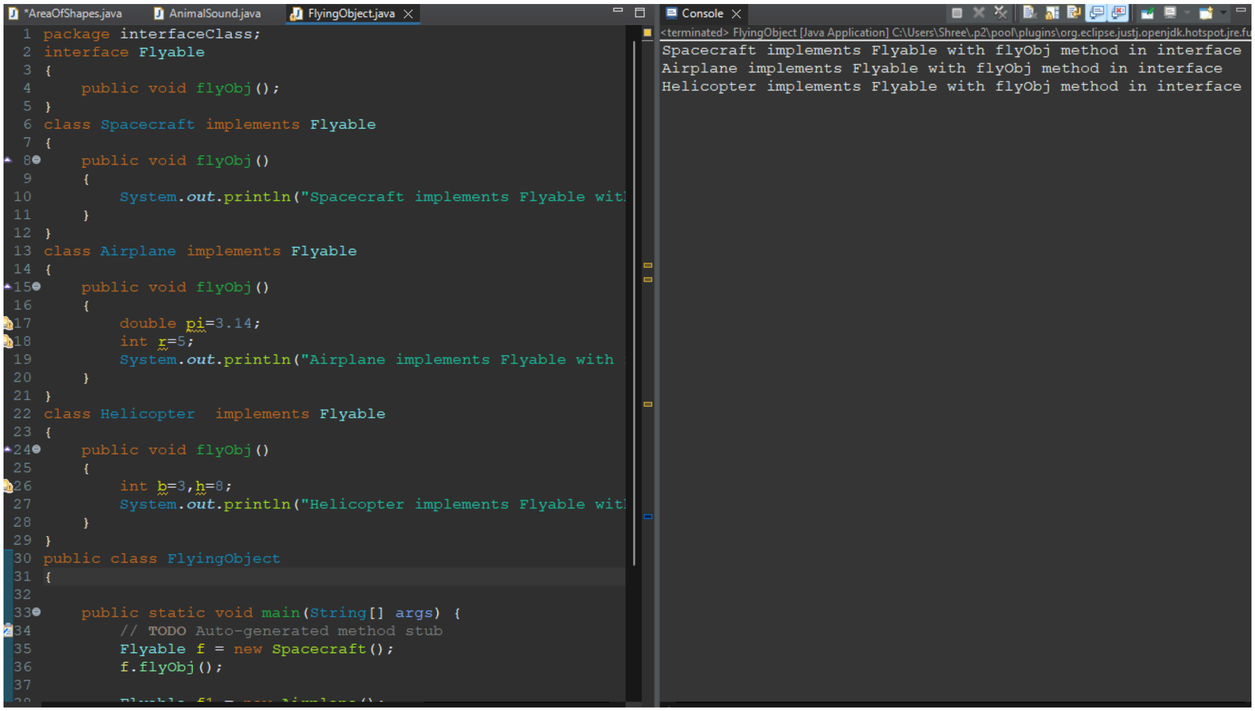
2. Write a Java program to create a Animal interface with a method called bark() that takes no arguments and returns void. Create a Dog class that implements Animal and overrides speak() to print "Dog is barking".



```java
package interfaceClass;
interface Animal
{
    public String bark();
}
class Dog implements Animal
{
    public String bark()
    {
        return "Dog is barking";
    }
}
public class AnimalSound
{
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Dog d = new Dog();
        String s=d.bark();
        System.out.println(s);
    }

}
```

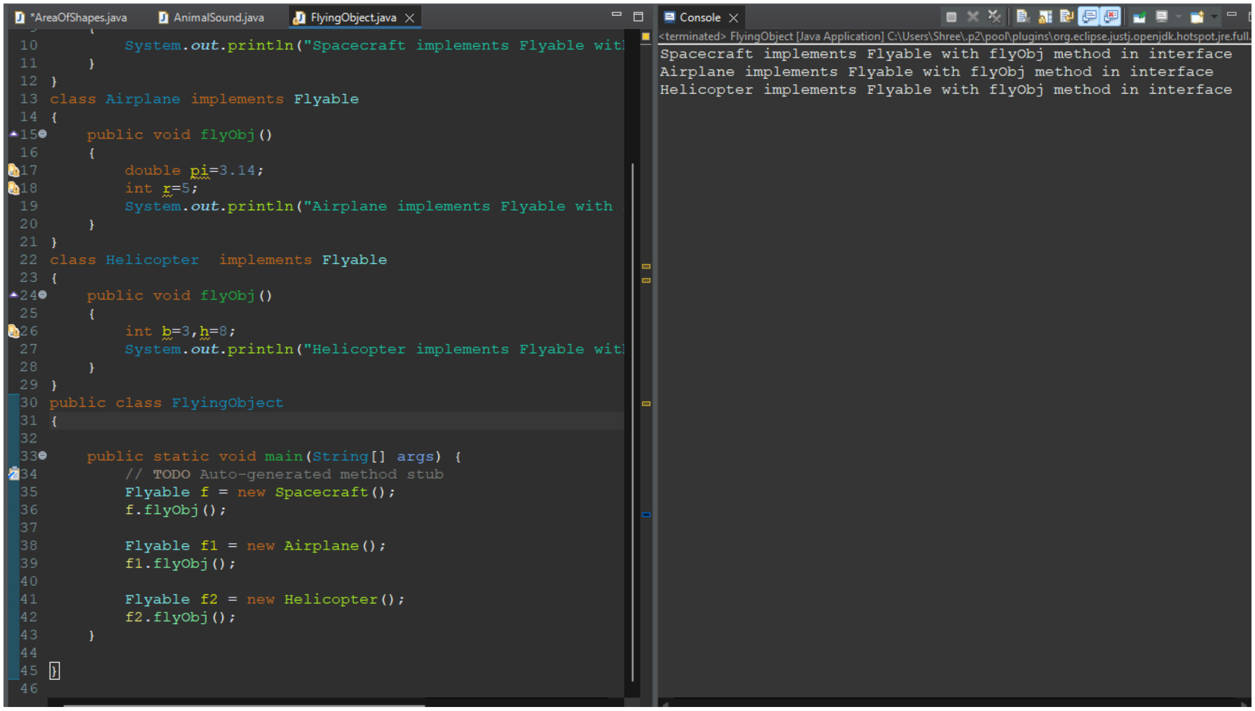
```
<terminated> AnimalSound (2) [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre
Dog is barking
```

3. Write a Java program to create an interface Flyable with a method called fly obj(). Create three classes Spacecraft, Airplane, and Helicopter that implement the Flyable interface. Implement the fly_obj() method for each of the three classes.



```java
package interfaceClass;
interface Flyable
{
    public void flyObj();
}
class Spacecraft implements Flyable
{
    public void flyObj()
    {
        System.out.println("Spacecraft implements Flyable wit
    }
}
class Airplane implements Flyable
{
    public void flyObj()
    {
        double pi=3.14;
        int r=5;
        System.out.println("Airplane implements Flyable with
    }
}
class Helicopter  implements Flyable
{
    public void flyObj()
    {
        int b=3,h=8;
        System.out.println("Helicopter implements Flyable wit
    }
}
public class FlyingObject
{

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Flyable f = new Spacecraft();
        f.flyObj();

        Flyable f1 = new Airplane();
```

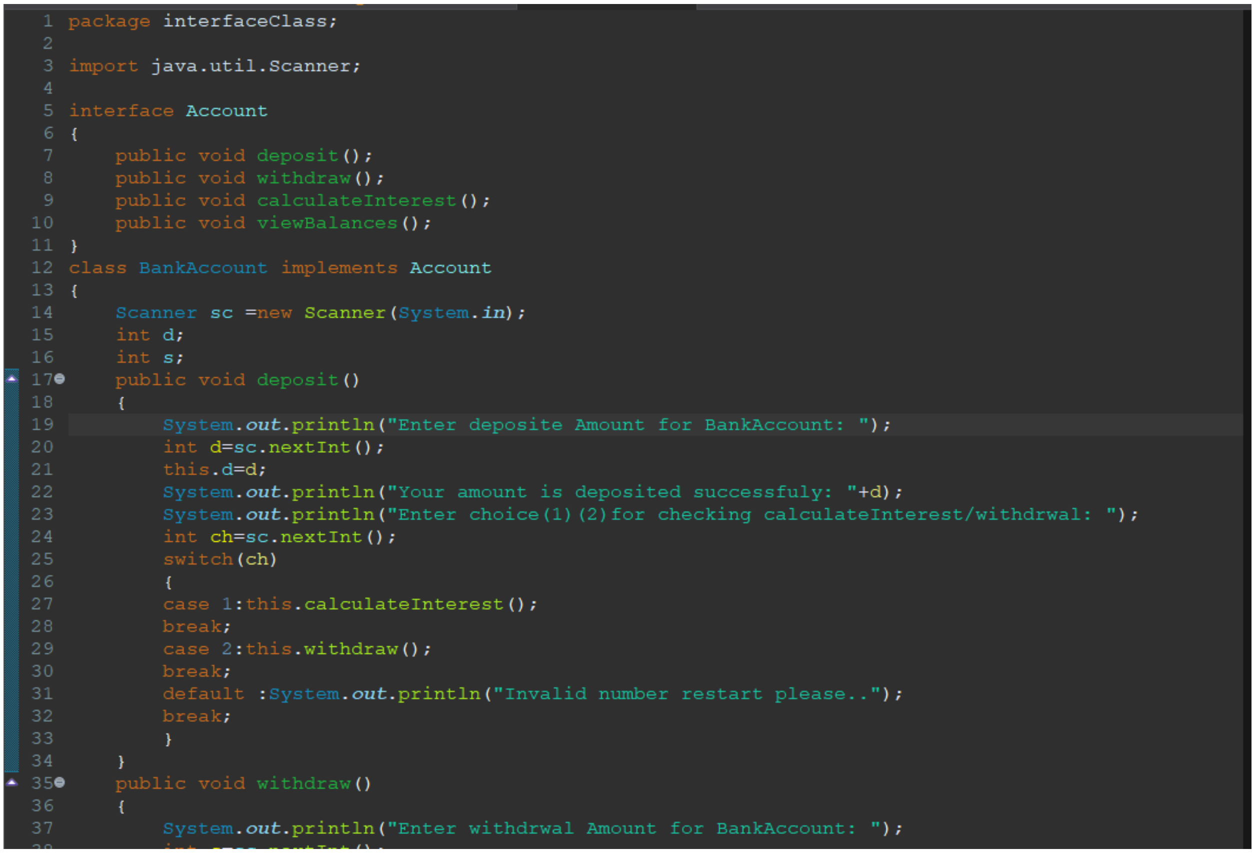
```
<terminated> FlyingObject [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.fu
Spacecraft implements Flyable with flyObj method in interface
Airplane implements Flyable with flyObj method in interface
Helicopter implements Flyable with flyObj method in interface
```



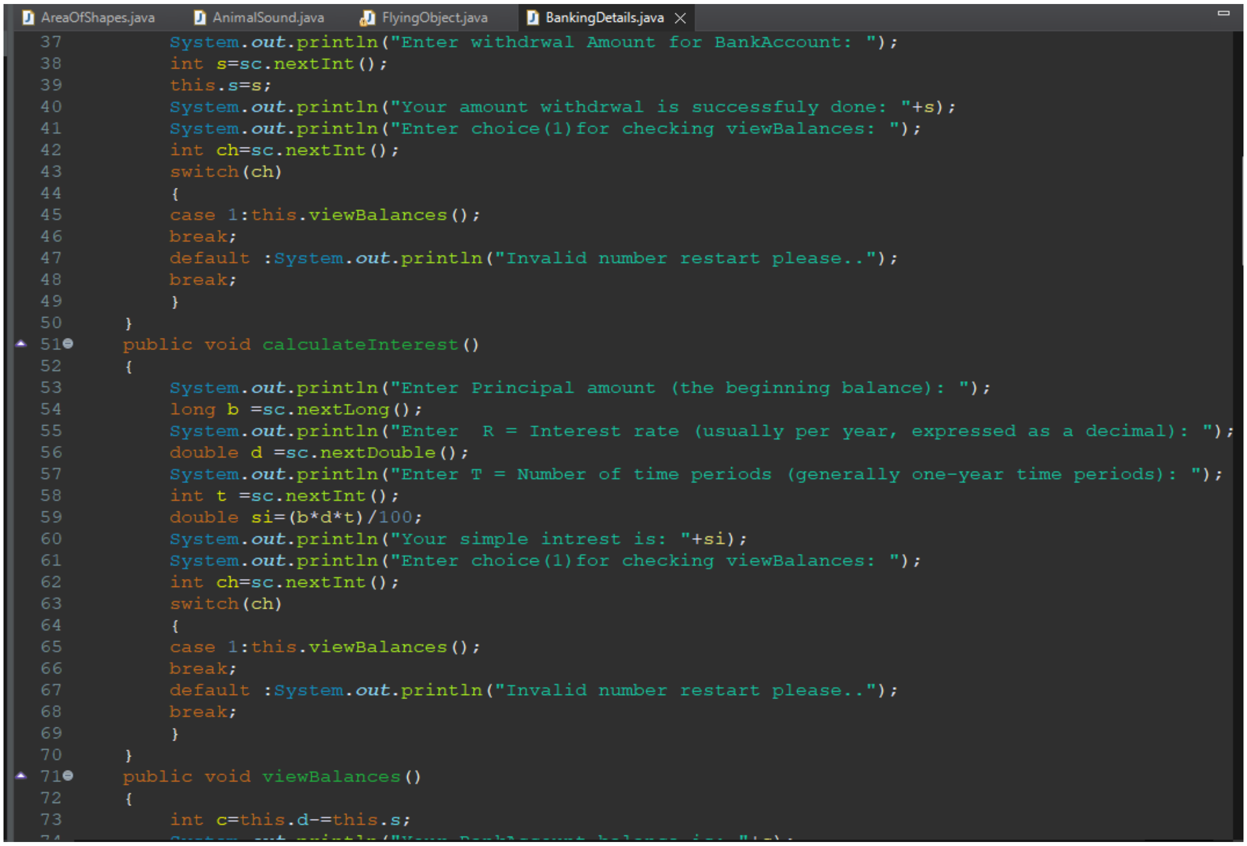
```java
        System.out.println("Spacecraft implements Flyable wit
    }
}
class Airplane implements Flyable
{
    public void flyObj()
    {
        double pi=3.14;
        int r=5;
        System.out.println("Airplane implements Flyable with
    }
}
class Helicopter  implements Flyable
{
    public void flyObj()
    {
        int b=3,h=8;
        System.out.println("Helicopter implements Flyable wit
    }
}
public class FlyingObject
{

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Flyable f = new Spacecraft();
        f.flyObj();

        Flyable f1 = new Airplane();
        f1.flyObj();

        Flyable f2 = new Helicopter();
        f2.flyObj();
    }

}
```

```
<terminated> FlyingObject [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full
Spacecraft implements Flyable with flyObj method in interface
Airplane implements Flyable with flyObj method in interface
Helicopter implements Flyable with flyObj method in interface
```

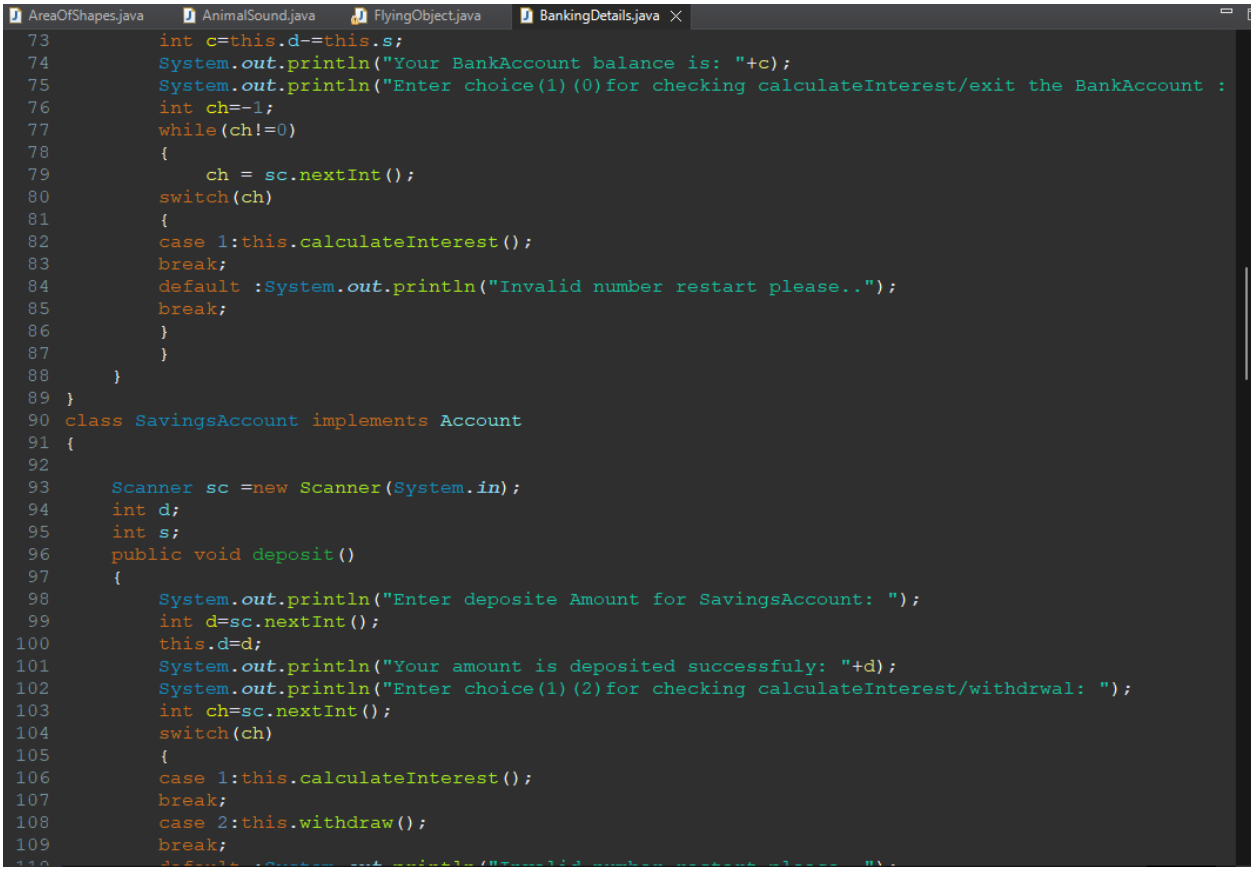
4. Write a Java programming to create a banking system with three classes - Bank, Account, SavingsAccount, and CurrentAccount. The bank should have a list of accounts and methods for adding them. Accounts should be an interface with methods to deposit, withdraw, calculate interest, and view balances. SavingsAccount and CurrentAccount should implement the Account interface and have their own unique methods.



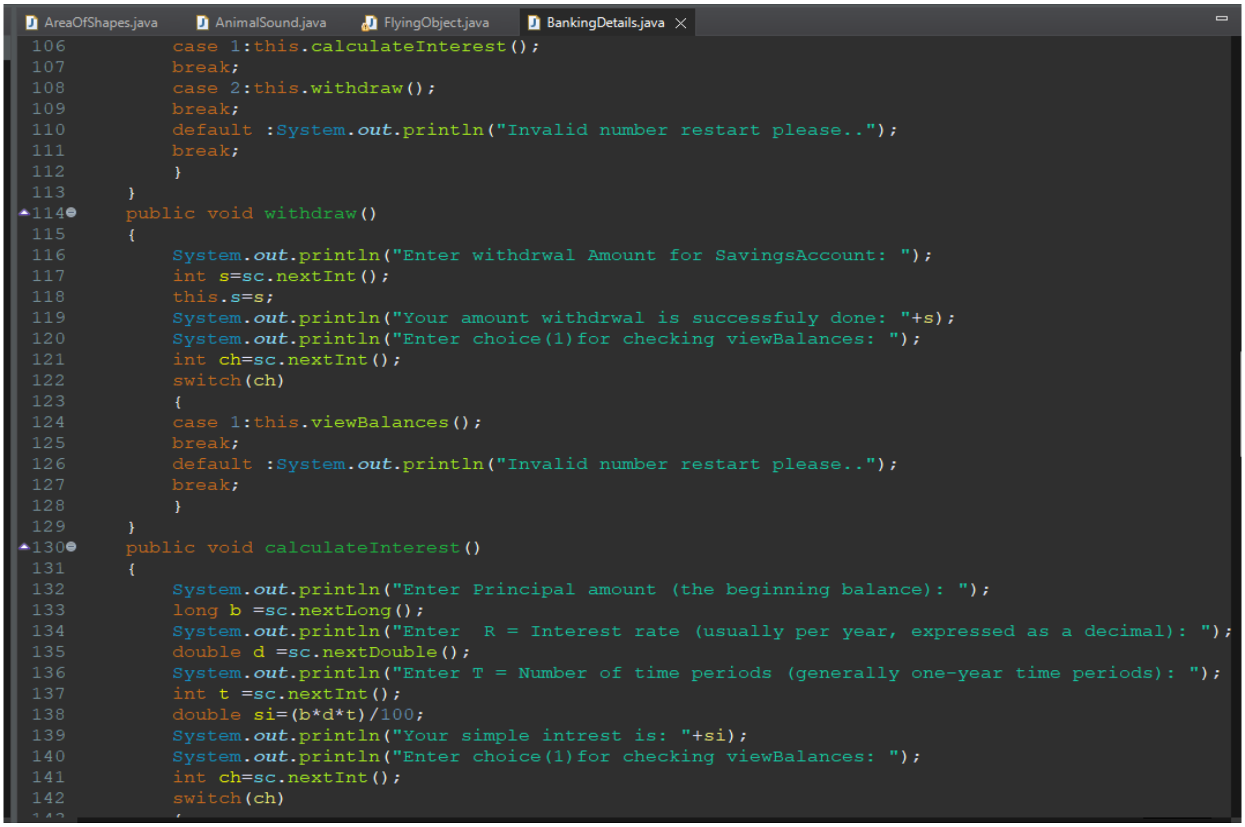
```java
package interfaceClass;

import java.util.Scanner;

interface Account
{
    public void deposit();
    public void withdraw();
    public void calculateInterest();
    public void viewBalances();
}
class BankAccount implements Account
{
    Scanner sc =new Scanner(System.in);
    int d;
    int s;
    public void deposit()
    {
        System.out.println("Enter deposite Amount for BankAccount: ");
        int d=sc.nextInt();
        this.d=d;
        System.out.println("Your amount is deposited successfuly: "+d);
        System.out.println("Enter choice(1)(2)for checking calculateInterest/withdrwal: ");
        int ch=sc.nextInt();
        switch(ch)
        {
        case 1:this.calculateInterest();
        break;
        case 2:this.withdraw();
        break;
        default :System.out.println("Invalid number restart please..");
        break;
        }
    }
    public void withdraw()
    {
        System.out.println("Enter withdrwal Amount for BankAccount: ");
```



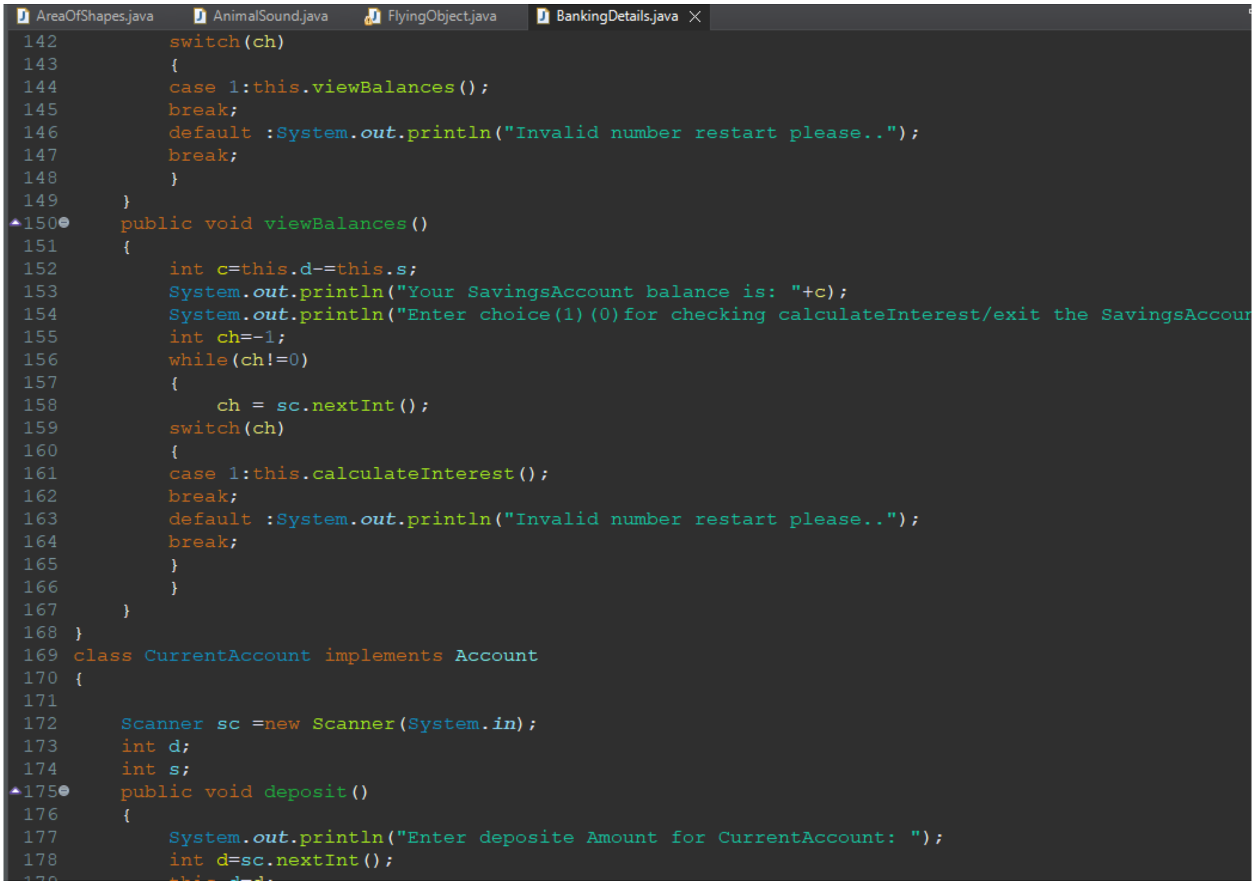
```
        System.out.println("Enter withdrwal Amount for BankAccount: ");
        int s=sc.nextInt();
        this.s=s;
        System.out.println("Your amount withdrwal is successfuly done: "+s);
        System.out.println("Enter choice(1)for checking viewBalances: ");
        int ch=sc.nextInt();
        switch(ch)
        {
        case 1:this.viewBalances();
        break;
        default :System.out.println("Invalid number restart please..");
        break;
        }
    }
    public void calculateInterest()
    {
        System.out.println("Enter Principal amount (the beginning balance): ");
        long b =sc.nextLong();
        System.out.println("Enter  R = Interest rate (usually per year, expressed as a decimal): ");
        double d =sc.nextDouble();
        System.out.println("Enter T = Number of time periods (generally one-year time periods): ");
        int t =sc.nextInt();
        double si=(b*d*t)/100;
        System.out.println("Your simple intrest is: "+si);
        System.out.println("Enter choice(1)for checking viewBalances: ");
        int ch=sc.nextInt();
        switch(ch)
        {
        case 1:this.viewBalances();
        break;
        default :System.out.println("Invalid number restart please..");
        break;
        }
    }
    public void viewBalances()
    {
        int c=this.d-=this.s;
        System.out.println("Your BankAccount balance is: "+c);
```



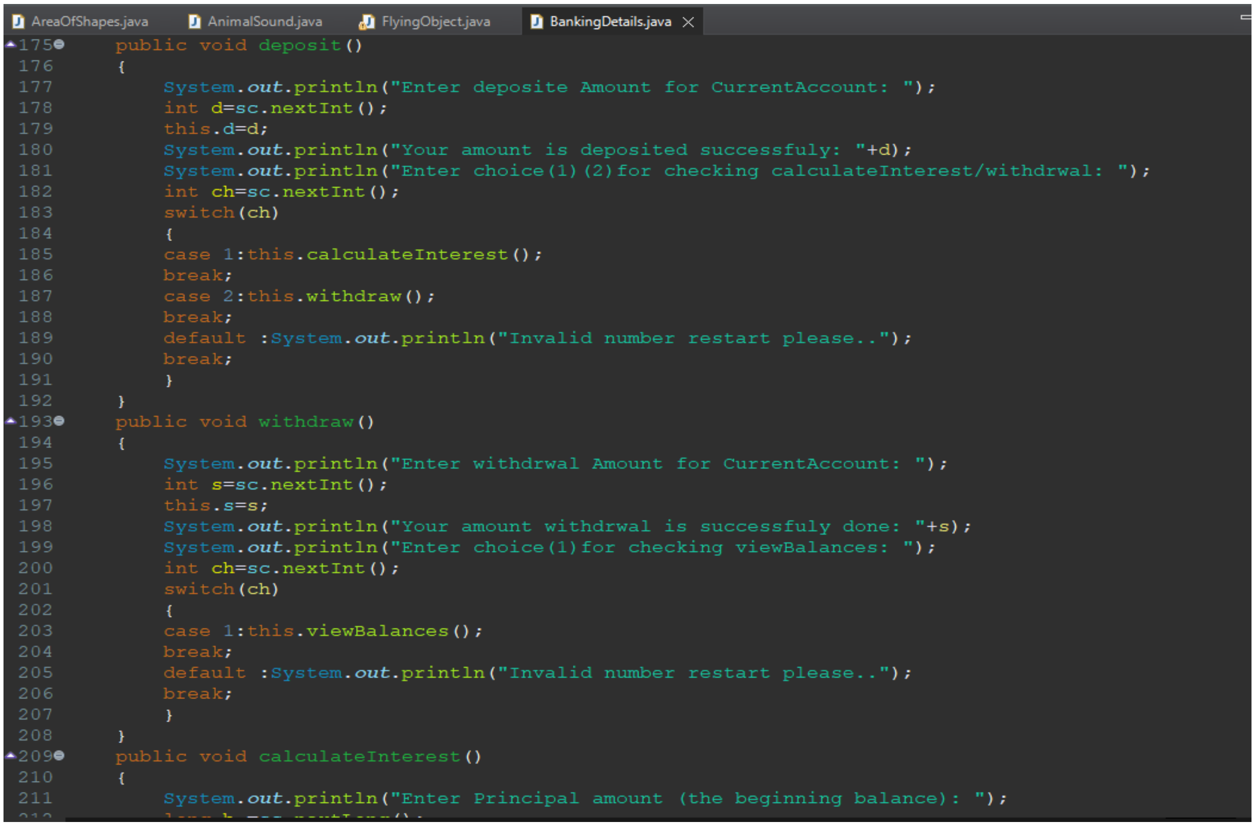
```
        int c=this.d-=this.s;
        System.out.println("Your BankAccount balance is: "+c);
        System.out.println("Enter choice(1)(0)for checking calculateInterest/exit the BankAccount :
        int ch=-1;
        while(ch!=0)
        {
            ch = sc.nextInt();
        switch(ch)
        {
        case 1:this.calculateInterest();
        break;
        default :System.out.println("Invalid number restart please..");
        break;
        }
        }
    }
}
class SavingsAccount implements Account
{

    Scanner sc =new Scanner(System.in);
    int d;
    int s;
    public void deposit()
    {
        System.out.println("Enter deposite Amount for SavingsAccount: ");
        int d=sc.nextInt();
        this.d=d;
        System.out.println("Your amount is deposited successfuly: "+d);
        System.out.println("Enter choice(1)(2)for checking calculateInterest/withdrwal: ");
        int ch=sc.nextInt();
        switch(ch)
        {
        case 1:this.calculateInterest();
        break;
        case 2:this.withdraw();
        break;
        default :System.out.println("Invalid number restart please..");
```



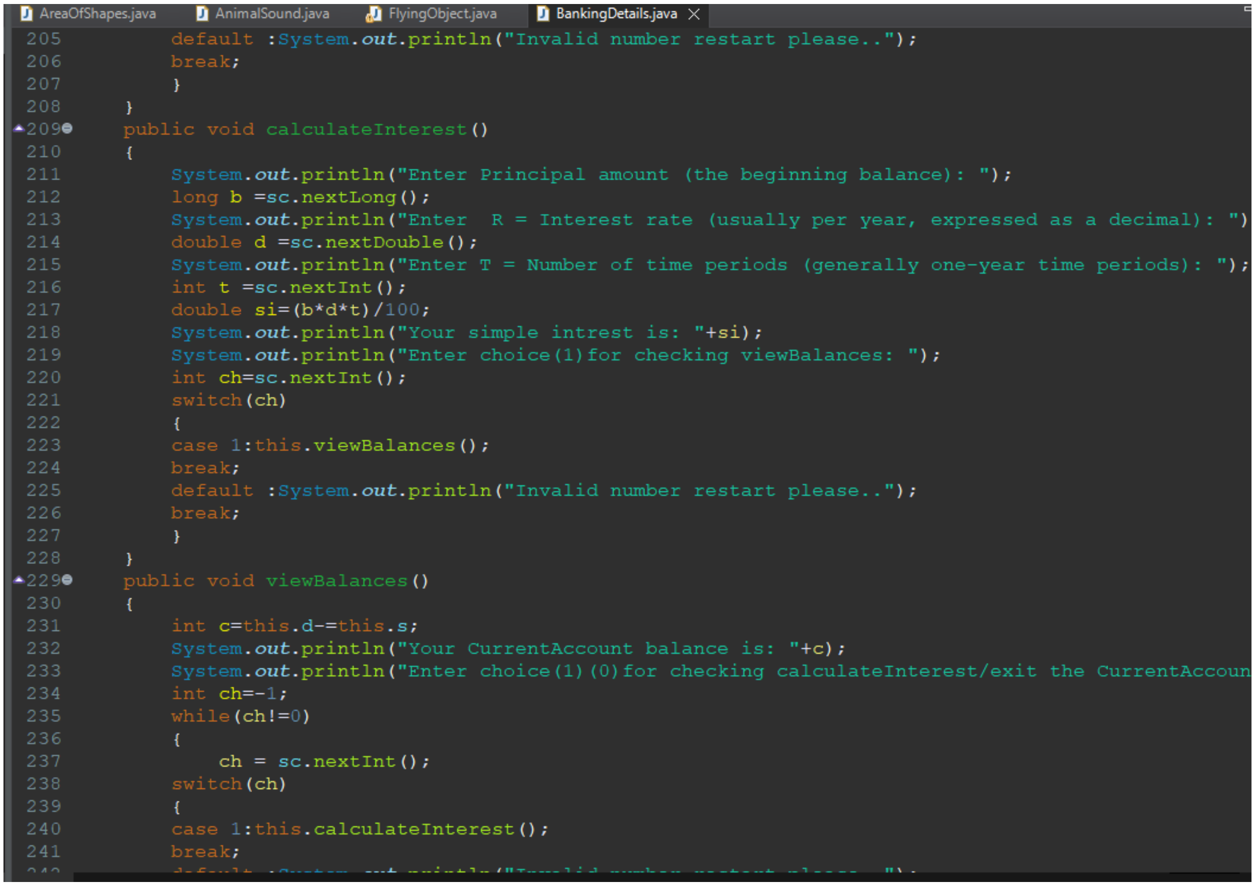
```java
        case 1:this.calculateInterest();
        break;
        case 2:this.withdraw();
        break;
        default :System.out.println("Invalid number restart please..");
        break;
        }
    }
    public void withdraw()
    {
        System.out.println("Enter withdrwal Amount for SavingsAccount: ");
        int s=sc.nextInt();
        this.s=s;
        System.out.println("Your amount withdrwal is successfuly done: "+s);
        System.out.println("Enter choice(1)for checking viewBalances: ");
        int ch=sc.nextInt();
        switch(ch)
        {
        case 1:this.viewBalances();
        break;
        default :System.out.println("Invalid number restart please..");
        break;
        }
    }
    public void calculateInterest()
    {
        System.out.println("Enter Principal amount (the beginning balance): ");
        long b =sc.nextLong();
        System.out.println("Enter  R = Interest rate (usually per year, expressed as a decimal): ");
        double d =sc.nextDouble();
        System.out.println("Enter T = Number of time periods (generally one-year time periods): ");
        int t =sc.nextInt();
        double si=(b*d*t)/100;
        System.out.println("Your simple intrest is: "+si);
        System.out.println("Enter choice(1)for checking viewBalances: ");
        int ch=sc.nextInt();
        switch(ch)
```



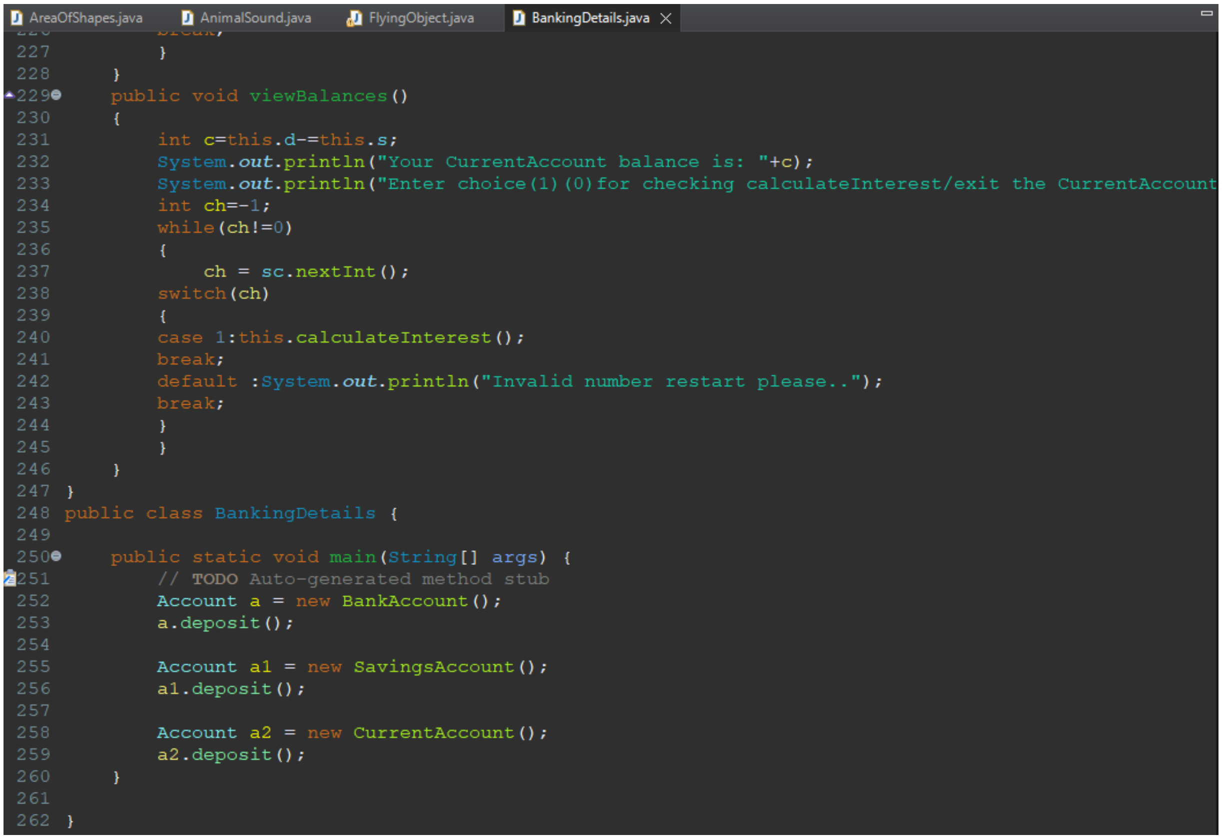
```java
        switch(ch)
        {
        case 1:this.viewBalances();
        break;
        default :System.out.println("Invalid number restart please..");
        break;
        }
    }
    public void viewBalances()
    {
        int c=this.d-=this.s;
        System.out.println("Your SavingsAccount balance is: "+c);
        System.out.println("Enter choice(1)(0)for checking calculateInterest/exit the SavingsAccour
        int ch=-1;
        while(ch!=0)
        {
            ch = sc.nextInt();
        switch(ch)
        {
        case 1:this.calculateInterest();
        break;
        default :System.out.println("Invalid number restart please..");
        break;
        }
        }
    }
}
class CurrentAccount implements Account
{

    Scanner sc =new Scanner(System.in);
    int d;
    int s;
    public void deposit()
    {
        System.out.println("Enter deposite Amount for CurrentAccount: ");
        int d=sc.nextInt();
```



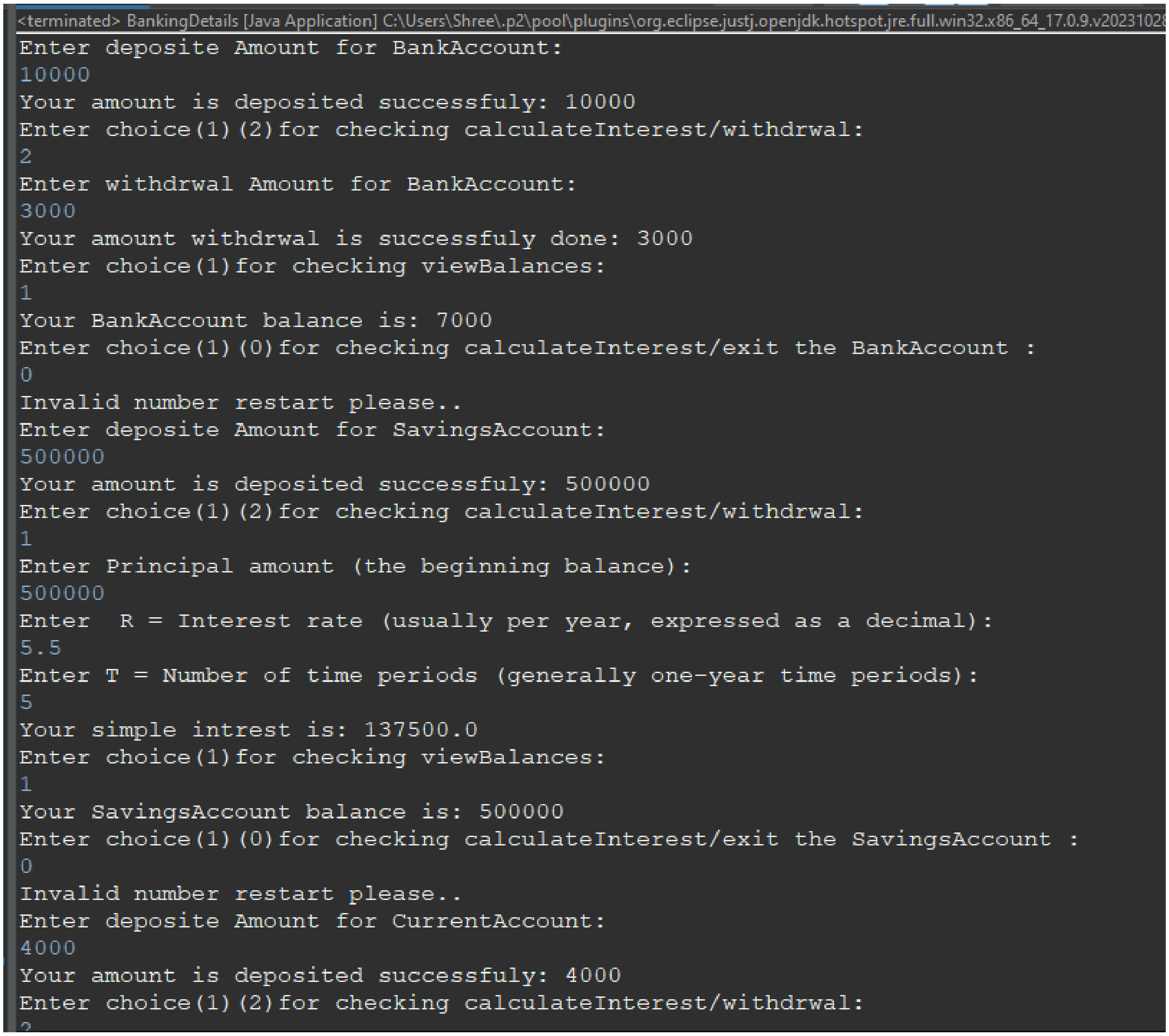
```java
    public void deposit()
    {
        System.out.println("Enter deposite Amount for CurrentAccount: ");
        int d=sc.nextInt();
        this.d=d;
        System.out.println("Your amount is deposited successfuly: "+d);
        System.out.println("Enter choice(1)(2)for checking calculateInterest/withdrwal: ");
        int ch=sc.nextInt();
        switch(ch)
        {
        case 1:this.calculateInterest();
        break;
        case 2:this.withdraw();
        break;
        default :System.out.println("Invalid number restart please..");
        break;
        }
    }
    public void withdraw()
    {
        System.out.println("Enter withdrwal Amount for CurrentAccount: ");
        int s=sc.nextInt();
        this.s=s;
        System.out.println("Your amount withdrwal is successfuly done: "+s);
        System.out.println("Enter choice(1)for checking viewBalances: ");
        int ch=sc.nextInt();
        switch(ch)
        {
        case 1:this.viewBalances();
        break;
        default :System.out.println("Invalid number restart please..");
        break;
        }
    }
    public void calculateInterest()
    {
        System.out.println("Enter Principal amount (the beginning balance): ");
        long b =sc.nextLong();
```



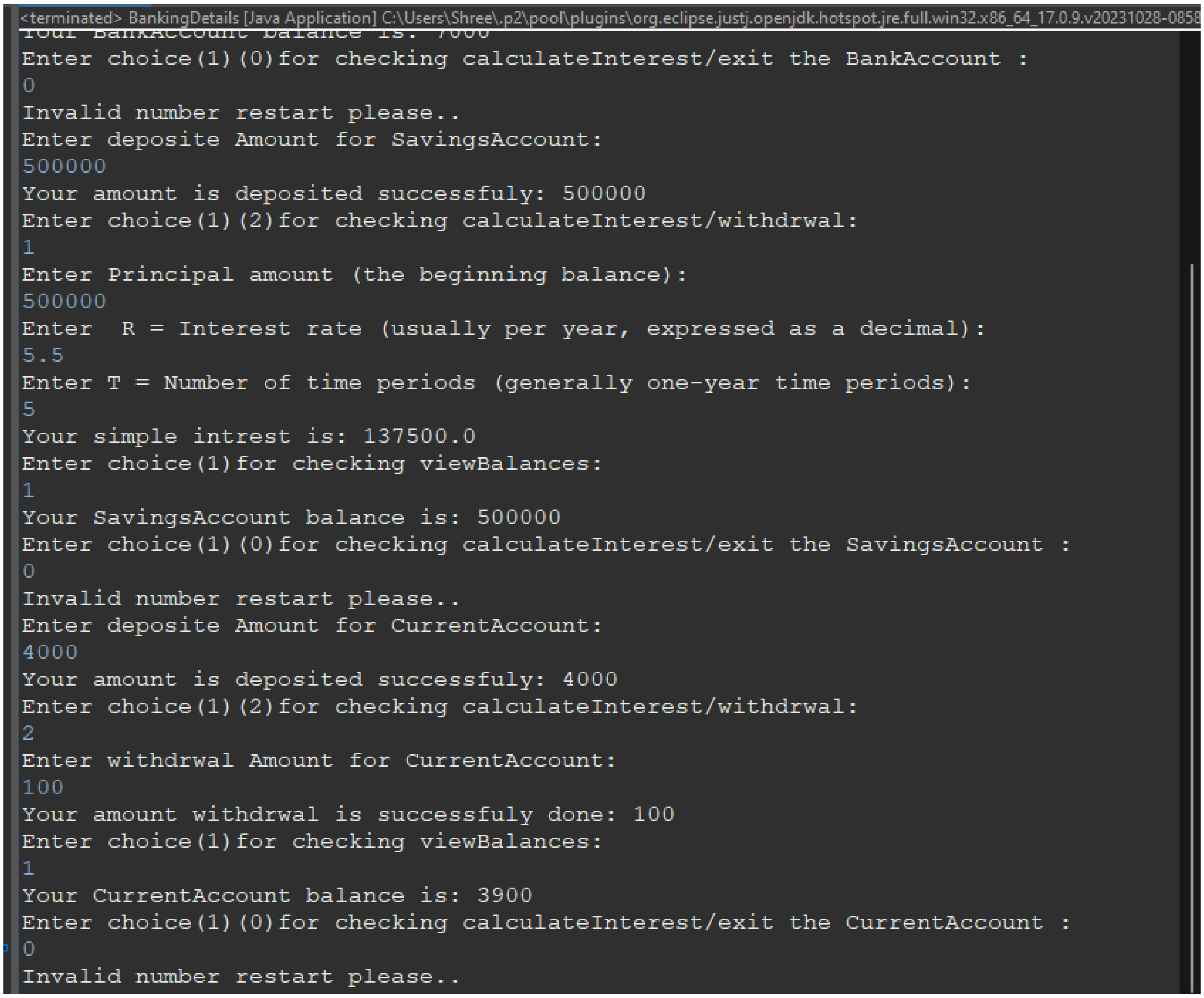
```java
        default :System.out.println("Invalid number restart please..");
        break;
        }
    }
    public void calculateInterest()
    {
        System.out.println("Enter Principal amount (the beginning balance): ");
        long b =sc.nextLong();
        System.out.println("Enter  R = Interest rate (usually per year, expressed as a decimal): ")
        double d =sc.nextDouble();
        System.out.println("Enter T = Number of time periods (generally one-year time periods): ");
        int t =sc.nextInt();
        double si=(b*d*t)/100;
        System.out.println("Your simple intrest is: "+si);
        System.out.println("Enter choice(1)for checking viewBalances: ");
        int ch=sc.nextInt();
        switch(ch)
        {
        case 1:this.viewBalances();
        break;
        default :System.out.println("Invalid number restart please..");
        break;
        }
    }
    public void viewBalances()
    {
        int c=this.d-=this.s;
        System.out.println("Your CurrentAccount balance is: "+c);
        System.out.println("Enter choice(1)(0)for checking calculateInterest/exit the CurrentAccoun
        int ch=-1;
        while(ch!=0)
        {
            ch = sc.nextInt();
        switch(ch)
        {
        case 1:this.calculateInterest();
        break;
        default :System.out.println("Invalid number restart please..");
```



```
        break;
        }
    }
    public void viewBalances()
    {
        int c=this.d-=this.s;
        System.out.println("Your CurrentAccount balance is: "+c);
        System.out.println("Enter choice(1)(0)for checking calculateInterest/exit the CurrentAccount
        int ch=-1;
        while(ch!=0)
        {
            ch = sc.nextInt();
        switch(ch)
        {
        case 1:this.calculateInterest();
        break;
        default :System.out.println("Invalid number restart please..");
        break;
        }
        }
    }
}
public class BankingDetails {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Account a = new BankAccount();
        a.deposit();

        Account a1 = new SavingsAccount();
        a1.deposit();

        Account a2 = new CurrentAccount();
        a2.deposit();
    }

}
```

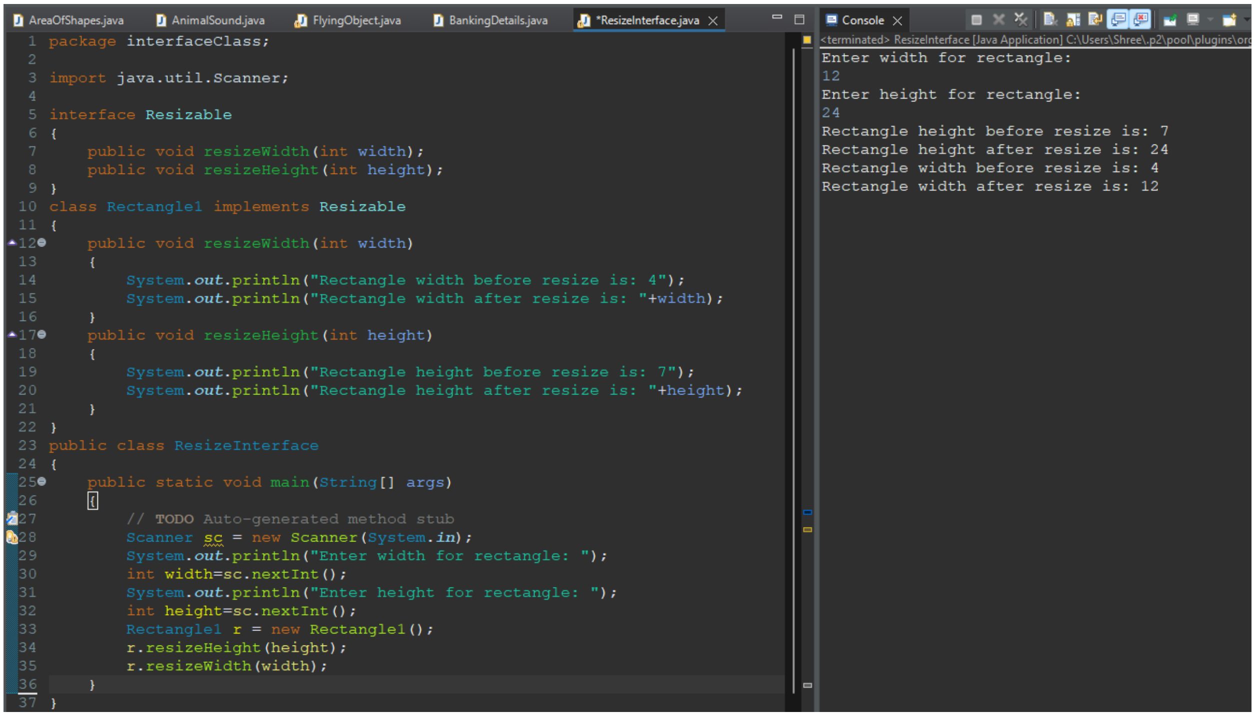


```
<terminated> BankingDetails [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028
Enter deposite Amount for BankAccount:
10000
Your amount is deposited successfuly: 10000
Enter choice(1)(2)for checking calculateInterest/withdrwal:
2
Enter withdrwal Amount for BankAccount:
3000
Your amount withdrwal is successfuly done: 3000
Enter choice(1)for checking viewBalances:
1
Your BankAccount balance is: 7000
Enter choice(1)(0)for checking calculateInterest/exit the BankAccount :
0
Invalid number restart please..
Enter deposite Amount for SavingsAccount:
500000
Your amount is deposited successfuly: 500000
Enter choice(1)(2)for checking calculateInterest/withdrwal:
1
Enter Principal amount (the beginning balance):
500000
Enter  R = Interest rate (usually per year, expressed as a decimal):
5.5
Enter T = Number of time periods (generally one-year time periods):
5
Your simple intrest is: 137500.0
Enter choice(1)for checking viewBalances:
1
Your SavingsAccount balance is: 500000
Enter choice(1)(0)for checking calculateInterest/exit the SavingsAccount :
0
Invalid number restart please..
Enter deposite Amount for CurrentAccount:
4000
Your amount is deposited successfuly: 4000
Enter choice(1)(2)for checking calculateInterest/withdrwal:
2
```



```
<terminated> BankingDetails [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0858
Your BankAccount balance is: 7000
Enter choice(1)(0)for checking calculateInterest/exit the BankAccount :
0
Invalid number restart please..
Enter deposite Amount for SavingsAccount:
500000
Your amount is deposited successfuly: 500000
Enter choice(1)(2)for checking calculateInterest/withdrwal:
1
Enter Principal amount (the beginning balance):
500000
Enter  R = Interest rate (usually per year, expressed as a decimal):
5.5
Enter T = Number of time periods (generally one-year time periods):
5
Your simple intrest is: 137500.0
Enter choice(1)for checking viewBalances:
1
Your SavingsAccount balance is: 500000
Enter choice(1)(0)for checking calculateInterest/exit the SavingsAccount :
0
Invalid number restart please..
Enter deposite Amount for CurrentAccount:
4000
Your amount is deposited successfuly: 4000
Enter choice(1)(2)for checking calculateInterest/withdrwal:
2
Enter withdrwal Amount for CurrentAccount:
100
Your amount withdrwal is successfuly done: 100
Enter choice(1)for checking viewBalances:
1
Your CurrentAccount balance is: 3900
Enter choice(1)(0)for checking calculateInterest/exit the CurrentAccount :
0
Invalid number restart please..
```

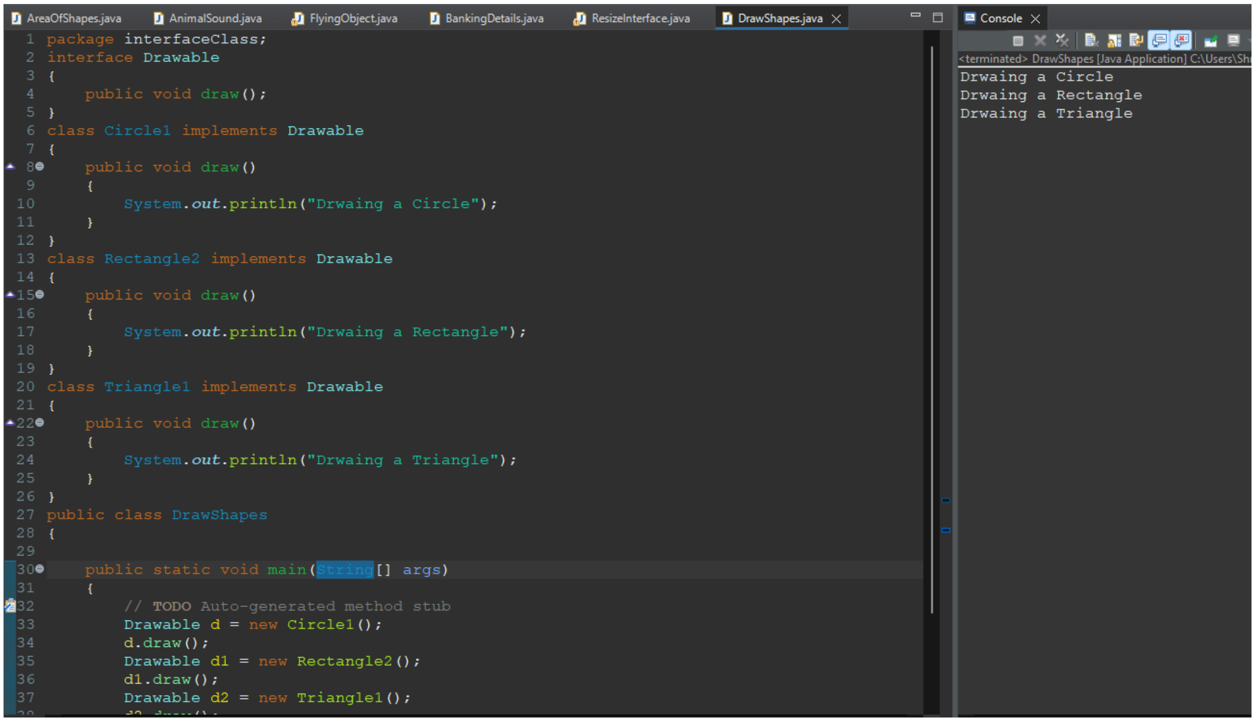
5. Write a Java program to create an interface Resizable with methods resizeWidth(int width) and resizeHeight(int height) that allow an object to be resized. Create a class Rectangle that implements the Resizable interface and implements the resize methods.



```java
package interfaceClass;

import java.util.Scanner;

interface Resizable
{
    public void resizeWidth(int width);
    public void resizeHeight(int height);
}
class Rectangle1 implements Resizable
{
    public void resizeWidth(int width)
    {
        System.out.println("Rectangle width before resize is: 4");
        System.out.println("Rectangle width after resize is: "+width);
    }
    public void resizeHeight(int height)
    {
        System.out.println("Rectangle height before resize is: 7");
        System.out.println("Rectangle height after resize is: "+height);
    }
}
public class ResizeInterface
{
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter width for rectangle: ");
        int width=sc.nextInt();
        System.out.println("Enter height for rectangle: ");
        int height=sc.nextInt();
        Rectangle1 r = new Rectangle1();
        r.resizeHeight(height);
        r.resizeWidth(width);
    }
}
```

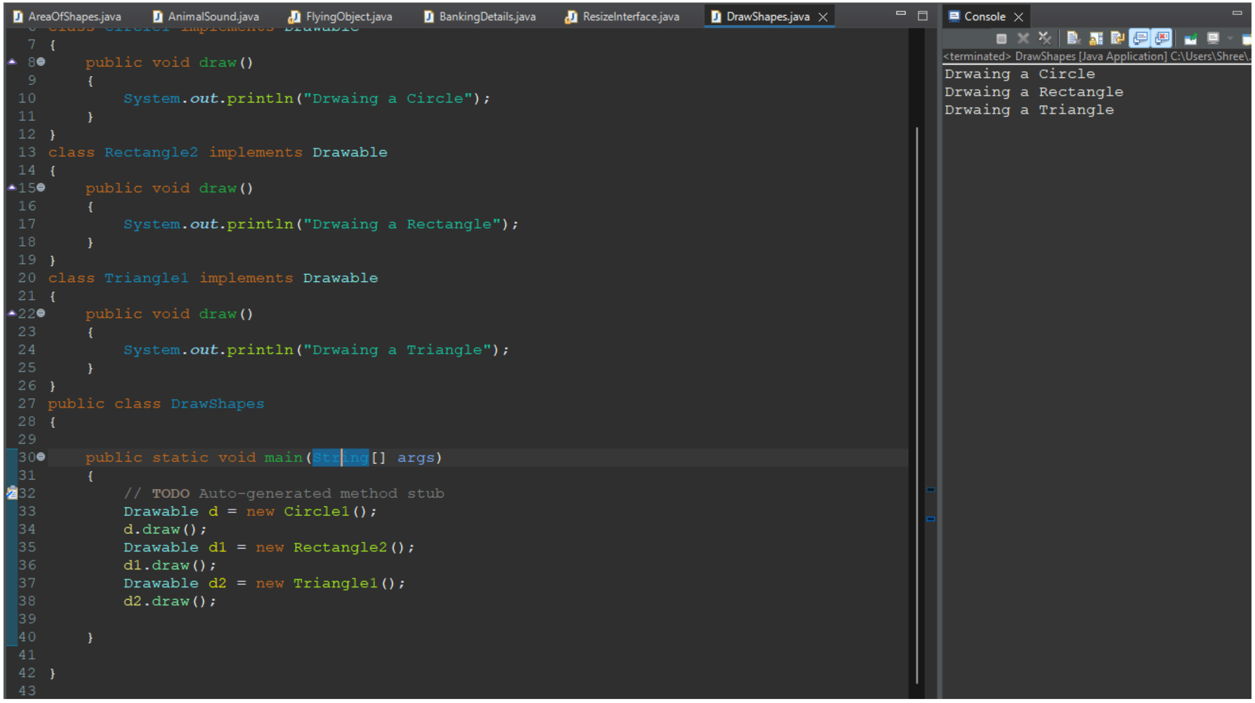
```
<terminated> ResizeInterface [Java Application] C:\Users\Shree\.p2\pool\plugins\org
Enter width for rectangle:
12
Enter height for rectangle:
24
Rectangle height before resize is: 7
Rectangle height after resize is: 24
Rectangle width before resize is: 4
Rectangle width after resize is: 12
```

6. Write a Java program to create an interface Drawable with a method draw() that takes no arguments and returns void. Create three classes Circle, Rectangle, and Triangle that implement the Drawable interface and override the draw() method to draw their respective shapes.



```java
package interfaceClass;
interface Drawable
{
    public void draw();
}
class Circle1 implements Drawable
{
    public void draw()
    {
        System.out.println("Drwaing a Circle");
    }
}
class Rectangle2 implements Drawable
{
    public void draw()
    {
        System.out.println("Drwaing a Rectangle");
    }
}
class Triangle1 implements Drawable
{
    public void draw()
    {
        System.out.println("Drwaing a Triangle");
    }
}
public class DrawShapes
{

    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Drawable d = new Circle1();
        d.draw();
        Drawable d1 = new Rectangle2();
        d1.draw();
        Drawable d2 = new Triangle1();
```

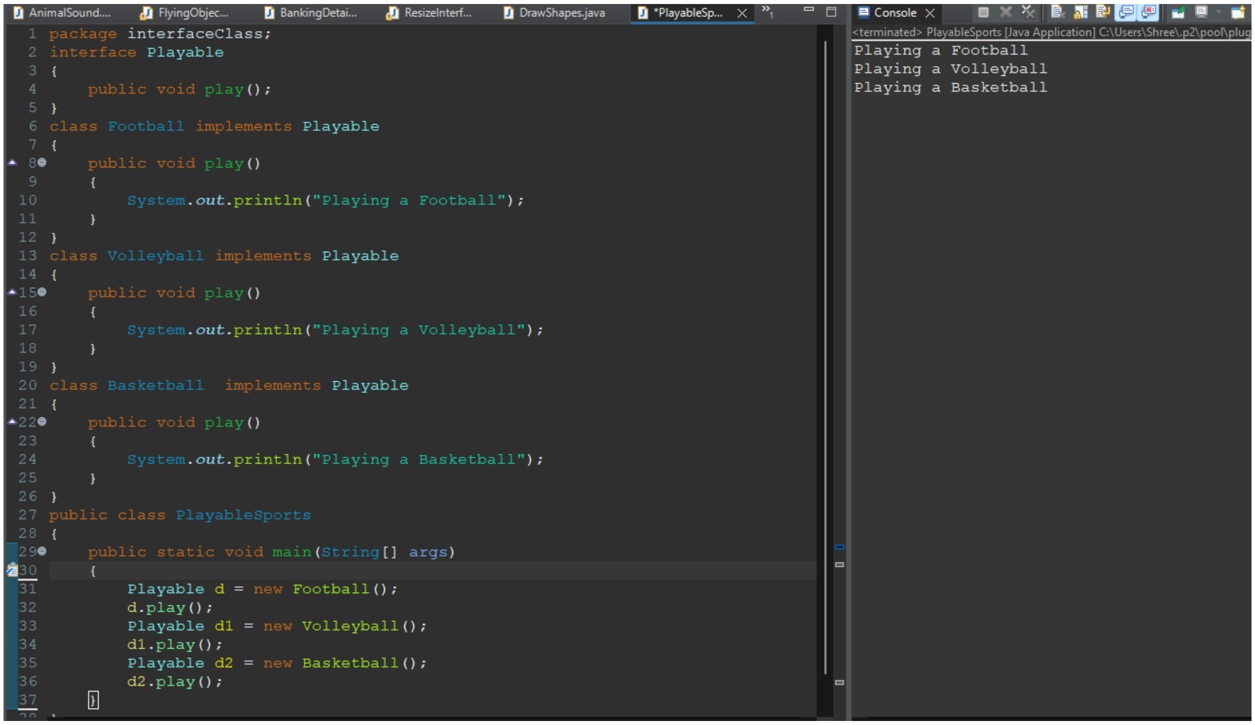
```
Drwaing a Circle
Drwaing a Rectangle
Drwaing a Triangle
```



```java
{
    public void draw()
    {
        System.out.println("Drwaing a Circle");
    }
}
class Rectangle2 implements Drawable
{
    public void draw()
    {
        System.out.println("Drwaing a Rectangle");
    }
}
class Triangle1 implements Drawable
{
    public void draw()
    {
        System.out.println("Drwaing a Triangle");
    }
}
public class DrawShapes
{

    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Drawable d = new Circle1();
        d.draw();
        Drawable d1 = new Rectangle2();
        d1.draw();
        Drawable d2 = new Triangle1();
        d2.draw();

    }

}

```

```
Drwaing a Circle
Drwaing a Rectangle
Drwaing a Triangle
```

7. Write a Java program to create an interface Playable with a method play() that takes no arguments and returns void. Create three classes Football, Volleyball, and Basketball that implement the Playable interface and override the play() method to play the respective sports.



```
package interfaceClass;
interface Playable
{
    public void play();
}
class Football implements Playable
{
    public void play()
    {
        System.out.println("Playing a Football");
    }
}
class Volleyball implements Playable
{
    public void play()
    {
        System.out.println("Playing a Volleyball");
    }
}
class Basketball  implements Playable
{
    public void play()
    {
        System.out.println("Playing a Basketball");
    }
}
public class PlayableSports
{
    public static void main(String[] args)
    {
        Playable d = new Football();
        d.play();
        Playable d1 = new Volleyball();
        d1.play();
        Playable d2 = new Basketball();
        d2.play();
    }
}
```

<terminated> PlayableSports [Java Application] C:\Users\Shree\.p2\pool\plug

```
Playing a Football
Playing a Volleyball
Playing a Basketball
```