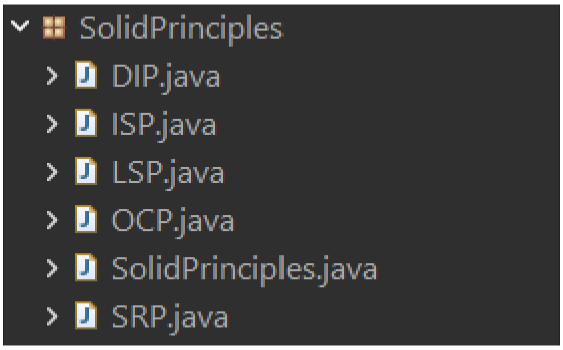
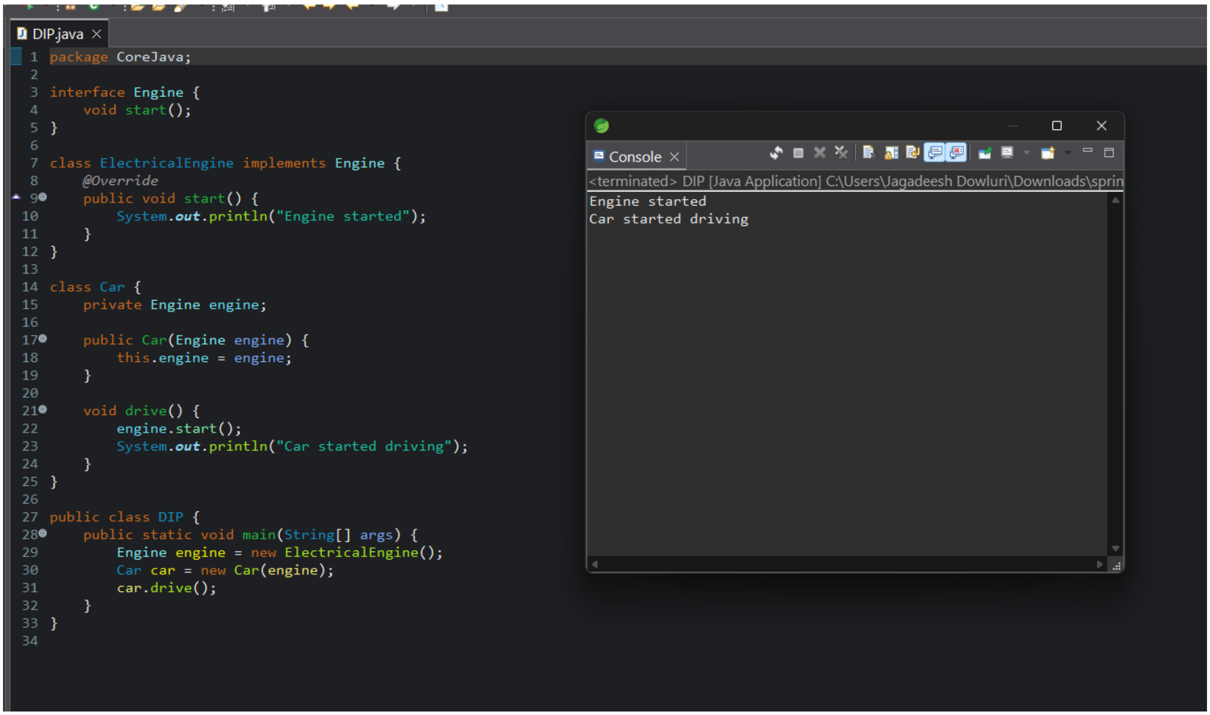
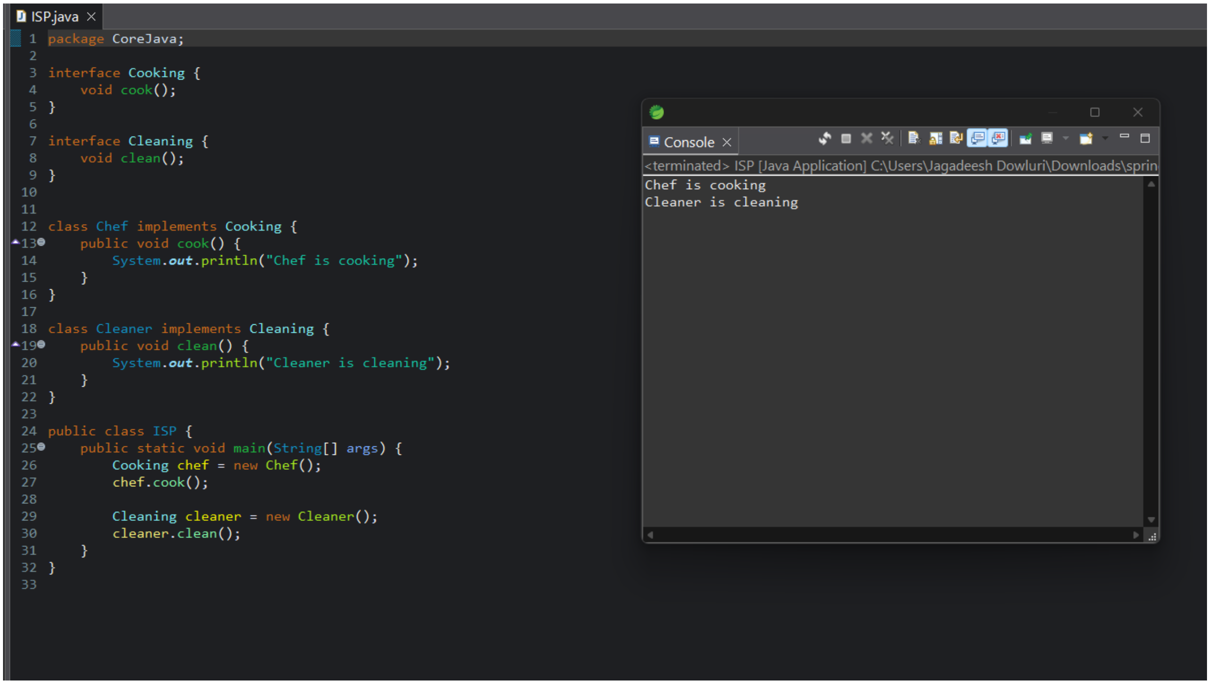
Solid Principles





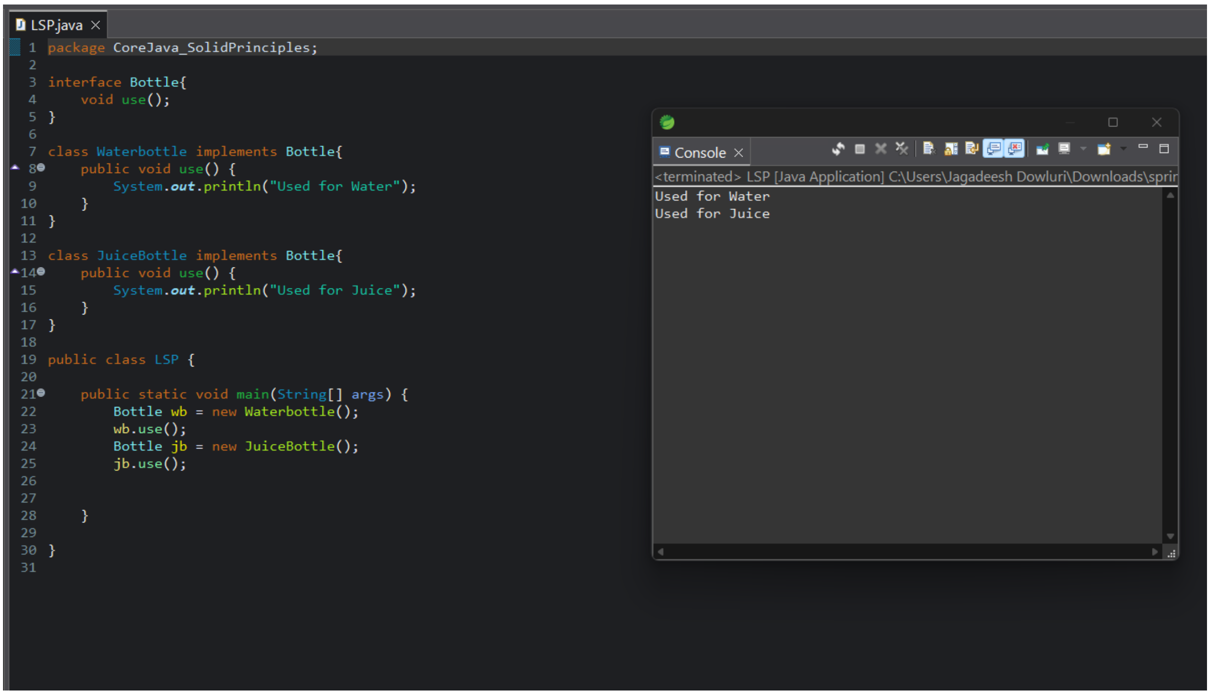
```java
package CoreJava;

interface Engine {
    void start();
}

class ElectricalEngine implements Engine {
    @Override
    public void start() {
        System.out.println("Engine started");
    }
}

class Car {
    private Engine engine;

    public Car(Engine engine) {
        this.engine = engine;
    }

    void drive() {
        engine.start();
        System.out.println("Car started driving");
    }
}

public class DIP {
    public static void main(String[] args) {
        Engine engine = new ElectricalEngine();
        Car car = new Car(engine);
        car.drive();
    }
}
```

```
<terminated> DIP [Java Application] C:\Users\Jagadeesh Dowluri\Downloads\sprin
Engine started
Car started driving
```



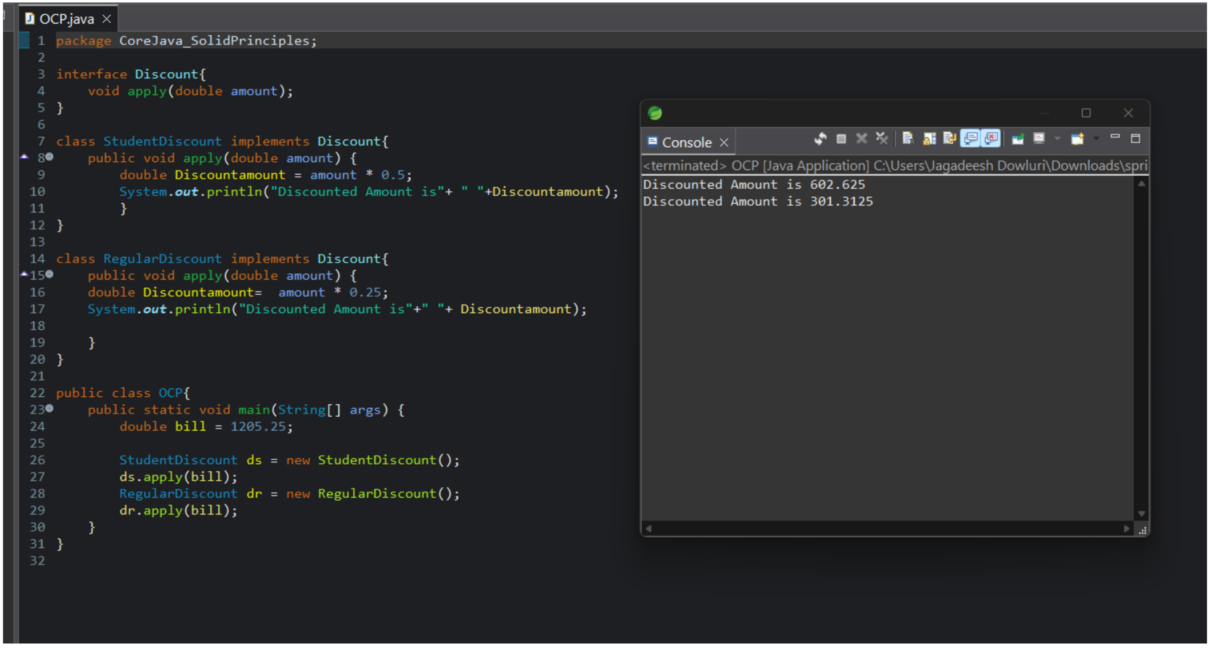
```java
package CoreJava;

interface Cooking {
    void cook();
}

interface Cleaning {
    void clean();
}


class Chef implements Cooking {
    public void cook() {
        System.out.println("Chef is cooking");
    }
}

class Cleaner implements Cleaning {
    public void clean() {
        System.out.println("Cleaner is cleaning");
    }
}

public class ISP {
    public static void main(String[] args) {
        Cooking chef = new Chef();
        chef.cook();

        Cleaning cleaner = new Cleaner();
        cleaner.clean();
    }
}
```

```
<terminated> ISP [Java Application] C:\Users\Jagadeesh Dowluri\Downloads\sprin
Chef is cooking
Cleaner is cleaning
```



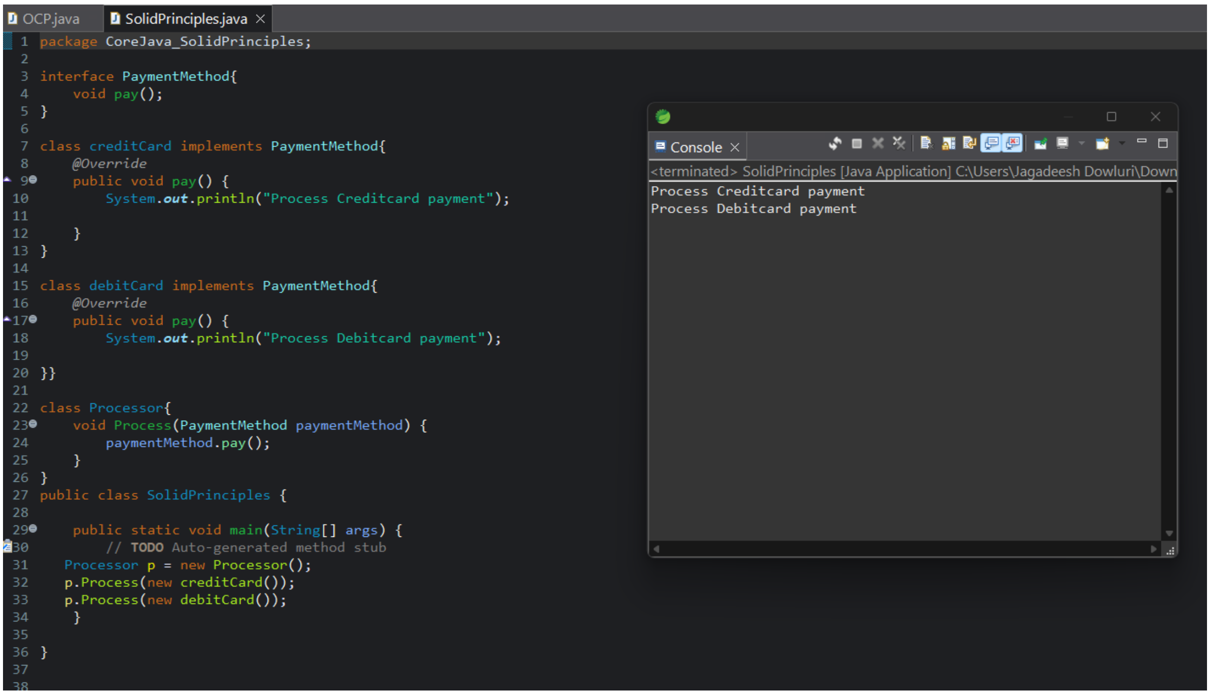
```java
package CoreJava_SolidPrinciples;

interface Bottle{
    void use();
}

class Waterbottle implements Bottle{
    public void use() {
        System.out.println("Used for Water");
    }
}

class JuiceBottle implements Bottle{
    public void use() {
        System.out.println("Used for Juice");
    }
}

public class LSP {

    public static void main(String[] args) {
        Bottle wb = new Waterbottle();
        wb.use();
        Bottle jb = new JuiceBottle();
        jb.use();


    }

}
```

```
<terminated> LSP [Java Application] C:\Users\Jagadeesh Dowluri\Downloads\sprir
Used for Water
Used for Juice
```



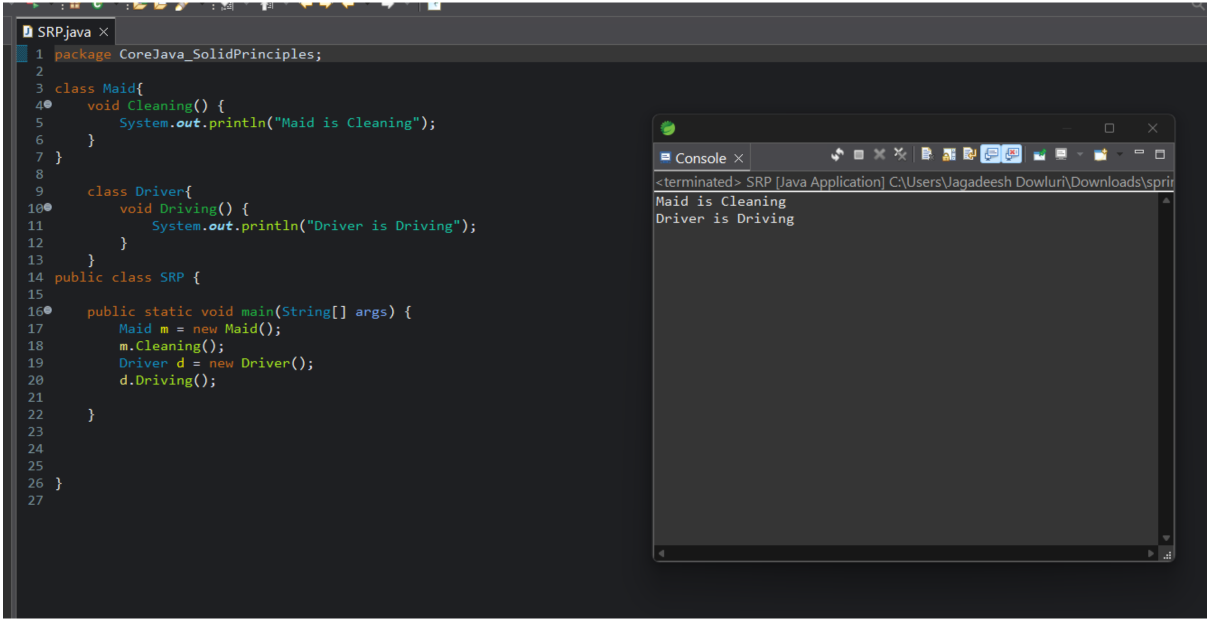
```java
package CoreJava_SolidPrinciples;

interface Discount{
    void apply(double amount);
}

class StudentDiscount implements Discount{
    public void apply(double amount) {
        double Discountamount = amount * 0.5;
        System.out.println("Discounted Amount is"+ " "+Discountamount);
        }
}

class RegularDiscount implements Discount{
    public void apply(double amount) {
    double Discountamount=  amount * 0.25;
    System.out.println("Discounted Amount is"+" "+ Discountamount);

    }
}

public class OCP{
    public static void main(String[] args) {
        double bill = 1205.25;

        StudentDiscount ds = new StudentDiscount();
        ds.apply(bill);
        RegularDiscount dr = new RegularDiscount();
        dr.apply(bill);
    }
}
```

```
<terminated> OCP [Java Application] C:\Users\Jagadeesh Dowluri\Downloads\spri
Discounted Amount is 602.625
Discounted Amount is 301.3125
```



```
package CoreJava_SolidPrinciples;

interface PaymentMethod{
    void pay();
}

class creditCard implements PaymentMethod{
    @Override
    public void pay() {
        System.out.println("Process Creditcard payment");

    }
}

class debitCard implements PaymentMethod{
    @Override
    public void pay() {
        System.out.println("Process Debitcard payment");

}}

class Processor{
    void Process(PaymentMethod paymentMethod) {
        paymentMethod.pay();
    }
}
public class SolidPrinciples {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
   Processor p = new Processor();
   p.Process(new creditCard());
   p.Process(new debitCard());
    }

}
```

```
<terminated> SolidPrinciples [Java Application] C:\Users\Jagadeesh Dowluri\Down
Process Creditcard payment
Process Debitcard payment
```



```
package CoreJava_SolidPrinciples;

class Maid{
    void Cleaning() {
        System.out.println("Maid is Cleaning");
    }
}

    class Driver{
        void Driving() {
            System.out.println("Driver is Driving");
        }
    }
public class SRP {

    public static void main(String[] args) {
        Maid m = new Maid();
        m.Cleaning();
        Driver d = new Driver();
        d.Driving();

    }



}
```

```
<terminated> SRP [Java Application] C:\Users\Jagadeesh Dowluri\Downloads\sprir
Maid is Cleaning
Driver is Driving
```