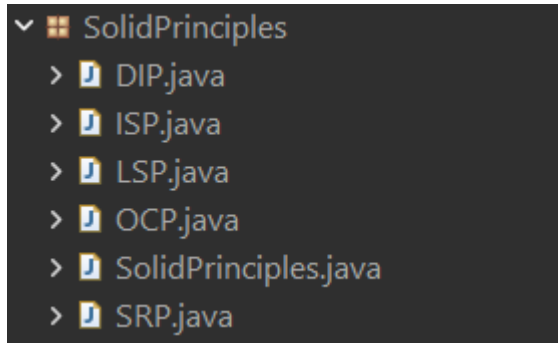


# Solid Principles



```
DIP.java ×
1 package CoreJava;
2
3 interface Engine {
4     void start();
5 }
6
7 class ElectricalEngine implements Engine {
8     @Override
9     public void start() {
10         System.out.println("Engine started");
11     }
12 }
13
14 class Car {
15     private Engine engine;
16
17     public Car(Engine engine) {
18         this.engine = engine;
19     }
20
21     void drive() {
22         engine.start();
23         System.out.println("Car started driving");
24     }
25 }
26
27 public class DIP {
28     public static void main(String[] args) {
29         Engine engine = new ElectricalEngine();
30         Car car = new Car(engine);
31         car.drive();
32     }
33 }
34
```

Console ×

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

```
ISP.java ×
1 package CoreJava;
2
3 interface Cooking {
4     void cook();
5 }
6
7 interface Cleaning {
8     void clean();
9 }
10
11 class Chef implements Cooking {
12     public void cook() {
13         System.out.println("Chef is cooking");
14     }
15 }
16
17 class Cleaner implements Cleaning {
18     public void clean() {
19         System.out.println("Cleaner is cleaning");
20     }
21 }
22
23
24 public class ISP {
25     public static void main(String[] args) {
26         Cooking chef = new Chef();
27         chef.cook();
28
29         Cleaning cleaner = new Cleaner();
30         cleaner.clean();
31     }
32 }
33
```

Console ×

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

```

1 package CoreJava_SolidPrinciples;
2
3 interface Bottle{
4     void use();
5 }
6
7 class Waterbottle implements Bottle{
8     public void use() {
9         System.out.println("Used for Water");
10    }
11 }
12
13 class JuiceBottle implements Bottle{
14     public void use() {
15         System.out.println("Used for Juice");
16    }
17 }
18
19 public class LSP {
20
21     public static void main(String[] args) {
22         Bottle wb = new Waterbottle();
23         wb.use();
24         Bottle jb = new JuiceBottle();
25         jb.use();
26     }
27 }
28
29 }
30
31

```

```

Console x
<terminated> LSP [Java Application] C:\Users\Vagadeesh Dowluri\Downloads\spring
Used for Water
Used for Juice

```

```

1 package CoreJava_SolidPrinciples;
2
3 interface Discount{
4     void apply(double amount);
5 }
6
7 class StudentDiscount implements Discount{
8     public void apply(double amount) {
9         double Discountamount = amount * 0.5;
10        System.out.println("Discounted Amount is"+ " " +Discountamount);
11    }
12 }
13
14 class RegularDiscount implements Discount{
15     public void apply(double amount) {
16         double Discountamount= amount * 0.25;
17        System.out.println("Discounted Amount is"+ " " + Discountamount);
18    }
19 }
20
21
22 public class OCP{
23     public static void main(String[] args) {
24         double bill = 1205.25;
25
26         StudentDiscount ds = new StudentDiscount();
27         ds.apply(bill);
28         RegularDiscount dr = new RegularDiscount();
29         dr.apply(bill);
30     }
31 }
32

```

```

Console x
<terminated> OCP [Java Application] C:\Users\Vagadeesh Dowluri\Downloads\spring
Discounted Amount is 602.625
Discounted Amount is 301.3125

```

1 OCP.java2 SolidPrinciples.java ×

```
1 package CoreJava_SolidPrinciples;
2
3 interface PaymentMethod{
4     void pay();
5 }
6
7 class creditCard implements PaymentMethod{
8     @Override
9     public void pay() {
10         System.out.println("Process Creditcard payment");
11     }
12 }
13
14 class debitCard implements PaymentMethod{
15     @Override
16     public void pay() {
17         System.out.println("Process Debitcard payment");
18     }
19 }
20
21
22 class Processor{
23     void Process(PaymentMethod paymentMethod) {
24         paymentMethod.pay();
25     }
26 }
27
28 public class SolidPrinciples {
29     public static void main(String[] args) {
30         // TODO Auto-generated method stub
31         Processor p = new Processor();
32         p.Process(new creditCard());
33         p.Process(new debitCard());
34     }
35 }
36
37
38
```

Console ×

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

1 SRP.java ×

```
1 package CoreJava_SolidPrinciples;
2
3 class Maid{
4     void Cleaning() {
5         System.out.println("Maid is Cleaning");
6     }
7 }
8
9 class Driver{
10     void Driving() {
11         System.out.println("Driver is Driving");
12     }
13 }
14
15 public class SRP {
16     public static void main(String[] args) {
17         Maid m = new Maid();
18         m.Cleaning();
19         Driver d = new Driver();
20         d.Driving();
21     }
22 }
23
24
25
26 }
27
```

Console ×

<terminated> SRP [Java Application] C:\Users\Vagadeesh Dowluri\Downloads\srp
Maid is Cleaning
Driver is Driving