

SOLID Principles

Single Responsibility Principle

```
Single_R_P.java X
1 package assignment;
2
3 //one class=one job
4 class Student{
5     String name;
6
7     Student(String name){
8         this.name=name;
9     }
10 }
11
12 class Print{
13     void print(Student s) {
14         System.out.println("Name is : "+s.name);
15     }
16 }
17
18 public class Single_R_P {
19
20     public static void main(String[] args) {
21
22         Student s=new Student("Archana");
23         Print p=new Print();
24         p.print(s);
25
26     }
27
28 }
```

<terminated> Single_R_P [Java Application] D:\sts\sts-4.31.0.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64\bin\java.exe -Xms128m -Xmx1024m -Djava.library.path=D:\sts\sts-4.31.0.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64\bin\native-images\win32\image\com.ibm.icu\dtdt\libicu4c.dll -Djava.class.path=D:\sts\sts-4.31.0.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64\bin\java.exe -Djava.ext.dirs=D:\sts\sts-4.31.0.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64\bin\ext\libext.jar,D:\sts\sts-4.31.0.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64\bin\ext\libext.jar -Djava.home=D:\sts\sts-4.31.0.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64\bin\jre\lib\rt.jar -Djava.io.tmpdir=C:\Users\pratik\AppData\Local\Temp\Eclipse IDE - Java(TM) SE Runtime Environment\12.0.2_10\temp -Djava.locale.providers=COMPAT,SUPPLIE

Open/Closed Principle

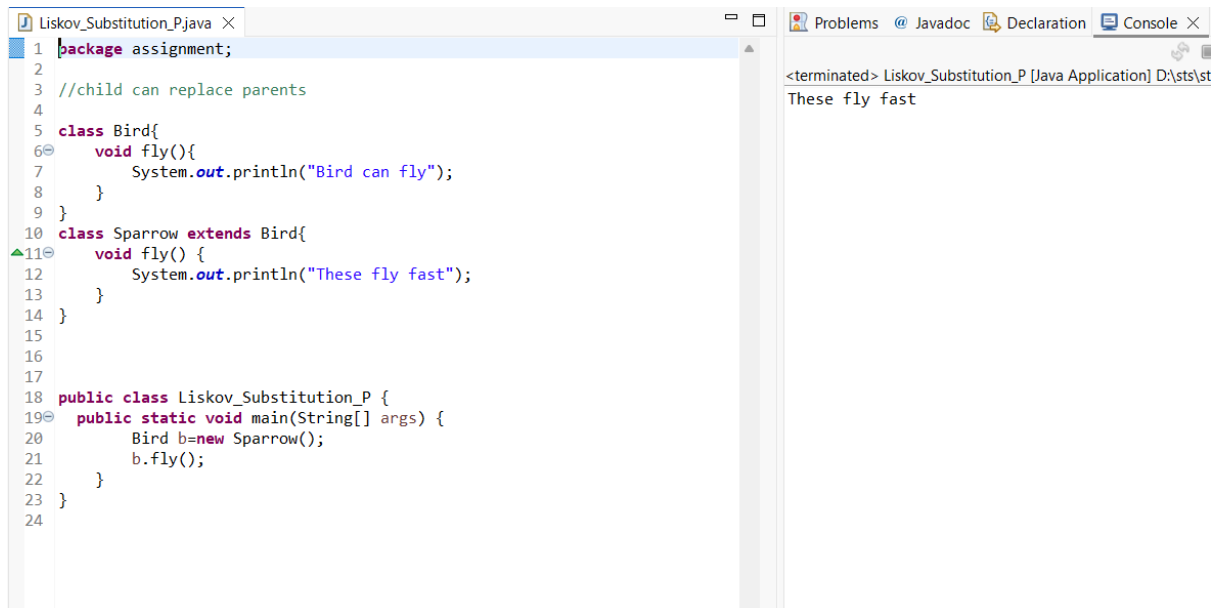
Open_Close_P.java ×

```
1 package assignment;
2
3 //Add new feature without changing old cod
4
5 interface PaymentMethod{
6     void pay();
7 }
8
9 class credit implements PaymentMethod{
10     public void pay() {
11         System.out.println("process CreditCard Payment");
12     }
13 }
14 //this is new feature
15 class debit implements PaymentMethod{
16     public void pay() {
17         System.out.println("process DebitCard Payment");
18     }
19 }
20 class Processor{
21     void Process(PaymentMethod paymentMethod) {
22         paymentMethod.pay();
23     }
24 }
25
26
27 public class Open_Close_P {
28
29     public static void main(String[] args) {
30         Processor p=new Processor();
31         p.Process(new credit());
32         p.Process(new debit());
33     }
34
35 }
```

Problems Javadoc Declaration Console ×

<terminated> Single_R_P [Java Application] D:\sts\sts-4.31.0.RE
Name is : Archana

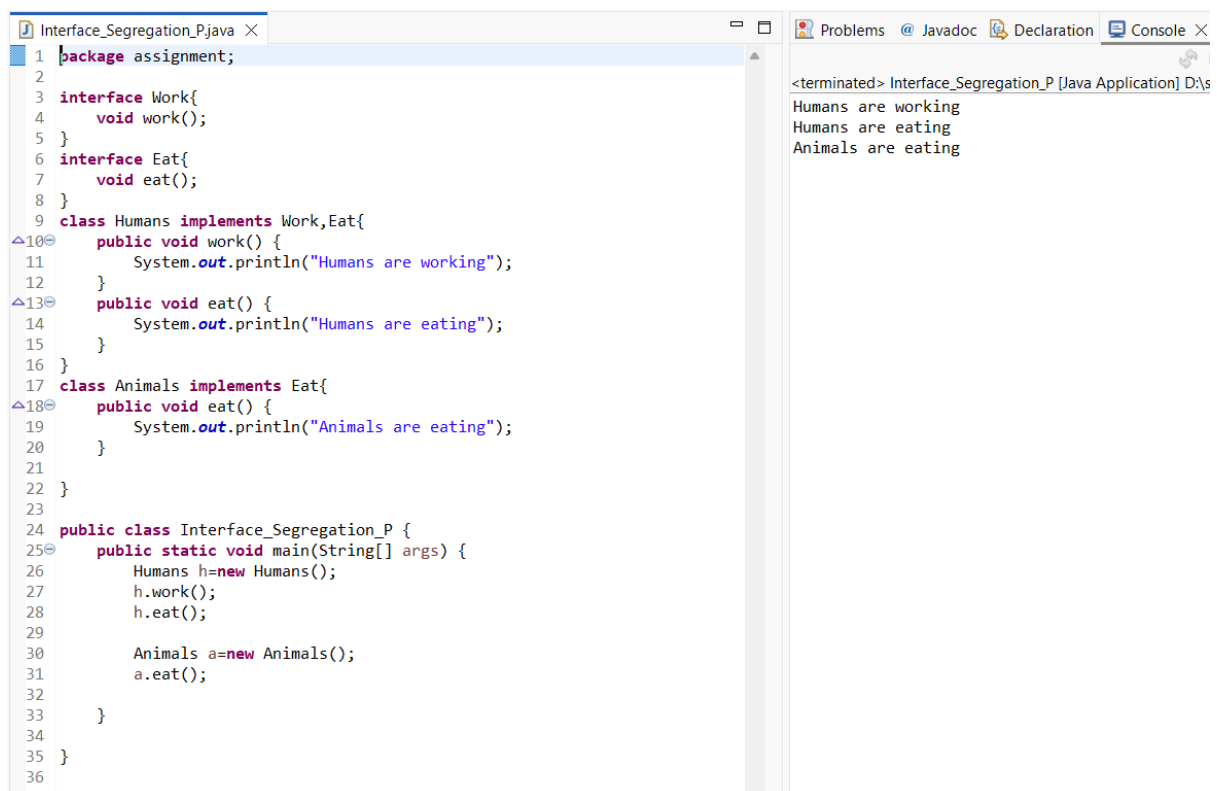
Liskov Substitution Principle



```
1 package assignment;
2
3 //child can replace parents
4
5 class Bird{
6     void fly(){
7         System.out.println("Bird can fly");
8     }
9 }
10 class Sparrow extends Bird{
11     void fly() {
12         System.out.println("These fly fast");
13     }
14 }
15
16
17
18 public class Liskov_Substitution_P {
19     public static void main(String[] args) {
20         Bird b=new Sparrow();
21         b.fly();
22     }
23 }
24
```

<terminated> Liskov_Substitution_P [Java Application] D:\sts\st
These fly fast

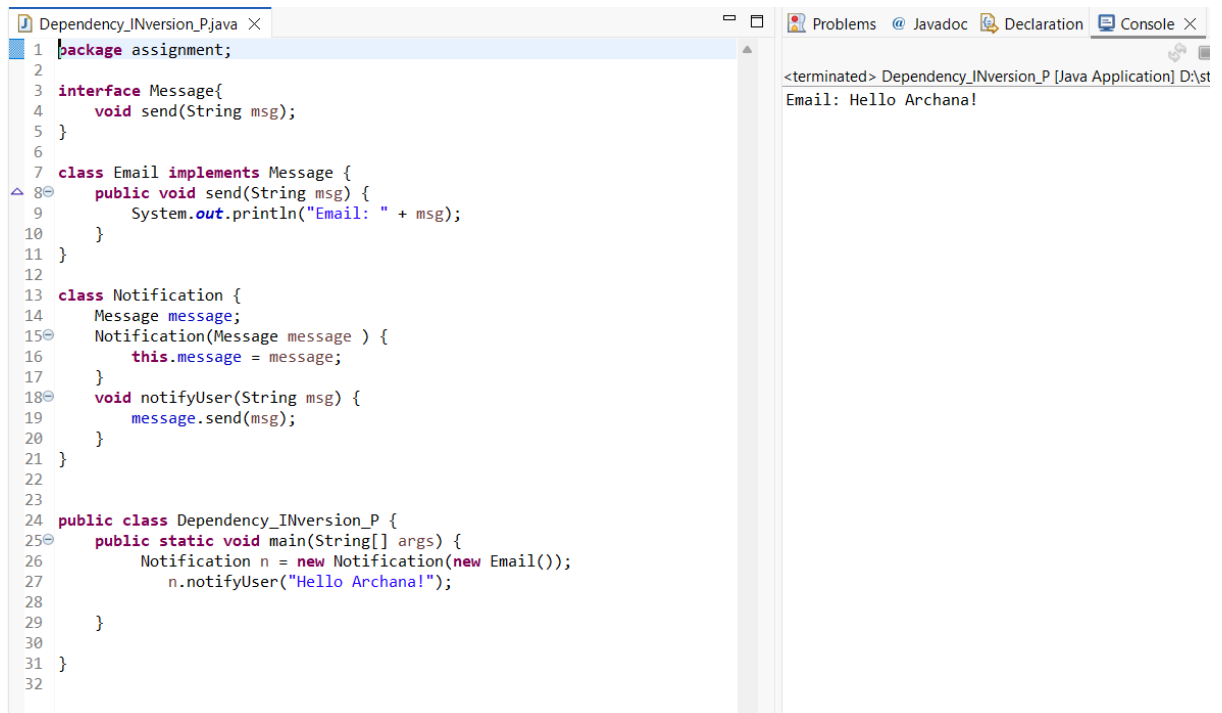
Interface Segregation Principle



```
1 package assignment;
2
3 interface Work{
4     void work();
5 }
6 interface Eat{
7     void eat();
8 }
9 class Humans implements Work,Eat{
10     public void work() {
11         System.out.println("Humans are working");
12     }
13     public void eat() {
14         System.out.println("Humans are eating");
15     }
16 }
17 class Animals implements Eat{
18     public void eat() {
19         System.out.println("Animals are eating");
20     }
21 }
22
23
24 public class Interface_Segregation_P {
25     public static void main(String[] args) {
26         Humans h=new Humans();
27         h.work();
28         h.eat();
29
30         Animals a=new Animals();
31         a.eat();
32     }
33 }
34
35 }
36
```

<terminated> Interface_Segregation_P [Java Application] D:\s
Humans are working
Humans are eating
Animals are eating

Dependency Inversion Principle



```
1 package assignment;
2
3 interface Message{
4     void send(String msg);
5 }
6
7 class Email implements Message {
8     public void send(String msg) {
9         System.out.println("Email: " + msg);
10    }
11 }
12
13 class Notification {
14     Message message;
15     Notification(Message message) {
16         this.message = message;
17     }
18     void notifyUser(String msg) {
19         message.send(msg);
20     }
21 }
22
23
24 public class Dependency_INversion_P {
25     public static void main(String[] args) {
26         Notification n = new Notification(new Email());
27         n.notifyUser("Hello Archana!");
28     }
29 }
30
31 }
32
```

<terminated> Dependency_INversion_P [Java Application] D:\st
Email: Hello Archana!