SOLID Documentation

Devapriya L

January 2024

1 Single Responsibility Principle

Interface:

```
package com.ilp.interfaces;

public interface NotificationService {
   void sendNotification(String message);
}
```

Different services :

```
package com.ilp.service;

import com.ilp.interfaces.NotificationService;

public class EmailNotificationService implements
   NotificationService{

  @Override
  public void sendNotification(String message) {
    // TODO Auto-generated method stub
     System.out.println("Email␣Notification:␣" +
        message);
  }

}
```

```java
package com.ilp.service;

import com.ilp.interfaces.NotificationService;

public class SMSNotificationService implements NotificationService {

  @Override
  public void sendNotification(String message) {
    // TODO Auto-generated method stub
    System.out.println("SMS␣Notification:␣" + message);
  }

}
```

```java
package com.ilp.service;

import com.ilp.interfaces.NotificationService;

public class WhatsAppNotificationService implements NotificationService{

  @Override
  public void sendNotification(String message) {
    // TODO Auto-generated method stub

  }

}
```

2 Open Close Principle

```java
package com.ilp.services;

import com.ilp.interfaces.SendNotification;

public class NotificationServices {
  public static SendNotification
     createNotification(String type) {
        if ("SMS".equals(type)) {
            return new SMSNotification();
        } else if ("Email".equals(type)) {
            return new EmailNotification();
        } else {
            throw new
               IllegalArgumentException("Invalid␣
               notification␣type");
        }
    }
}
```

```java
    package com.ilp.utility;

import com.ilp.interfaces.SendNotification;
//import com.ilp.services.EmailNotification;
import com.ilp.services.NotificationServices;
//import com.ilp.services.SMSNotification;

public class NotificationSystem {

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

    SendNotification smsNotification =
       NotificationServices.createNotification("SMS");
        smsNotification.sendNotifications("Your␣
           booking␣is␣successful.");

        SendNotification emailNotification =
           NotificationServices.createNotification("Email");
        emailNotification.sendNotifications("Your␣
           booking␣is␣successful.");
  }

}
```

3 Liskov Substitution Principle

Base class:

```
    package com.ilp.services;

import com.ilp.interfaces.SendNotification;

public class BaseNotification implements SendNotification{

  @Override
  public void sendNotifications(String message) {
    // TODO Auto-generated method stub
     System.out.println("Base␣Notification:␣" + message);

  }

}
```

Child classes:

```
    package com.ilp.services;

public class EmailNotification extends BaseNotification{

  @Override
  public void sendNotifications(String message) {
    // TODO Auto-generated method stub
    System.out.println("Email␣Notification:"+ message);
  }

}
```

```
    package com.ilp.services;

public class SMSNotification extends BaseNotification{

  @Override
  public void sendNotifications(String message) {
    System.out.println("SMS␣Notification:"+ message);
  }

}
```

4 Interface Segregation Principle

Interfaces:

```
    package com.ilp.interfaces;

public interface EmailNotification {
   void sendEmail(String message);
}
```

```
 package com.ilp.interfaces;

public interface PushNotification {
   void sendPushNotification(String message);
}
```

Classes:

```
    package com.ilp.services;

import com.ilp.interfaces.EmailNotification;

public class EmailNotificationService implements EmailNotification{

  @Override
  public void sendEmail(String message) {
    // TODO Auto-generated method stub
    System.out.println("Email␣Notification:"+ message);
  }

}
```

```
    package com.ilp.services;

import com.ilp.interfaces.PushNotification;

public class PushNotificationService implements PushNotification{

  @Override
  public void sendPushNotification(String message) {
    // TODO Auto-generated method stub
    System.out.println("Push␣Notification:"+ message);
  }
}
```

5 Dependency Inversion Principle

```
    package com.ilp.utility;

import com.ilp.interfaces.SendNotification;
import com.ilp.services.EmailNotification;
import com.ilp.services.SMSNotification;

public class NotificationSystem {

    private final SendNotification sendNotification;

    public NotificationSystem(SendNotification sendNotification) {
        this.sendNotification = sendNotification;
    }

    public void sendNotifications(String message) {
        sendNotification.sendNotifications(message);
    }

    public static void main(String[] args) {
        SendNotification smsNotification = new SMSNotification();
        SendNotification emailNotification = new EmailNotification();

        NotificationSystem smsSystem = new NotificationSystem(smsNotification);
        smsSystem.sendNotifications("Your␣booking␣is␣successful.");

        NotificationSystem emailSystem = new NotificationSystem(emailNotification);
        emailSystem.sendNotifications("Your␣booking␣is␣successful.");
    }
}
```