

Experiment – 8

Aim –Adapter design pattern (Structural pattern)

Name – Shivraj Nakum , 21BCP125

Concept -converts the interface of a class into another interface that a client wants . In other words, to provide the interface according to client requirement while using the services of a class with a different interface.

Files –

Interface Food.java (old)

Interface NewRecipe.java (new)

Italian_kitchen.java

Mexiacn_kitchen.java

Recipe.java

RecipeAdapter.java

Client.java

////////////////////////////////////

Food.java

```
public interface Food {  
    public void cook(String cuisine , String dishName);  
}
```

NewRecipe.java

```
public interface NewRecipe {
    public void cookMexican(String dishName);
    public void cookItalian(String dishName);
}
```

Italian_kitchen.java

```
public class Italian_kitchen implements NewRecipe{
    @Override
    public void cookMexican(String dishName) {
        //do nothing
    }

    @Override
    public void cookItalian(String dishName) {
        System.out.println("Cooking Italian food " + ", " + "DishName: "
+ dishName);
    }
}
```

Mexiacn_kitchen.java

```
public class Mexican_kitchen implements NewRecipe{

    @Override
    public void cookMexican(String dishName) {
        System.out.println("Cooking Mexican food " + ", " + "DishName: "
+ dishName);
    }

    @Override
    public void cookItalian(String dishName) {
        //do nothing
    }
}
```

Recipe.java

```
public class Recipe implements Food {

    RecipeAdapter recipeAdapter;

    @Override
    public void cook(String cuisine, String dishName) {

        //inbuilt food of Dal-Roti
        if (cuisine.equalsIgnoreCase("normal")){
            System.out.println("Cooking normal food " + ", " + "DishName: "
+ dishName);
        } else if (cuisine.equalsIgnoreCase("mexican") ||
cuisine.equalsIgnoreCase("italian")) {
            recipeAdapter = new RecipeAdapter(cuisine);
            recipeAdapter.cook(cuisine,dishName);
        }else {
            System.out.println("invalid cuisine " + dishName + " not
available");
        }
    }
}
```

```
    }  
}  
}
```

RecipeAdapter.java

```
public class RecipeAdapter implements Food{  
  
    NewRecipe newRecipe;  
  
    public RecipeAdapter(String cuisine){  
        if (cuisine.equalsIgnoreCase("mexican")){  
            newRecipe = new Mexican_kitchen();  
        } else if (cuisine.equalsIgnoreCase("italian")) {  
            newRecipe = new Italian_kitchen();  
        }  
    }  
  
    @Override  
    public void cook(String cuisine, String dishName) {  
        if (cuisine.equalsIgnoreCase("mexican")){  
            newRecipe.cookMexican(dishName);  
        } else if (cuisine.equalsIgnoreCase("italian")) {  
            newRecipe.cookItalian(dishName);  
        }  
    }  
}
```

Client.java

```
public class Client {  
    public static void main(String[] args) {  
        Recipe obj = new Recipe();  
  
        obj.cook("normal", "dal-roti");  
        obj.cook("mexican", "tacos");  
        obj.cook("italian", "pizza");  
        obj.cook("chinese", "noodles");  
    }  
}
```

Output :

```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community
Cooking normal food , DishName: dal-roti
Cooking Mexican food , DishName: tacos
Cooking Italian food , DishName: pizza
invalid cuisine nooodles not available
```