* Proxy design pattern provide a surrogate or placeholder for another object to control access to it.
* It is a structural design pattern.
* It is used whenever we need a placeholder or representational object that can work in place of the real object.
* The proxy acts as an intermediatory layer between the client and the real object and hence can control the access to the real object, add additional functionality, and even restrict client access.
* It is also known as Surrogate Pattern.

It has the following use-cases :

* Access restriction
* Caching ( Proxy will check the orginal request from the client whether it is already available in the cache )
* Preprocessing and Postprocessing the actual client request ( Marshalling and unmarshalling )

public interface EmployeeService {  
  
 void create(String client ) throws Exception;  
 void delete(String client ) throws Exception;  
 String get(String client ) throws Exception;  
  
}

public class EmployeeProxyService implements EmployeeService{  
  
 EmployeeService employeeService ;  
  
 EmployeeProxyService(){  
 employeeService = new EmployeeConcreteService() ;  
 }  
  
  
 @Override  
 public void create(String client) throws Exception {  
 if(client.equals("ADMIN")){  
 employeeService.create( client );  
 }else {  
 throw new Exception("Access Denied") ;  
 }  
 }  
  
 @Override  
 public void delete(String client) throws Exception {  
 if(client.equals("ADMIN")){  
 employeeService.delete( client );  
 }else {  
 throw new Exception("Access Denied") ;  
 }  
 }  
  
 @Override  
 public String get(String client) throws Exception {  
 if(client.equals("ADMIN") || client.equals("USER") ){  
 return employeeService.get( client );  
 }else {  
 throw new Exception("Access Denied") ;  
 }  
 }  
}

public class EmployeeConcreteService implements EmployeeService{  
 @Override  
 public void create(String client ) {  
 System.*out*.println("Created new row ");  
 }  
  
 @Override  
 public void delete(String client ) {  
 System.*out*.println("Deleted a row");  
 }  
  
 @Override  
 public String get(String client ) {  
 return "Employee details";  
 }  
}

public class ProxyDesignPattern {  
  
 public static void main(String[] args) {  
 EmployeeService employeeService = new EmployeeProxyService();  
  
 try {  
 employeeService.create("ADMIN");  
 employeeService.delete("ADMIN");  
 System.*out*.println(employeeService.get("USER"));  
 employeeService.create("USER");  
 } catch (Exception e) {  
 System.*out*.println(e.getMessage());  
 }  
 }  
  
}

Output :

Created new row

Deleted a row

Employee details

Access Denied

Process finished with exit code 0