

The proxyDesign Pattern Potential “Solution”

~gibson/Teaching/CSC4522/CSC4522-DesignPatterns-Proxy-Solution.pdf

Proxy Problem

Create a service, as a method of a Java class, that will take an integer and return if it is odd or even

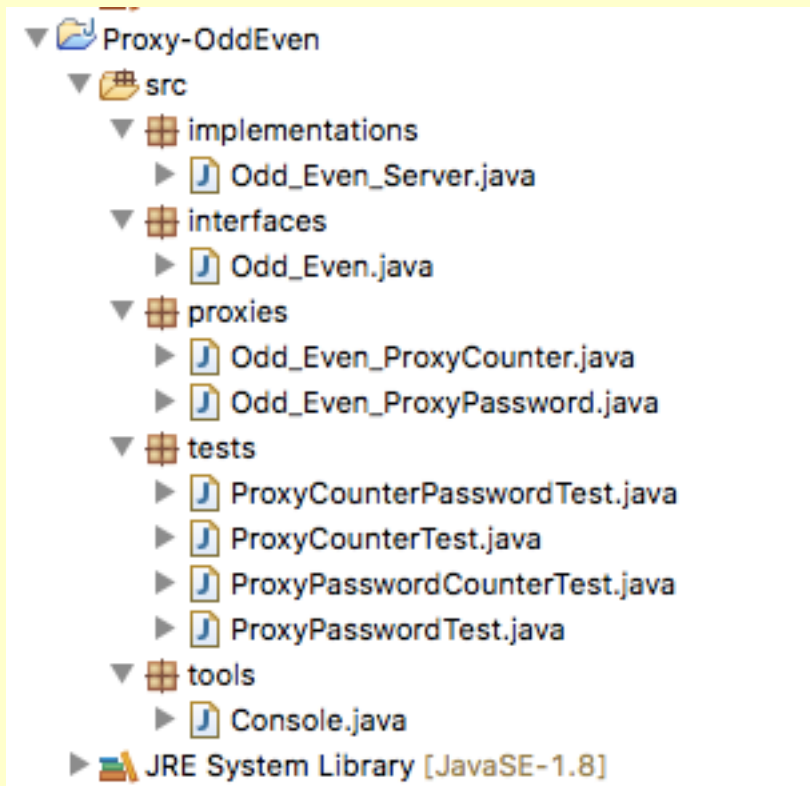
Write a proxy for the service that will ask for a password before the service is executed

Write a proxy that will count the number of times the service is executed

Implement 2 double proxys:

- 1) Asks a password, then counts
- 2) Counts, then asks a password

Proxy Problem



Proxy Problem

```
public interface Odd_Even {  
  
    /**  
     * return if given integer is even  
     */  
    boolean isEven (int x);  
  
}
```

Proxy Problem

```
public class Odd_Even_Server implements Odd_Even{  
  
    public boolean isEven(int x) {  
  
        return x%2 == 0;  
    }  
  
}
```

Proxy Problem

```
public class Odd_Even_ProxyCounter implements Odd_Even{

    private int count;

    Odd_Even oddOrEven;

    public Odd_Even_ProxyCounter(Odd_Even oddOrEven){

        this.oddOrEven = oddOrEven;
        count =0;
    }

    public boolean isEven(int x) {

        count++;
        return oddOrEven.isEven(x);
    }

    public int getCount(){
        return count;
    }
}
```

Proxy Problem

```
public class Odd_Even_ProxyPassword implements Odd_Even{

    private final String PASSWORD = "password";

    Odd_Even oddOrEven;

    public Odd_Even_ProxyPassword(Odd_Even oddOrEven){

        this.oddOrEven = oddOrEven;

    }

    public boolean isEven(int x){

        String input;

        do{
            System.out.println("Please enter the password: ");
            input= Console.readLine();
        }
        while (! input.equals(PASSWORD));
        return oddOrEven.isEven(x);

    }

}
```

Proxy Problem

```
public class ProxyCounterTest {  
  
    public static void main(String args[]){  
  
        Odd_Even_Server server = new Odd_Even_Server();  
  
        Odd_Even_ProxyCounter proxyCounter = new Odd_Even_ProxyCounter(server);  
  
        System.out.println("Proxy count = "+proxyCounter.getCount() );  
  
        for (int i=0; i<5; i++){  
            System.out.println("Proxy checking if "+i+" is even returns "+ proxyCounter.isEven(i));  
            System.out.println("Proxy count = "+proxyCounter.getCount() );  
        }  
    }  
}
```


Proxy Problem

```
public class ProxyPasswordTest {  
  
    public static void main(String args[]){  
  
        Odd_Even_Server server = new Odd_Even_Server();  
  
        Odd_Even_ProxyPassword proxyPassword = new Odd_Even_ProxyPassword(server);  
  
        for (int i=0; i<3; i++){  
            System.out.println("Proxy checking if "+i+" is even returns "+ proxyPassword.isEven(i));  
        }  
  
    }  
  
}
```

Proxy Problem

```
public class ProxyCounterPasswordTest {

    public static void main(String args[]){

        Odd_Even_Server server = new Odd_Even_Server();
        Odd_Even_ProxyPassword proxyPassword = new Odd_Even_ProxyPassword(server);
        Odd_Even_ProxyCounter proxyCounterPassword =
            new Odd_Even_ProxyCounter(proxyPassword);

        System.out.println("Proxy count = "+proxyCounterPassword.getCount() );

        for (int i=0; i<3; i++){
            System.out.println("Proxy checking if "+i+" is even returns "+
                proxyCounterPassword.isEven(i));
            System.out.println("Proxy count = "+proxyCounterPassword.getCount() );
        }

    }

}
```

Proxy Problem

```
public class ProxyPasswordCounterTest {

    public static void main(String args[]){

        Odd_Even_Server server = new Odd_Even_Server();
        Odd_Even_ProxyCounter proxyCounter = new Odd_Even_ProxyCounter(server);
        Odd_Even_ProxyPassword proxyPasswordCounter =
            new Odd_Even_ProxyPassword(proxyCounter);

        System.out.println("Proxy count = "+proxyCounter.getCount() );

        for (int i=0; i<3; i++){
            System.out.println("Proxy checking if "+i+" is even returns "+
                proxyPasswordCounter.isEven(i));
            System.out.println("Proxy count = "+proxyCounter.getCount() );
        }

    }

}
```

QUESTION: What is the difficult problem that we have not considered in our solution?

HOW COULD WE FIX IT?