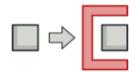






命/设计模式/代理模式/Java



Java 代理模式讲解和代码示例

代理是一种结构型设计模式,让你能提供真实服务对象的替代品给客户端使用。 代理接收客户端的请求并进行一些处理(访问控制和缓存等),然后再将请求传 递给服务对象。

代理对象拥有和服务对象相同的接口,这使得当其被传递给客户端时可与真实对象互换。

Ⅲ 进一步了解代理模式 →

在 Java 中使用模式

复杂度: ★★☆

流行度: ★☆☆

使用示例:尽管代理模式在绝大多数 Java 程序中并不常见,但它在一些特殊情况下仍然非常方便。 当你希望在无需修改客户代码的前提下于已有类的对象上增加额外行为时,该模式是无可替代的。

Java 标准程序库中的一些代理模式的示例:

- java.lang.reflect.Proxy
- java.rmi.*
- javax.ejb.EJB (查看评论)
- javax.inject.Inject (查看评论)
- javax.persistence.PersistenceContext

识别方法: 代理模式会将所有实际工作委派给一些其他对象。除非代理是某个服务的子类,否则每个代理方法最后都应该引用一个服务对象。

导航

- 囯 简介
- 囯 缓存代理
- some_cool_media_library
- ThirdPartyYouTubeClass
- ₩ Video
- proxy

- YouTubeDownloader
- OutputDemo

缓存代理

在本例中,代理模式有助于实现延迟初始化,并对低效的第三方 YouTube 集成程序库进行缓存。

当你需要在无法修改代码的类上新增一些额外行为时,代理模式的价值无可估量。

⇒ some_cool_media_library

🖟 some_cool_media_library/ThirdPartyYouTubeLib.java: 远程服务接口

```
package refactoring_guru.proxy.example.some_cool_media_library;
import java.util.HashMap;

public interface ThirdPartyYouTubeLib {
    HashMap<String, Video> popularVideos();

    Video getVideo(String videoId);
}
```

🖟 some_cool_media_library/ThirdPartyYouTubeClass.java: 远程服务实现

```
package refactoring_guru.proxy.example.some_cool_media_library;
import java.util.HashMap;
public class ThirdPartyYouTubeClass implements ThirdPartyYouTubeLib {
    a0verride
    public HashMap<String, Video> popularVideos() {
        connectToServer("http://www.youtube.com");
        return getRandomVideos();
    }
    a0verride
    public Video getVideo(String videoId) {
        connectToServer("http://www.youtube.com/" + videoId);
        return getSomeVideo(videoId);
    }
    // Fake methods to simulate network activity. They as slow as a real life.
    private int random(int min, int max) {
        return min + (int) (Math.random() * ((max - min) + 1));
    }
    private void experienceNetworkLatency() {
        int randomLatency = random(5, 10);
        for (int i = 0; i < randomLatency; i++) {</pre>
            try {
                Thread.sleep(100);
            } catch (InterruptedException ex) {
                ex.printStackTrace();
        }
    }
    private void connectToServer(String server) {
        System.out.print("Connecting to " + server + "... ");
        experienceNetworkLatency();
        System.out.print("Connected!" + "\n");
    }
    private HashMap<String, Video> getRandomVideos() {
        System.out.print("Downloading populars... ");
        experienceNetworkLatency();
        HashMap<String, Video> hmap = new HashMap<String, Video>();
        hmap.put("catzzzzzzzz", new Video("sadgahasgdas", "Catzzzz.avi"));
        hmap.put("mkafksangasj", new Video("mkafksangasj", "Dog play with ball.mp4"));
        hmap.put("dancesvideoo", new Video("asdfas3ffasd", "Dancing video.mpq"));
        hmap.put("dlsdk5jfslaf", new Video("dlsdk5jfslaf", "Barcelona vs RealM.mov"));
```

```
hmap.put("3sdfgsd1j333", new Video("3sdfgsd1j333", "Programing lesson#1.avi"));

System.out.print("Done!" + "\n");
   return hmap;
}

private Video getSomeVideo(String videoId) {
   System.out.print("Downloading video... ");

   experienceNetworkLatency();
   Video video = new Video(videoId, "Some video title");

   System.out.print("Done!" + "\n");
   return video;
}
```

🖟 some_cool_media_library/Video.java: 视频文件

```
package refactoring_guru.proxy.example.some_cool_media_library;

public class Video {
    public String id;
    public String title;
    public String data;

    Video(String id, String title) {
        this.id = id;
        this.title = title;
        this.data = "Random video.";
    }
}
```

⊳ proxy

🖟 proxy/YouTubeCacheProxy.java: 缓存代理

```
package refactoring_guru.proxy.example.proxy;

import refactoring_guru.proxy.example.some_cool_media_library.ThirdPartyYouTubeClass;
import refactoring_guru.proxy.example.some_cool_media_library.ThirdPartyYouTubeLib;
import refactoring_guru.proxy.example.some_cool_media_library.Video;
import java.util.HashMap;
```

```
public class YouTubeCacheProxy implements ThirdPartyYouTubeLib {
    private ThirdPartyYouTubeLib youtubeService;
    private HashMap<String, Video> cachePopular = new HashMap<String, Video>();
    private HashMap<String, Video> cacheAll = new HashMap<String, Video>();
    public YouTubeCacheProxy() {
        this.youtubeService = new ThirdPartyYouTubeClass();
    @Override
    public HashMap<String, Video> popularVideos() {
        if (cachePopular.isEmpty()) {
            cachePopular = youtubeService.popularVideos();
            System.out.println("Retrieved list from cache.");
        return cachePopular;
    }
    a0verride
    public Video getVideo(String videoId) {
       Video video = cacheAll.get(videoId);
        if (video == null) {
            video = youtubeService.getVideo(videoId);
            cacheAll.put(videoId, video);
        } else {
            System.out.println("Retrieved video '" + videoId + "' from cache.");
        return video;
    }
    public void reset() {
        cachePopular.clear();
        cacheAll.clear();
    }
```


🖟 downloader/YouTubeDownloader.java: 媒体下载应用

```
package refactoring_guru.proxy.example.downloader;

import refactoring_guru.proxy.example.some_cool_media_library.ThirdPartyYouTubeLib;
import refactoring_guru.proxy.example.some_cool_media_library.Video;

import java.util.HashMap;

public class YouTubeDownloader {
    private ThirdPartyYouTubeLib api;
```

```
public YouTubeDownloader(ThirdPartyYouTubeLib api) {
       this.api = api;
   public void renderVideoPage(String videoId) {
      Video video = api.getVideo(videoId);
       System.out.println("\n----");
       System.out.println("Video page (imagine fancy HTML)");
       System.out.println("ID: " + video.id);
       System.out.println("Title: " + video.title);
       System.out.println("Video: " + video.data);
       System.out.println("-----\n");
   }
   public void renderPopularVideos() {
       HashMap<String, Video> list = api.popularVideos();
       System.out.println("\n----");
       System.out.println("Most popular videos on YouTube (imagine fancy HTML)");
       for (Video video : list.values()) {
          System.out.println("ID: " + video.id + " / Title: " + video.title);
      System.out.println("-----\n");
   }
}
```

🖟 Demo.java: 初始化代码

```
package refactoring_guru.proxy.example;
import refactoring_guru.proxy.example.downloader.YouTubeDownloader;
import refactoring_guru.proxy.example.proxy.YouTubeCacheProxy;
import refactoring_guru.proxy.example.some_cool_media_library.ThirdPartyYouTubeClass;
public class Demo {
    public static void main(String[] args) {
        YouTubeDownloader naiveDownloader = new YouTubeDownloader(new ThirdPartyYouTubeClass()
        YouTubeDownloader smartDownloader = new YouTubeDownloader(new YouTubeCacheProxy());
        long naive = test(naiveDownloader);
        long smart = test(smartDownloader);
        System.out.print("Time saved by caching proxy: " + (naive - smart) + "ms");
    }
    private static long test(YouTubeDownloader downloader) {
        long startTime = System.currentTimeMillis();
        // User behavior in our app:
        downloader.renderPopularVideos();
```

```
downloader.renderVideoPage("catzzzzzzzzz");
  downloader.renderPopularVideos();
  downloader.renderVideoPage("dancesvideoo");
  // Users might visit the same page quite often.
  downloader.renderVideoPage("catzzzzzzzzzz");
  downloader.renderVideoPage("someothervid");

long estimatedTime = System.currentTimeMillis() - startTime;
  System.out.print("Time elapsed: " + estimatedTime + "ms\n");
  return estimatedTime;
}
```

🖹 OutputDemo.txt: 执行结果

```
Connecting to http://www.youtube.com... Connected!
Downloading populars... Done!
_____
Most popular videos on YouTube (imagine fancy HTML)
ID: sadgahasgdas / Title: Catzzzz.avi
ID: asdfas3ffasd / Title: Dancing video.mpq
ID: 3sdfgsd1j333 / Title: Programing lesson#1.avi
ID: mkafksangasj / Title: Dog play with ball.mp4
ID: dlsdk5jfslaf / Title: Barcelona vs RealM.mov
_____
Connecting to http://www.youtube.com/catzzzzzzzzz... Connected!
Downloading video... Done!
Video page (imagine fancy HTML)
ID: catzzzzzzzzz
Title: Some video title
Video: Random video.
Connecting to http://www.youtube.com... Connected!
Downloading populars... Done!
_____
Most popular videos on YouTube (imagine fancy HTML)
ID: sadgahasgdas / Title: Catzzzz.avi
ID: asdfas3ffasd / Title: Dancing video.mpq
ID: 3sdfgsd1j333 / Title: Programing lesson#1.avi
ID: mkafksangasj / Title: Dog play with ball.mp4
ID: dlsdk5jfslaf / Title: Barcelona vs RealM.mov
Connecting to http://www.youtube.com/dancesvideoo... Connected!
```

```
Downloading video... Done!
_____
Video page (imagine fancy HTML)
ID: dancesvideoo
Title: Some video title
Video: Random video.
Connecting to http://www.youtube.com/catzzzzzzzzz... Connected!
Downloading video... Done!
_____
Video page (imagine fancy HTML)
ID: catzzzzzzzz
Title: Some video title
Video: Random video.
_____
Connecting to http://www.youtube.com/someothervid... Connected!
Downloading video... Done!
Video page (imagine fancy HTML)
ID: someothervid
Title: Some video title
Video: Random video.
_____
Time elapsed: 9354ms
Connecting to http://www.youtube.com... Connected!
Downloading populars... Done!
Most popular videos on YouTube (imagine fancy HTML)
ID: sadgahasgdas / Title: Catzzzz.avi
ID: asdfas3ffasd / Title: Dancing video.mpq
ID: 3sdfgsd1j333 / Title: Programing lesson#1.avi
ID: mkafksangasj / Title: Dog play with ball.mp4
ID: dlsdk5jfslaf / Title: Barcelona vs RealM.mov
_____
Connecting to http://www.youtube.com/catzzzzzzzzz... Connected!
Downloading video... Done!
_____
Video page (imagine fancy HTML)
ID: catzzzzzzzzz
Title: Some video title
Video: Random video.
Retrieved list from cache.
-----
```

```
Most popular videos on YouTube (imagine fancy HTML)
ID: sadgahasgdas / Title: Catzzzz.avi
ID: asdfas3ffasd / Title: Dancing video.mpq
ID: 3sdfgsd1j333 / Title: Programing lesson#1.avi
ID: mkafksangasj / Title: Dog play with ball.mp4
ID: dlsdk5jfslaf / Title: Barcelona vs RealM.mov
Connecting to http://www.youtube.com/dancesvideoo... Connected!
Downloading video... Done!
_____
Video page (imagine fancy HTML)
ID: dancesvideoo
Title: Some video title
Video: Random video.
_____
Retrieved video 'catzzzzzzzz' from cache.
Video page (imagine fancy HTML)
ID: catzzzzzzzzz
Title: Some video title
Video: Random video.
_____
Connecting to http://www.youtube.com/someothervid... Connected!
Downloading video... Done!
Video page (imagine fancy HTML)
ID: someothervid
Title: Some video title
Video: Random video.
Time elapsed: 5875ms
Time saved by caching proxy: 3479ms
```

继续阅读

Java 责任链模式讲解和代码示例 →

返回

← Java 享元模式讲解和代码示例

代理在其他编程语言中的实现

主页
重构
设计模式
会员专属内容
论坛
联系我们

© 2014-2020 Refactoring.Guru.版权所有
圆 Khmelnitske shosse 19 / 27, Kamianets-Podilskyi, 乌克兰, 32305
☑ Email: support@refactoring.guru
図图片作者: Dmitry Zhart

条款与政策
隐私政策
内容使用政策