Proxy

```
public interface ThirdPartyYouTubeLib {
       public Object listVideos();
       public Object getVideoInfo(int id);
       public void downloadVideo(int id);
}
public class ThirdPartyYouTubeClass implements ThirdPartyYouTubeLib {
       @Override
       public Object listVideos() {
              return "[LIST]: CachedYouTube videos";
       }
       @Override
       public Object getVideoInfo(int id) {
              return "[INFO]: " + id;
       }
       @Override
       public void downloadVideo(int id) {
              System.out.println("[DOWNLOAD]: " + id);
       }
}
public class CachedYouTubeProxyClass implements ThirdPartyYouTubeLib {
       ThirdPartyYouTubeClass service;
       private Object listCache, videoCache;
       private boolean needReset = true;
       public CachedYouTubeProxyClass(ThirdPartyYouTubeClass service) {
              this.service = service;
       }
       @Override
       public Object listVideos() {
              if(listCache == null || needReset) {
                     listCache = service.listVideos();
              }
              return listCache;
       }
       @Override
       public Object getVideoInfo(int id) {
              if(videoCache == null || needReset) {
                     videoCache = service.getVideoInfo(id);
              }
              return videoCache;
       }
       @Override
       public void downloadVideo(int id) {
              if(!downloadExists(id) || needReset)
                     service.downloadVideo(id);
       }
       boolean downloadExists(int id) {
              // TODO Auto-generated method stub
              return false;
       }
}
public class YouTubeManager {
       protected ThirdPartyYouTubeLib service;
       public YouTubeManager(ThirdPartyYouTubeLib service) {
              this.service = service;
       }
       public void getThing(int id) {
              service.downloadVideo(id);
       }
}
```

```
public class Main {
        public static void main(String[] args) {
                ThirdPartyYouTubeClass youTubeService = new ThirdPartyYouTubeClass();
                CachedYouTubeProxyClass youTubeProxy = new
CachedYouTubeProxyClass(youTubeService);
                YouTubeManager manager = new YouTubeManager(youTubeProxy);
                manager.getThing(4);
        }
}
```