

代理模式 - 电话虫

📅 发表于 2022-09-07 | 🕒 更新于 2023-04-06 | 📁 设计模式

| 📄 字数总计: 1.1k | ⌚ 阅读时长: 3 分钟 | 👁 阅读量: 415 | 💬 评论数: 0



配套视频课程已更新完毕，大家可通过以下两种方式观看视频讲解：



关注公众号： [📱 爱编程的大丙](#) ，或者进入 [📱 大丙课堂](#) 学习。

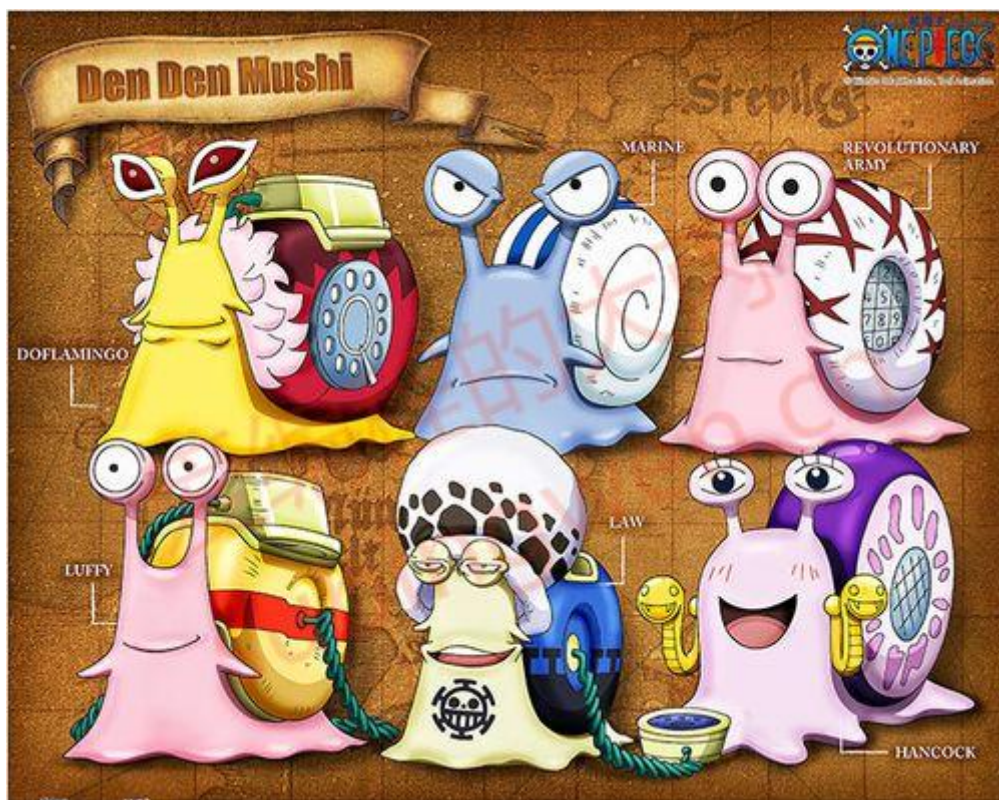


苏丙楦

合抱之木，生于毫末；九层之台，起于垒土；千里之行，始于足下。

1. 电话虫

在海贼中，有一种神奇的通信工具叫做电话虫（Den Den Mushi），外形如蜗牛，身上带有斑点或条纹或通体纯色，壳顶上有对讲机或按键，不接通时会睡觉，接通时会惊醒，并发出“波噜波噜”的声音，在通话时电话虫的嘴巴会如同讲话人的嘴巴一样动，也有人的感情听得懂人类说话，工作原理是将人的声音转化为电话虫的声音进行长距离传接，经作者尾田荣一郎在 SBS 上证实这是自然生长的一种虫。



文章	标签	分类
134	37	12

大丙课堂



公告

微信公众号 爱编程的大丙 和
大丙课堂 上线了，可
点击上方 图标关注 ~ ~ ~

三 目录

1. 电话虫
2. 解构电话虫
3. 通话

🕒 最新文章

如果拥有了属于对方的电话虫，不论彼此相隔有多远都可以进行时时通信，通过电话虫除了可以听到对方的声音，还能看到对方的表情，妥妥的一个代理人。在设计模式中有一种模式叫做 **代理模式**，代理模式和电话虫差不多，都是**为其他对象提供一种代理，以控制对这个对象的访问。**

生活中关于代理的例子也有很多，比如：

1. 通过信用卡、微信、支付宝等代替现金支付
2. 开发一套对接数据库服务器的接口并提供给客户使用，用于提高服务器的访问效率
3. 跑腿小哥代替大聪明给异地的女盆友送花。
4. 通过 VPN 架梯子访问外网。

🔗 2. 解构电话虫

如果我们想要用代理模式来描述一下电话虫的行为，里边有如下几个细节：

1. 说话的人是一个对象，电话虫也是一个对象，电话虫模拟的是说话的人
2. **说话的人和电话虫有相同的行为，所以需要为二者提供一个抽象类**
3. **电话虫是在为说话的人办事，所以电话虫和说话人应该有关联关系。**

根据上面的描述，先把对应的 UML 类图画一下：



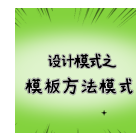
CMake 保姆级教程
(下)
2023-03-15



CMake 保姆级教程
(上)
2023-03-06



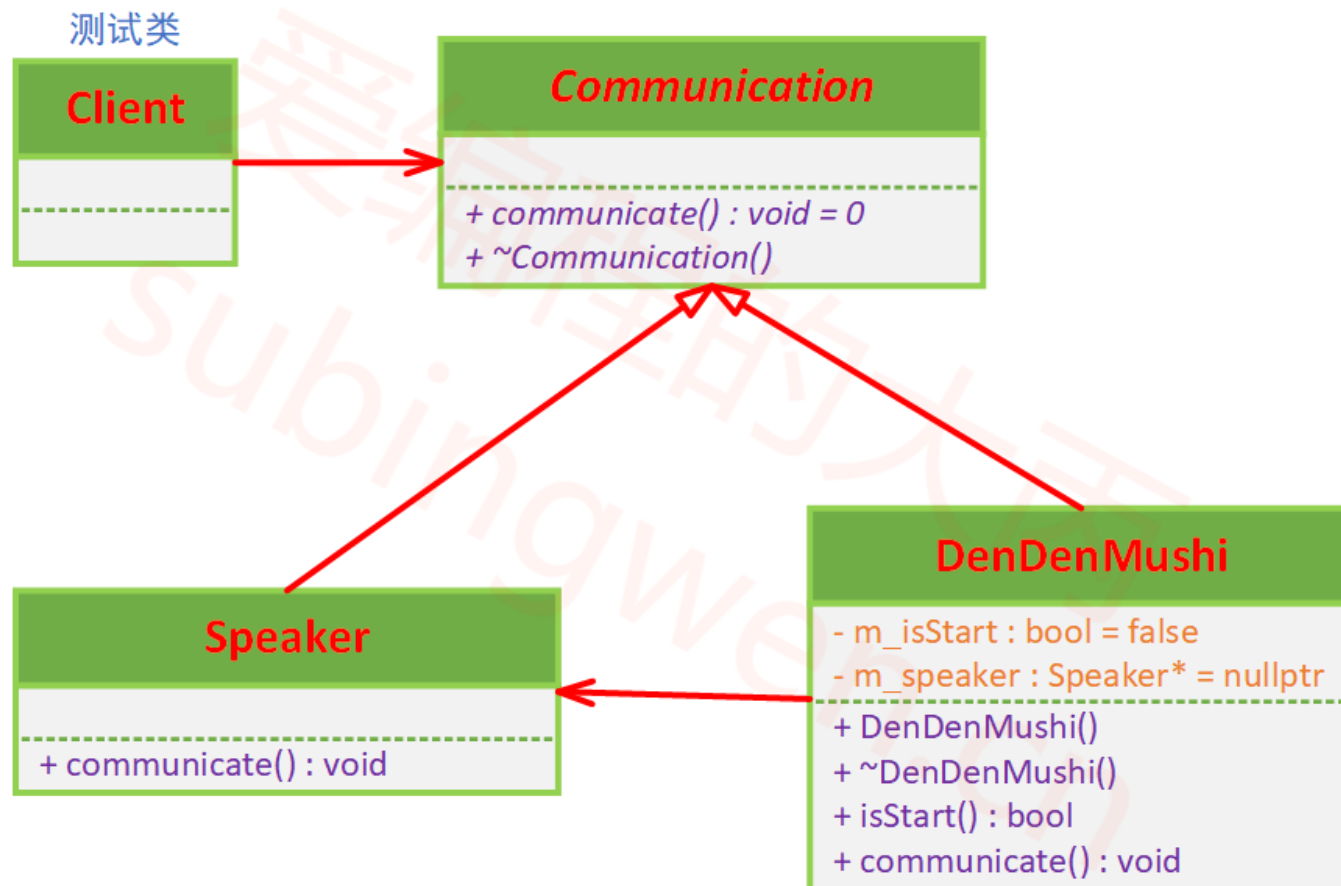
访问者模式 - 再见，
香波地群岛
2022-09-22



模板方法模式 - 和平
主义者
2022-09-21



状态模式 - 文斯莫
克·山治
2022-09-20



由于电话虫类和讲话者类不是部分与整体的关系，所以这二者的关系是关联关系。

3. 通话

根据上面的 UML 类图，先把通话的抽象类定义出来：

▼

C++



```
1 // 抽象通信类
2 class Communication
3 {
4 public:
5     virtual void communicate() = 0; // 通话
6     virtual ~Communication() {}
7 };
```

然后在根据这个抽象类，派生出两个子类：讲话者类和电话虫类：

▼ C++

```
1 // 讲话的人
2 class Speaker : public Communication
3 {
4 public:
5     void communicate() override
6     {
7         cout << "开始说话..." << endl;
8         cout << "通话时发生了一些列的表情变化..." << endl;
9     }
10 };
11
12 // 电话虫
13 class DenDenMushi : public Communication
14 {
15 public:
16     DenDenMushi()
17     {
18         m_isStart = true;
19         m_speaker = new Speaker;
```

```
20     }
21     ~DenDenMushi()
22     {
23         if (m_speaker != nullptr)
24         {
25             delete m_speaker;
26         }
27     }
28     // 判断是否已经开始通话了
29     bool isStart()
30     {
31         return m_isStart;
32     }
33     void communicate() override
34     {
35         if (isStart())
36         {
37             // 得到通话者语言和表情信息，并加以模仿
38             cout << "电话虫开始实时模仿通话者的语言和表情..." << endl;
39             m_speaker->communicate();
40         }
41     }
42 private:
43     bool m_isStart = false;
44     Speaker* m_speaker = nullptr;
45 };
```

海贼官方给出的电话虫的名字叫做 **DenDenMushi**，所以电话虫类也以此命名。

在代理类也就是电话虫类中，一般都会判断是否允许代理（对应示例程序中的 `isStart()`，表示通话对否开始了），如果允许则通过被代理的对象 `m_speaker`，调用它的操作函数

`communicate()`。

最后是测试代码：

```

1  int main()
2  {
3      // 直接交流
4      Communication* comm = new Speaker;
5      comm->communicate();
6      delete comm;
7      cout << "===== " << endl;
8      // 使用电话虫
9      comm = new DenDenMushi;
10     comm->communicate();
11     delete comm;
12
13     return 0;
14 }
```

上面的测试程序中一共使用了两种方式进行通信，第二种使用的是代理模式，我们可以在代理类中有有效的管理被代理的对象的工作的时机，但是并没有改变被代理的对象的行为。



通过测试程序我们可以得到如下结论：如果使用代理模式，不能改变所代理的类的接口，使用代理模式的目的是为了加强控制。

文章作者: 苏丙楦



文章链接: <https://subingwen.cn/design-patterns/proxy/>

版权声明: 本博客所有文章除特别声明外，均采用 [CC BY-NC-SA 4.0](#) 许可协议。转载请注明来自 爱编程的大丙！

设计模式



打赏

上一篇

外观模式 - 桑尼号的加特林炮

“外观模式”

下一篇

设计模式之
桥接模式 - 大海贼时代

👍 相关推荐



评论

昵称

邮箱

网址(http://)

来都来了, 说点什么吧...



提交

来发评论吧~

Powered By [Valine](#)

v1.5.1

©2021 - 2023 By 苏丙楹

冀 ICP 备 2021000342 号 - 1



冀公网安备 13019902000353 号