

Java多线程基础（九）——Thread-Per-Message模式



Ressimix
发布于 2018-07-07

一、定义

Thread-Per-Message模式是指每个message一个线程，message可以理解成“消息”、“命令”或者“请求”。每一个message都会分配一个线程，由这个线程执行工作，使用Thread-Per-Message Pattern时，“委托消息的一端”与“执行消息的一端”回会是不同的线程。

二、模式案例

该案例中，由Host分发请求，每一个请求分发一个新的线程进行处理。**BIO模型**

Host类定义：

```

public class Host {
    private final Helper helper = new Helper();
    public void request(final int count, final char c) {
        System.out.println("    request(" + count + ", " + c + ") BEGIN");
        new Thread() {
            public void run() {
                helper.handle(count, c);
            }
        }.start();
        System.out.println("    request(" + count + ", " + c + ") END");
    }
}
```

Helper类定义：

```

public class Helper {
    public void handle(int count, char c) {
        System.out.println("        handle(" + count + ", " + c + ") BEGIN");
        for (int i = 0; i < count; i++) {
            slowly();
            System.out.print(c);
        }
        System.out.println("");
        System.out.println("        handle(" + count + ", " + c + ") END");
    }
    private void slowly() {
        try {
            Thread.sleep(100);
        } catch (InterruptedException e) {
        }
    }
}
```

执行：

```
public class Main {
    public static void main(String[] args) {
        System.out.println("main BEGIN");
        Host host = new Host();
        host.request(10, 'A');
        host.request(20, 'B');
        host.request(30, 'C');
        System.out.println("main END");
    }
}
```

三、模式讲解

Thread-Per-Message模式的角色如下：

- Client(委托人)参与者

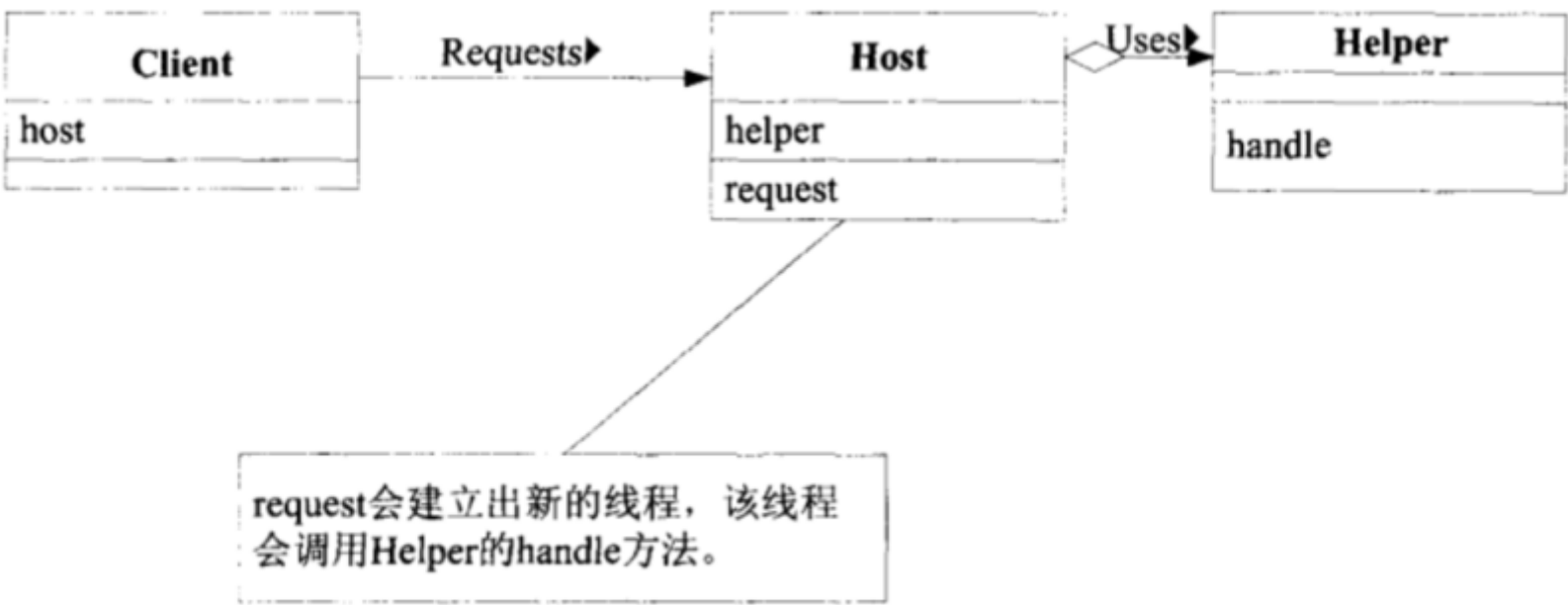
Client参与者会对Host参与者送出请求(Request)。上述案例中，Client参与者就是Main类。

- Host参与者

Host参与者接受来自Client的请求，然后建立新的线程处理它。

- Helper(帮助者)参与者

Helper实际处理请求的。



Thread-Per-Message Pattern 的类图

[多线程](#) [java](#)

阅读 3.5k • 更新于 2018-08-02

👍 赞 2

🔖 收藏 1

🔗 分享

本作品系原创，采用《署名-非商业性使用-禁止演绎 4.0 国际》许可协议



透彻理解Java并发编程

Java并发编程是整个Java开发体系中最难以理解但也是最重要的知识点，也是各类开源分布式框架中各...

关注专栏



Ressmix