

# 委派模式 (delegate)

作用：负责任务的调用和分配任务，可以看做是一种特殊情况下的静态到了的全权代理，但是代理模式注重过程，而委派模式注重结果。

在spring中应用非常多，如DispatchServlet，一般以delegate结尾，或者dispatch。

例子：老板给项目经理下达任务，项目经理会根据实际情况给员工派发任务，待员工完成任务后，再由项目经理汇报工作进度和结果给老板。

Boss类：

```
package dml.delegate;

public class Boss {
    public void work(String thing, Leader leader) {
        leader.work(thing);
    }
}
```

Leader类：

```
package dml.delegate;

public class Leader {
    public void work(String thing) {
        if (thing.contains("java")) {
            new JavaEngineer().work(thing);
        } else {
            new PythonEngineer().work(thing);
        }
    }
}
```

工程师接口：

```
package dml.delegate;

public interface Engineer {
    void work(String thing);
}
```

java开发工程师类：

```
package dml.delegate;

public class JavaEngineer implements Engineer {
    @Override
    public void work(String thing) {
        System.out.println("java开发工程师完成工作: " + thing);
    }
}
```

python工程师类：

```
package dml.delegate;  
  
public class PythonEngineer implements Engineer {  
    @Override  
    public void work(String thing) {  
        System.out.println("python开发工程师完成工作: " + thing);  
    }  
}
```

测试类：

```
package dml.delegate;  
  
public class DelegateTest {  
    public static void main(String[] args) {  
        new Boss().work("java项目", new Leader());  
    }  
}
```

类图：可以看出Boss和工程师是没有直接的联系的，而是由项目经理进行分发任务进行完成。



