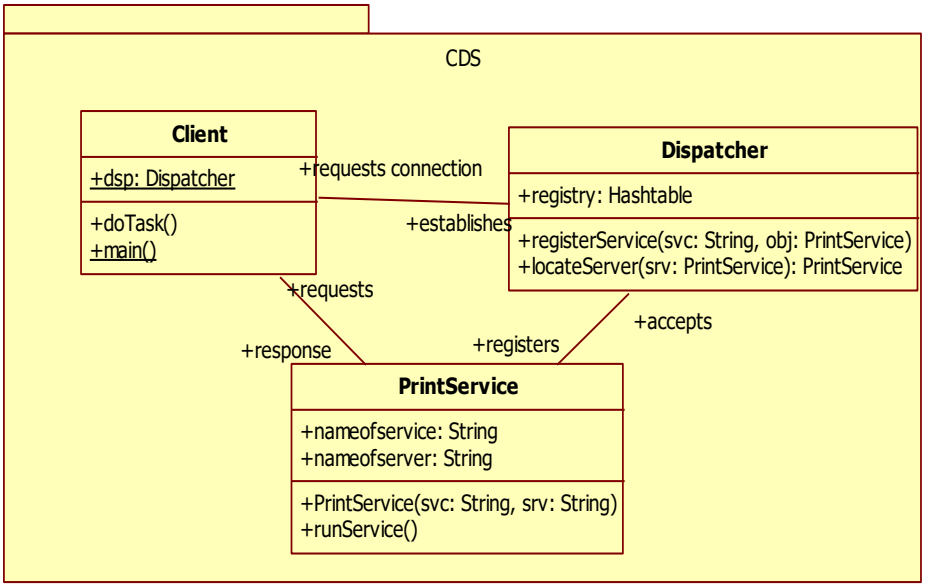
Client-Dispatcher-Server Pattern

1. Client-Dispatcher-Server 패턴 [1]

클라이언트와 서버 간의 통신을 위한 세부 구현을 숨기고 이름 서비스를 제공하기 위해 사용함

- Client: 도메인 분야의 작업을 수행하며, 서버에서 작업하는 서비스에 접근 가능하다.
- Dispatcher: Client와 Server의 통신 채널을 설정하는 것을 담당한다. 서버의 이름을 주소와 실제 위치와 맵핑한다. 서버와의 통신을 설정하고 나서 그 통신 제어를 클라이언트에게 넘긴다.
- Server: Client에 서비스를 제공하며, Dispatcher에 이름과 주소를 등록한다.

2. Client-Dispatcher-Server 패턴 구조 [1]



3. Client-Dispatcher-Server 패턴 사용

코드 템플릿에 있는 Client-Dispatcher-Server 패턴을 사용하는 메인 함수는 다음과 같다.

```
public class TestClient {
	public static void main(String[] args) {
		PrintService s1 = new PrintService("printSvc1","Printer1");
		PrintService s2 = new PrintService("printSvc2","Printer2");
		Client client = new Client ();
		client.doTask ();
	}
}
```

4. Client-Dispatcher-Server 패턴 활용

(1) 코드 텟플릿의 서버의 서비스는 제공되는 서비스 명을 명시할 뿐 아무런 작업도 하지 않는다. 각각에 서로 다른 언어로 인삿말이 나오도록 수정해 보자.

(2) 현재는 클라이언트가 하나이다. 둘 이상의 클라이언트가 둘 이상의 서버에 접속할 때의 템플릿 코드에서의 변경 사항은 어떠한 것들이 있는지 생각해서 나열하자.

5. References

[1] Client-Dispatcher-Server Pattern - https://amcecmcale2014.files.wordpress.com/2015/09/client-dispatcher-serverpattern.docx