

% 각각의 서비스의 특징  
% - 몇개의 리소스를 필요로 하는지에 대한 정보

```
classdef ServiceInfo
```

```
    properties
```

```
        % 서비스 종류는 1,2,3,...,SVC_LIMIT 로 구분함
```

```
        SVC_LIMIT
```

```
        % 서비스 별, 하나의 req 크기/사이즈는 [1,MAX] 에서 임의 선택
```

```
        PerSVC_REQ_SZ_MIN
```

```
        PerSVC_REQ_SZ_MAX
```

```
        % 각 서비스별: 하나의 req 의 크기가 얼마인지
```

```
        inst_size_per_type
```

```
        % 각 서비스는 우선순위를 가짐
```

```
        PRI_MIN
```

```
        PRI_MAX
```

```
        % 각 서비스별: 우선순위가 무엇인지
```

```
        pri_per_svc
```

```
    end
```

```
    methods
```

```
        function obj = ServiceInfo(SERVICE_INDEX_MAX, MIN_PerSVC_REQ_SZ,  
MAX_PerSVC_REQ_SZ, MIN_PRI_LIMIT, MAX_PRI_LIMIT)
```

```
            % MEC 클래스 생성자: 반드시 생성자를 통해서만 클래스  
인스턴스를 생성할 수 있도록...
```

```
            obj.SVC_LIMIT = SERVICE_INDEX_MAX;
```

```
            % 서비스별, 처리에 필요한 리소스 개수 결정하기(= 서비스별 1 개  
REQ 의 크기)
```

```
            obj.PerSVC_REQ_SZ_MIN = MIN_PerSVC_REQ_SZ;
```

```
            obj.PerSVC_REQ_SZ_MAX = MAX_PerSVC_REQ_SZ;
```

```
            obj.inst_size_per_type = randi([obj.PerSVC_REQ_SZ_MIN,  
obj.PerSVC_REQ_SZ_MAX], obj.SVC_LIMIT, 1);
```

```
            % 서비스별, 우선순위 결정하기
```

```
            obj.PRI_MIN = MIN_PRI_LIMIT;
```

```
            obj.PRI_MAX = MAX_PRI_LIMIT;
```

```
            obj.pri_per_svc = randi([obj.PRI_MIN, obj.PRI_MAX], obj.SVC_LIMIT, 1);
```

```
    end
```

```
    function res = getReqResPerSvc(obj, svc)
```

```
    assert(svc >= 1 && svc <= obj.SVC_LIMIT);  
    res = obj.inst_size_per_type(svc);  
end  
end  
end
```