

% msk(masking): 특정 위치에 0/1 값으로 표시함

% - 예를 들어 1,2,3,4 서비스 이미지 중에서 2,3 번만 해당한다면

% 0,1,1,0 으로 표기함

% cntmsk(count masking): 특정 위치에 0,1,2,... 값으로 표시함

% - 예를 들어, 1,2,3,4 서비스 종류 중에서 1 번을 2 개, 4 번을 3 개 받았다면

% 2,0,0,3 으로 표기함

```
classdef MEC
```

```
    properties
```

```
        id % MEC 의 ID 번호
```

```
        installed_svc_msk % 이 MEC 는 어떤 서비스를 처리할 수 있는지
```

```
% 서비스 종류별 REQ 를 각각 몇개씩 받았는지: 이걸 시뮬레이션 할 때,  
그때 그때 설정해 줄 값임...
```

```
        recvd_requests_per_svc_cntmsk
```

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

```
        avail_capa % service capacity
```

```
    end
```

```
    methods
```

```
        function obj = MEC(mec_id)
```

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

```
        obj.id = mec_id;
```

```
    end
```

```
        function obj = setInitialize(obj, MAX_SVC_LIMIT, MEC_CAPA_MIN,  
MEC_CAPA_MAX)
```

```
            obj.SVC_LIMIT = MAX_SVC_LIMIT;
```

```
            obj.installed_svc_msk = zeros(obj.SVC_LIMIT, 1);
```

```
            obj.recvd_requests_per_svc_cntmsk = zeros(obj.SVC_LIMIT, 1);
```

```
            obj.avail_capa = randi([MEC_CAPA_MIN, MEC_CAPA_MAX], 1, 1);
```

```
    end
```

```
        function ret = returnID(obj)
```

```
            % MEC 의 id 값을 리턴하는 메소드
```

```
            ret = obj.id;
```

```
    end
```

```
        function obj = setInstalledSvc(obj, prob)
```

```
            assert(obj.SVC_LIMIT >= 1); % 값이 설정되어 있는지를 먼저 확인
```

