|  |
| --- |
| 전자 제품을 사면 대개의 경우 사용설명서를 함께 줍니다. 사용설명서에는  전자 제품을 처음 사용하는 사람이 어떤 기능을 어떻게 해야 정상적으로  사용할 수 있는지 알려줍니다. 사용하기 복잡한 전자 제품도 사용 순서대로  번호를 붙여가며 작동 순서를 자세히 안내하고 있습니다.   현재 시간을 설정하는 것을 예를 들면 다음과 같습니다. |
| http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |
| |  | | --- | | 1. Set 이라는 단추를 누릅니다.  2. 시간을 Up 혹은 Down키를 눌러 정합니다. 3. Set을 눌러 시간을 확정합니다. 4. 분을 맞추기 Up혹은 Down 키를 눌러 원하는 분을 정합니다. 5. Set을 눌러 분을 확정합니다. | |
| http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |
| 시퀀스 다이어그램은 위와 같은 전자제품의 사용설명서와 비슷한 형태를 가지고 있습니다. 시퀀스 다이어그램은 시간 순서를 중시하고 처리를 위한 객체 간 메시지를 중요시 합니다. 그리고 설명의 대상은 전자제품이 아닌 하나의  유즈케이스입니다.  새로운 모델, 시퀀스 다이어그램... 그 실체를 알기 위해 출발합시다. |

1. 개요

|  |  |
| --- | --- |
|  | http://www.e-campus.co.kr/vmc/3310/3310_09/02/grp/tit0101.gif |
| http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | | --- | | UML이 기존 모델링 체계와 가장 크게 차이나는 점은 기존 모델링 체계에서 제대로 제공하지 못하는 동적 모델링(Dynamic Modeling)기법을 제공한다는 데 있습니다. UML의 동적 모델들 중에 객체와 객체 사이의 동적 상호관계의 정의를 위한 모델에는 시퀀스 다이어그램과 컬레보레이션 다이어그램이 있는데, 이 두 다이어그램을 합쳐서 UML에서는 Interaction Diagram이라고 부릅니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | |  | |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | 시퀀스 다이어그램은 "해결해야 할 문제가 주어진 상황에서 그 문제를 해결하기 위해 필요한 객체를 정의하고, 객체간의 동적인 상호관계를 시간 순서에 따라 정의함으로써 주어진 문제를 해결하는 모델입니다. 시퀀스 다이어그램은 어떻게 하기로 정의된 것을 실제로 그렇게 되도록 실현하는 것입니다.  시퀀스 다이어그램에서는 시간개념이 중시됩니다. 시간은 다이어그램에서 위쪽에서 아래쪽으로 증가하는 것을 전제로 하고 있습니다. 이것은 다이어그램의 수직방향이 시간흐름을 나타냄을 의미입니다. 즉, 아래쪽에 표현된 메시지는 위쪽에 표현된 메시지보다 나중에 발생합니다. 그러면 시퀀스 다이어그램의 이해를 위해 다음의 예를 들어봅시다.  유람선이 인천에서 제주도로 가는 것이 해결해야 할 문제(유즈케이스)라면 이 문제상황은 다음과 같이 하여 실현할 수 있습니다.(시퀀스 다이어그램에 표현되는 내용입니다.) | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  | |  |  | | --- | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o1.gif | 유람선에 연료를 넣습니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o2.gif | 선원들을 태웁니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o3.gif | 승객들을 태웁니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o4.gif | 시동을 겁니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o5.gif | 닻을 올립니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o6.gif | 출항신고를 합니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o7.gif | 배를 출발시켜 속도를 높입니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o8.gif | 항로를 따라 배를 조종합니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o9.gif | 제주항에 배를 접안시킵니다. | | | | |

|  |  |
| --- | --- |
|  | http://www.e-campus.co.kr/vmc/3310/3310_09/02/grp/tit0102.gif |
| http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | |

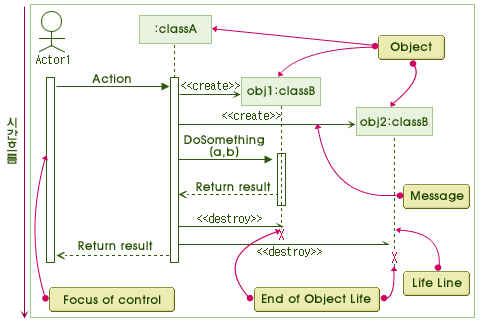
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | --- | | 시퀀스 다이어그램을 작성하는 목적은 다음과 같습니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  |  | | --- | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o1.gif | 객체간의 동적 상호작용을 시간적 개념을 중시하여 모델링합니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | |  | 하나의 객체가 할 수 있는 일은 제한되어 있습니다. 객체지향은 각기 작고 독립적인 일을 하는 객체가 서로 긴밀하게 일을 분담하여 처리함으로써 주어진 문제를 해결하는 방식이 전제되어 있는 체계입니다. 주어진 문제에 관련된 객체는 무엇 무엇인지 그리고 그 문제를 해결하기 위해 객체는 어떤 일을 하고 어떤 일을 다른 객체에게 의뢰해야 하는지를 정의해야 합니다. 시퀀스 다이어그램은 다이어그램의 수직방향이 시간의 흐름을 나타내고 있습니다. 그렇듯이 시간개념을 특별히 중시합니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o2.gif | 객체의 오퍼레이션과 속성을 상세히 정의합니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | |  | 객체간 상호작용을 정의하는 과정에서 객체들이 가져야 하는 오퍼레이션과 속성을 구체적으로 정의할 수 있습니다. 객체는 다른 객체가 의뢰하는 일을 처리해야 합니다. 이것은 객체의 책임(responsibility)으로서, 객체의 책임은 오퍼레이션으로 정의되어야 합니다. 또한 이 행위를 위해 필요한 객체의 속성도 정의되어야 합니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o3.gif | 유즈케이스를 실현(Realization)합니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | |  | 유즈케이스 다이어그램에서는 시스템이 제공해야하는 서비스를 정의합니다. 프로젝트 초기에 정의된 유즈케이스는 프로그램으로 의해 구현되기 전에 시퀀스 다이어그램으로 설계되어야 합니다. 따라서 시퀀스 다이어그램은 각 유즈케이스 별로 작성됩니다. 유즈케이스에 필요한 객체가 주인공으로 등장하고, 객체간의 메시지를 통해서 유즈케이스의 기능이 실현 됩니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o4.gif | 프로그래밍 사양을 정의합니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | |  | 상세화된 시퀀스 다이어그램은 곧바로 프로그래밍될 수 있는 수준으로 정의됩니다. 이렇게 정의된 시퀀스 다이어그램은 훌륭한 프로그램 사양이 됩니다. 일부 케이스 도구에서는 실제로 프로그램이 자동 생성되기도 합니다. | | |
|  | http://www.e-campus.co.kr/vmc/3310/3310_09/02/grp/tit0103.gif | |
| http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |

|  |  |  |
| --- | --- | --- |
|  | |  | | --- | | 시퀀스 다이어그램을 작성하는 시기는 유즈케이스 다이어그램이 정의된 후부터 프로그램 코딩을 하기 전까지입니다. 시퀀스 다이어그램은 다른 다이어그램처럼 여러 번에 걸쳐 정제되는 과정을 거칩니다. 분석 단계에서는 비즈니스 관점에서의 객체가 등장하지만, 설계 단계에서는 구현 관점의 객체들이 등장합니다.  시퀀스 다이어그램을 작성하기 위한 준비물 및 선행과정이 있는데, 이것은 유즈케이스 다이어그램이 작성되어야 한다는 것입니다. | |

 2. 구성요소

|  |  |
| --- | --- |
|  | http://www.e-campus.co.kr/vmc/3310/3310_09/03/grp/tit0101.gif |
| http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | | --- | | 시퀀스 다이어그램은 액터, 객체, 메시지, 기타 등으로 구성되어 있는데, 다음 그림을 보며 자세히 살펴봅시다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o1.gif | Things : 액터(Actor), 객체(Object) | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o2.gif | Relationships : 메시지(Massage) | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o3.gif | 기타 : Life Line , Focus of control | | | |



|  |  |
| --- | --- |
|  | http://www.e-campus.co.kr/vmc/3310/3310_09/03/grp/tit0102.gif |
| http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot1.gif | 액터의 표기 | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  | |  |  |  | | --- | --- | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/03/grp/0102a.gif |  | 액터는 사람 모양의 그림입니다.  작대기로 만든 사람이라는 의미로 "스틱맨"이라는 애칭으로 부르며, 이것은 클래스의 일종입니다. 그리고 스틱맨의 아래에 액터의 이름을 명사로 표시합니다. | | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot1.gif | 액터의 정의 | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  | 시스템의 외부에 존재하면서 시스템과 교류 혹은 상호작용(interaction)하는 것입니다. 자신에 대해 시스템이 서비스를 해주기를 요청하는 존재들이자, 시스템이 정보를 제공하는 대상이기도 합니다. 그리고 시스템의 일부가 아니라 외부에 독립적으로 존재합니다. 이것은 또한 시퀀스 다이어그램을 시작하게 하는 역할을 하고, 유즈케이스 다이어그램에서 액터와 유즈케이스 간의 communicates 관계는 시퀀스 다이어그램에서 객체에 다수의 메시지를 보내는 형태로 상세화됩니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot1.gif | 액터의 예 | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |



|  |  |
| --- | --- |
|  | http://www.e-campus.co.kr/vmc/3310/3310_09/03/grp/tit0103.gif |
| http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot1.gif | 객체의 표기 | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  | 다음 표기 예들을 살펴보면, 객체는 사각형 모양으로 표기한다는 것을 알 수 있습니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  | |  |  | | --- | --- | | 표기 예 | 의미 | | |  | | --- | | 홍길동:사원 | | : 사원 클래스의 객체중 홍길동을 의미 | | |  | | --- | | 홍길동 | | : 홍길동 객체 (클래스는 아직 정의되지 않았음) | | |  | | --- | | :사원 | | : 사원 클래스의 일반객체 (일반적인 객체를 지칭) | | |  | | --- | | 홍길동:사원 | | 사번=16009 이름=홍길동 급여=4000 성별=남 직위=과장 | | : 사원 클래스의 홍길동 객체 (객체 속성을 모두 표현) | | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  | 이렇게 객체명은 "객체명:클래스명"으로 사각형 내에 표기합니다. 객체명의 표기 중 "객체명"(클래스명 생략), ":클래스명"(객체명 생략,일반 객체를 표현) 등으로 생략되어 표기되기도 합니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot1.gif | 객체의 정의 | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  | 객체는 클래스의 인스턴스이며, 시스템에서는 해당 클래스 타입으로 선언된 변수의 형태로 존재합니다. 그 예와 해설은 다음과 같습니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  | |  |  | | --- | --- | | 예 | Class 사원 홍길동; | | 해설 | 사원 클래스의 객체로서 홍길동이 선언되었습니다. | | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  | 그리고 객체는 생성되고 소멸되기까지의 생명주기 동안 다양한 상태의 변화를 겪습니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot1.gif | 객체의 예 | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  | 객체의 예는 다음과 같습니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot2.gif | 사과 : 과일 | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot2.gif | 헬리콥터 : 운송수단 | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot2.gif | 계약자 : 고객 | | | | |
|  | http://www.e-campus.co.kr/vmc/3310/3310_09/03/grp/tit0104.gif | |
| http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | | --- | | 메시지는 객체지향 패러다임에서 객체와 객체가 통신하는 유일한 수단입니다.  객체지향에서 필수적인 객체간 협력작업이 가능하기 위해서는 객체들은 서로 메시지를 통해서 상호작용을 해야 합니다. 이러한 메시지는 다양한 종류로 존재합니다. 그리고 메시지는 향후 클래스의 오퍼레이션으로 구현됩니다. 메시지의 종류는 다음과 같습니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  |  | | --- | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot1.gif | Flat flow of control | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  | |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/03/grp/0104a.gif |  | 가장 일반적인 메시지 형태로, 객체에 메시지를 연결할 때 사용합니다. 그리고 열린 화살표가 한쪽에 붙은 실선으로 표기합니다. | | | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot1.gif | Nested flow of control | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  | |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/03/grp/0104b.gif |  | procedural call과 같은 의미로 사용되며 메시지가 중첩(nested)되는 경우에는 내부 메시지의 결과가 모두 돌아와야 가장 처음 시작한 메시지의 결과가 되돌려지게 됩니다. 이것은 메시지의 결과가 돌려지게 될때까지 다음 처리를 진행하지 않는  Synchronous 메시지입니다. 그리고 닫힌 화살표가 한쪽에 붙은 실선으로 표기합니다. | | | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot1.gif | Asynchronous flow of control | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  | |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/03/grp/0104c.gif |  | 객체가 보낸 메시지의 결과를 기다리지 않고 다음 처리를 진행할 경우에 사용하며, 위쪽의 반쪽  화살표가 붙은 실선으로 표기합니다. | | | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot1.gif | Return flow | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  | |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/03/grp/0104d.gif |  | 메시지를 처리한 결과(return)를 의미하며, 꼭 필요한 경우에 표현합니다. 그리고 화살표가 달린 점선으로 표기합니다. | | | | |

|  |  |
| --- | --- |
|  | |
| http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | http://www.e-campus.co.kr/vmc/3310/3310_09/03/grp/tit0105.gif |
| http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/03/grp/0105a.gif |  | http://www.e-campus.co.kr/vmc/3310/3310_09/03/grp/0105b.gif | | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | Life Line은 객체의 생존 기간을 의미하며, 객체에 붙은 수직방향의 점선으로 표기합니다. 그리고 점선에 X표시가 덧붙여질 경우, 이 시점이 객체의 소멸시점입니다.  Activation는 객체가 활성화 되어 있는 기간을 의미하며, 객체가 외부의 메시지를 받고, 다른 객체에 보낸 메시지에 대한 return flow를 기다리는 기간을 의미합니다. 그리고 Life line 위에 좁고 세로로 긴 사각형으로 표기합니다. 처리가 끝나고 대기하는 시간은 일반적인 life line 표기로 합니다. | |

  3. 사례연구

|  |  |
| --- | --- |
|  | |
| http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | http://www.e-campus.co.kr/vmc/3310/3310_09/04/grp/tit0101.gif |
| http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | --- | | 다음은 어느 음식점의 요리주문에 대한 시퀀스 다이어그램의 간단한 사례입니다.  다이어그램을 잘 살펴보고 의미하는 바를 해석해 봅시다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  | | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/yungu.gif | | |  | | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/04/grp/0101a.gif | | | | |
|  | 이 기능은 세 액터가 사용합니다. | |
| http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |
|  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o1.gif | 손님이 메뉴판을 보고 메뉴를 고릅니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o2.gif | 손님이 웨이터에게 메뉴를 정해 알려줍니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o3.gif | 웨이터는 주문된 요리를 입력합니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o4.gif | 주방장이 요리예약을 조회합니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o5.gif | 주방장이 요리를 제작합니다. | | | |

|  |  |
| --- | --- |
|  | http://www.e-campus.co.kr/vmc/3310/3310_09/04/grp/tit0102.gif |
| http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | | --- | | 다음의 시퀀스 다이어그램은 "사원정보 등록"이라는 유즈케이스에 대한 것입니다. 다이어그램을 잘 살펴보고 의미하는 바를 해석해 봅시다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  | | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/yungu.gif | | |  | | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/04/grp/0102a.gif | | | |

보통 초기단계의 시퀀스 다이어그램에서는 화면을 제어하는데 관련된 객체  
(위 사례에서 인사기본정보 Form)이나 객체 사이의 순서를 통제하거나 트랜잭션  
처리를 담당하는 객체(위 사례에서 사원 Manager)는 등장하지 않습니다. 이러한  
객체는 설계단계를 거치면서 정의됩니다. 즉, 설계관점에서 Boundary 클래스  
(인사기본정보 From)와 Control 클래스(사원 Manager)가 추가되었음을 알 수  
있습니다.

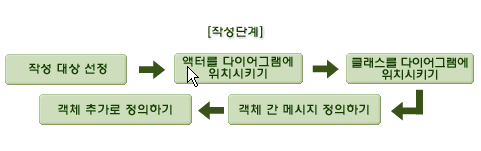
|  |  |
| --- | --- |
|  | http://www.e-campus.co.kr/vmc/3310/3310_09/04/grp/tit0103.gif |
| http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | --- | | 실제 현실에서 경험할 수 있는 사례를 가지고 모델링을 수행한 결과를 살펴보도록 하겠습니다.  아래에 기술된 문제영역은 모델링의 대상이 되는 주제와 이것과 관련된 주변정보를  제공합니다. 이 문제영역을 분석하여 시퀀스 다이어그램을 작성해 보세요. 그런 다음,  다이어그램 보기 버튼을 클릭하여 시퀀스 다이어그램과 비교해 보시기 바랍니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  | | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/yungu.gif | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  | | --- | | A병원은 이번 기회에 SW 시스템을 구축하여 업무의 일부를 자동화 하기를 원합니다. A병원에는 기존 시스템으로 병원의 수입과 지출을 관리하는 "회계시스템"이 운영되고 있습니다. 이 시스템과 연계하면서 병원정보, 환자의 병력 및 진료정보, 의료진 정보의 각종 정보관리와 진료예약 처리 및 수납지원을 수행하는 신규 시스템을 구축하기를 원하고 있으며, 병원의 각 관계자는 시스템을 통해 아래와 같은 기능을 수행하기를 원합니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot2.gif | 진료비는 진료정보를 입력하면 자동 산정됩니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot2.gif | 환자는 직접 진료예약을 하고, 환자의 과거병력과 진료정보는 관리됩니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot2.gif | 일반사용자는 병원 정보와 의료진 정보를 조회하고 상담 및 문의를 할 수 | |  | 있습니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot2.gif | 의료진은 자신의 진료스케쥴을 자동 생성하고, 진료 내역을 관리하고, 환자 | |  | 정보를 조회하기를 원합니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot2.gif | 원무과 직원은 이 시스템을 통해 진료비 청구서를 조회,발행하고, 진료예약을 | |  | 확정하기를 원합니다. | | | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | 본 진료시스템의 유즈케이스 중 하나는 진료예약입니다. 진료예약에 대해 생각해 보고 그 결과를 바탕으로 시퀀스 다이어그램을 작성해 보십시오.(위 문제영역 설명에서 제시되지 않은 구체적인 상황은 각자 일반적 병원 상황을 상상하며 설정해 보시기 바랍니다.) | | | | |
|  | | |
|  | 아래의 다이어그램에서는 여러분의 이해를 돕기 위해 한글로 클래스 명과 메시지 명을 정의하였습니다. 실제로 설계할 경우 상세 설계에서는 모든 명칭이 영문으로 정의됩니다. 왜냐하면 프로그램으로 쉽게 구현할 수 있기 때문입니다. 그리고 상세 설계에서는 실제 구현할 로직을 머릿속에 그리면서 최대한 자세한 메시지가 나타나도록 정의해야 합니다 | |
| http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |
|  | http://www.e-campus.co.kr/vmc/3310/3310_09/04/grp/0103a.gif | |

 4. 작성단계 및 주의사항

|  |  |
| --- | --- |
|  | http://www.e-campus.co.kr/vmc/3310/3310_09/05/grp/tit0101.gif |
| http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | |

|  |  |  |
| --- | --- | --- |
|  | |  | | --- | | 먼저, 시퀀스 다이어그램의 작성순서를 알아보겠습니다. | |



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | --- | |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | |  | |  | | |  |  | | --- | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o1.gif | 작성 대상을 선정합니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | |  | 유즈케이스 다이어그램을 이용하여 컬레버레이션 다이어그램의 작성대상을 선정한 후, 하나의 유즈케이스를 선택하고, 유즈케이스 정의서를 분석합니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o2.gif | 유즈케이스의 액터을 파악하여 컬레버레이션 다이어그램에 위치시킵니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | |  | 액터가 둘 이상일 경우라도 모두 좌측부터 액터를 위치시켜야 합니다. 순서는 중요하지 않습니다만, 메시지 선이 적게 교차하도록 배치시키는 것이 좋습니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o3.gif | 유즈케이스를 실현하기 위해 참여할 클래스(객체)들을 정해 시퀀스 | |  | 다이어그램에 위치시킵니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | |  | 정의된 클래스 중에 유즈케이스의 처리에 참여하는 것들을 식별하고 시퀀스 다이어그램에 위치시켜야 하지만 순서는 중요하지 않습니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o4.gif | 시간 순서를 감안하여 객체간 메시지를 정의해 나갑니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | |  | 유즈케이스를 실현하기 위해 필요한 객체들 간의 메시지를 정의하되, 시간 순서에 유의하도록 합니다. (시간흐름은 위에서 아래로) | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/o5.gif | 필요한 객체를 추가로 정의합니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | |  | 요구된 처리를 위해 필요하지만 아직 정의되지 않은 객체를 새로 정의하여 시퀀스 다이어그램에 추가해 나갑니다. 또 이렇게 추가된 객체 사이의 메시지도 정의하여 추가합니다. | | |
|  | http://www.e-campus.co.kr/vmc/3310/3310_09/05/grp/tit0102.gif | |
| http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | | --- | | 다음은 시퀀스 다이어그램 작성시 주의사항을 알아보겠습니다.  시퀀스 다이어그램은 유즈케이스 별로 하나씩 작성합니다. 하지만 경우에 따라서 하나의 유즈케이스를 여러 개의 시퀀스 다이어그램으로, 또는 여러 유즈케이스에서 공통으로 사용하는 상호작용을 하나의 시퀀스 다이어그램으로 작성할 수도 있습니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif | | |  | | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/attention.gif | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot2.gif | 동일한 상호작용을 여러 시퀀스 다이어그램에서 중복되게 작성하는 것을 | |  | 피합니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot2.gif | 중복을 최소화 시키기 위해서 UI별 시퀀스 다이어그램을 작성해도 됩니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot2.gif | 시퀀스 다이어그램의 가독성이 좋도록 적당한 코멘트(comment)를 | |  | 사용하여야 합니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot2.gif | 시퀀스 다이어그램상에서 메시지의 흐름은 액터로부터 시작되게 작성합니다. | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/tlmargin.gif |  | | http://www.e-campus.co.kr/vmc/3310/3310_09/00/grp/dot2.gif | 클래스 다이어그램에 표기된 클래스명과 매핑 가능하도록 객체 이름을 | |  | 표기하여야 합니다.(단, 시퀀스 다이어그램에 나타나는 객체는 클래스 다이어그램에서 정의된 클래스와 매핑이 가능해야 합니다.) | | | | |