UML이란 무엇인가?

20512

김택서

INDEX

1. UML이란?
2. UML의 종류, 예시
3. 사용하는 이유
4. UML이란?

UML(통합모델링언어)란 객체지향 소프트웨어를 개발할 때 산출물을 명세화 시각화 문서화 할 때 사용합니다 모델이란 어떤것을 실제로 제작에 들어가기전 미리 검증해보는것이며 바로 제작에 들어갈 때보다 소모되는 자원과 실제 제작에 소요되는 시간을 줄일 수 있는것이 장점입니다 하지만 다른 분야의 모델링과는 다르게 소프트웨어 개발에 있어서의 모델링은 그렇지 아니할때보다 비 효율적일수도 있는데 다음과 같을때입니다

1. 개발을 혼자 진행하게 될 때
2. 소프트웨어의 크기가 작거나 간단할때

다음과 같은경우에는 UML을 제작하기위한 자원이 바로 제작에 들어갈 때보다 더 많이 소모되기 때문에 UML을 작성할 필요가 없어집니다

1. UML의 종류, 예시

정식으로는 총 14개의 UML다이어그램 타입이 있다 크게 두 범주로 나눌때는 구조다이어그램과 행동다이어그램으로 나눌 수 있는데 구조다이어그램은 모델링되는 시스템에 존재해야하는 사항을 강조합니다 따라서 이러한 구조의 다이어그램은 시스템의 구조를 나타내는데에 적합하기 때문에 소프트웨어 시스템의 구성요소별로 분리되는 방법을 설명하는데 적합합니다

행동다이어그램은 모렐링되는 시스템에서 어떠한 일이 발생하는지를 강조합니다 이는 시스템의 동작을 보여주기 때문에 시스템의 기능을 설명하는데 사용됩니다그중 수업시간중에 예시로 제시된 구조다이어그램에 속하는 클래스 다이어그램과 순서도와 비슷한 형태를 가진 행동 다이어그램인 액티비티 다이어그램, 마지막으로 하드웨어와의 자원연결을 표시하는 구조다이어그램인 디플로이먼트 다이어그램을 소개합니다 다음은 각각의 다이어그램에 사용될 예시 코드입니다

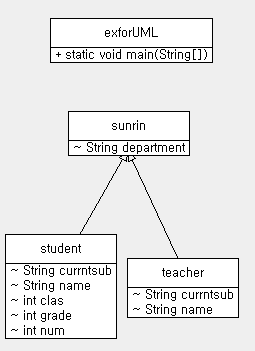
|  |
| --- |
| [코드] |
| package reported;  class sunrin{  String department;  }  class student extends sunrin{  int grade;  int class;  int num;  String name;  String currntsub;  }  class teacher extends sunrin{  String name;  String currntsub;  }  public class ex1sourcecode {    public static void main(String[] args) {  // TODO Auto-generated method stub  student taekseo = new student;  teacher jungtaeSam = new teacher;  taekseo.department="web";  taekseo.grade = 2;  taekseo.num=12  taekseo.class = 5;  taekseo.name = "택서";  taekseo.currntsub="club activity";  jungtaeSam.currntsub="unknown";  jungtaeSam.name = "정태쌤";  jungtaeSam.department="web";  }    } |

2.1클래스 다이어그램

클래스다이어그램은 수업시간에 보았던 정적인 UML이다 각각의 클래스로 구분지어지는 칸 안에는 클래스안에 들어가는 메소드와 변수들로 나타내며 그 이름옆에는 반환형과 접근형식(private, public, protected)을기재한다 클래스간의 관계는 칸과 칸사이를 실선과 점선등의 화살표를 이용하여(그림 참고)

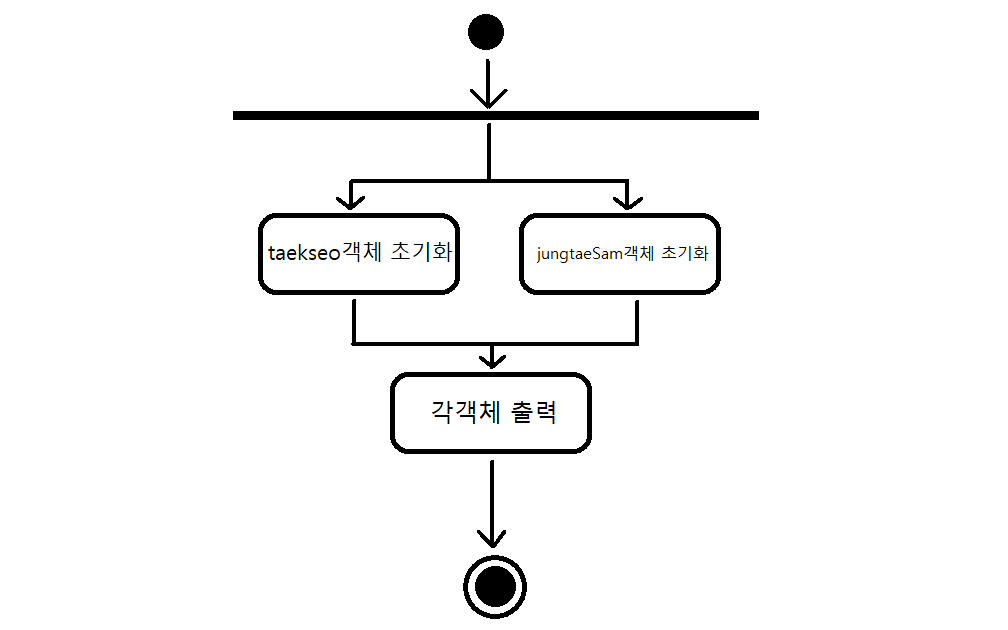
나타낸다 이 다이어그램은 직접적인 설계사양을 제공하며 논리, 물리적인 관점까지 일관된 형식으로 시스템을 설계할 수 있습니다

다음은 위 예제코드로 작성된 클래스 다이어그램입니다



2.2액티비티 다이어그램

액티비티 다이어그램은 순서도와 비슷한 형태를 가진다 행동다이어그램인 만큼 위의 구조다이어그램과 추구하는 목적이 다르며 모델링되는 프로그램의 실행 또는 행동하는 순서와 기능이 부각되어집니다 작성법은 프로그램의 시작점과 끝점인 initialState와 finalState를 생성한후 조건부와 동시경로 등 을 이용하여 프로그램의 실행 순서에 따라 작성하면 됩니다 다음은 위 작성된 예시코드를 액티비티 다이어그램에 따라 작성한 것 입니다



2.3디플로이먼트 다이어그램

디플로이먼트 다이어그램의 특징은 시스템의 물리적인 구성을 나타낸다는 것 입니다 디플로이먼트다이어그램은 실제 하드웨어적인 배치와 연결상태를 나타내며 각각의 노드는 처리능력을 가진 하드웨어를 의미합니다 주로 모니터 서버 DB서버 클라이언트 라우터등이 사용됩니다

1. 사용하는 이유

UML을 설명할 때 언급했듯이 가장 큰 장점은 실제 프로그램의 제작과정에 바로 들어갈 때보다 미리 계획도를 그려봄으로써 실제 제작에 들어가는 시간과 자원을 절약할 수 있다는 점입니다 하지만 프로그램의 크기나 복잡도가 UML을 작성하지 않아도 될 정도의 프로젝트라면 굳이 UML을 작성 할 필요가 없습니다 UML을 사용하는 이유는 단지 자원의 소모를 절약하는 것 뿐만이 아니라 다른 관계자에게 이해를 돕고 개발 팀원간의 프로그램개발 방향을 잡는 데에도 도움이 되는데 이는 팀 개발에 있어서 개발 역할 분담과 개발 능률향상에 있어서 필수적이라고 할 수 있습니다

참고

<http://www.nextree.co.kr/p6753/> (클래스 다이어그램)

<http://sfeg.tistory.com/339> (다이어그램 정리)

<http://hersheysman.tistory.com/204> (다이어그램 정리, 사진)