

9 주차 (챕터 8)

1. 객체지향 모델링의 특성을 이해한다.
2. 객체지향 모델링을 위한 UML 다이어그램의 종류를 알아본다.
3. **객체지향 기반의 기능 모델링, 구조 모델링, 행위 모델링을 이해한다. (중요해보임)**
4. 객체지향 분석 모델 간의 일관성 점검 대상을 이해한다.

1. 객체지향

특정 기능을 수행하기 위한 데이터와 데이터를 사용하는 연산을 하나로 묶어서 객체로 표현하는 접근 방법

객체지향의 특징

1. 클래스와 오브젝트

클래스는 실세계의 사물을 정의하기 위한 틀을 제공한다. 클래스(틀)로부터 생성되는 인스턴스가 객체 즉, 오브젝트이다. 오브젝트는 클래스 하나에서 여러 개 생성할 수 있으며 실제로 연산에 사용된다.

2. 캡슐화와 정보 은닉

클래스는 속성과 연산으로 정의하고 포장된다. 이렇게 포장하는 것, 껍질을 씌우는 것이 캡슐화이다. 캡슐화로 인해 클래스의 속성과 연산이 감춰지며 필요한 최소한의 정보만 외부에 공개(선언)된다.

3. 상속

구체(Concrete) 클래스들의 공동 속성과 공통 연산을 뽑아내어 추상 클래스로 선언한다. 추상 클래스에 정의된 속성은 구체 클래스에서 정의하지 않고 추상 클래스로부터 상속받는다.

4. 다형성

클래스에 정의된 요소들은 실제 클래스의 오브젝트가 사용되는 시점에서 행위가 결정된다. 따라서 동일한 속성과 연산이 클래스에 중복적으로 정의될 수 있으며, 어떤 시점에서 어떤 객체에 사용되는가에 따라 서로 다른 행위를 갖는다.

UML 모델링 언어

UML(Undefined Modeling Language) - 다이어그램을 표현하기 위한 기본 모델 요소를 표준적으로 정의

SW 개발 생명 주기 단계별 다이어그램

분석 단계 - 유스 케이스, 클래스, 순차 다이어그램

설계 단계 - 패키지, 타이밍, 순차(메소드 명세) 다이어그램

2. 기능 모델링

사용자로부터 도출한 요구사항을 입력 받아 소프트웨어 시스템이 해야 할 기능이 무엇인지를 식별해가는 과정. UML의 유스 케이스 다이어그램을 이용하여 분석 결과를 표현한다.

요구사항에 대한 충분한 이해가 부족하거나 도메인 지식이 부족한 경우 비즈니스 프로세스 모델링을 액티비티 다이어그램을 이용하여 수행한다. (액티비티 다이어그램 중요하지는 않음)

유스 케이스 다이어그램(중요!!!)

유스케이스 다이어그램은 시스템 최상위 수준에 해당하는 기능을 사용자 관점으로 나타낸 것이다. 사용자가 시스템을 통해 제공받는 주요 기능, 시스템과 사용자 간의 어떤 상호작용이 이루어지는지 표현

작성 절차(중요!)

요구사항 정의 내용 검토

요구사항을 토대로 시스템의 경계를 결정

경계가 설정된 시스템에 대한 명칭을 정의

시스템 외부에 존재하는 액터Actor를 식별하고, 각 액터의 역할 정의

각 액터가 시스템에서 사용하는 기능을 식별

식별된 사용자 기능과 액터 간의 관계를 정의

식별된 사용자 기능에 대한 설명서를 작성

액터

외부에 존재하는 상호작용 대상 시스템, 외부 장치, 연동하는 하드웨어, 타이머 등 개발 대상 시스템과 상호작용하는 모든 대상을 역할 중심으로 식별한다.

유스 케이스 식별 고려사항

1. 시스템이 제공하는 기능을 나타내야한다.
2. 사용자가 한 장소에서 일정한 시간에 수행하는 상호 구별되는 비즈니스 프로세스여야한다.
3. 서로 다른 시간대에 수행되는 행위는 분리하여 각각 별도의 유스 케이스로 식별
4. 유스 케이스의 이름은 동사구 형태로 기술한다.
5. 유스 케이스는 최초에 액터에 의해 시작된다.

유스 케이스 다이어그램에서 관계

포함 관계: 하나의 유스 케이스에서 반드시 호출되어 실행되어야 하는 동작.

확장 관계: 하나의 유스 케이스에서 선택적으로 호출되는 관계

유스 케이스 설명서(그닥?)

유스 케이스 식별부: 유스 케이스에 대한 ID, 이름, 간단한 설명 다른 유스케이스와의 관계 등에 대한 정보를 기술

정상 시나리오 정의부: 유스 케이스 내부 기능을 단계별로 관찰자 입장에서 액터와 시스템간의 상호작용에 초점을 맞춰서 작성한다.

예외 처리부: 정상 시나리오의 각 스텝에서 발생 가능한 예외 처리를 작성하는 부분

3. 구조 모델링

개발하려는 소프트웨어가 어떤 구조적 요소들로 이루어질 수 있는지 분석하고 설계 단계에서 사용
생성되고 사용되어야 할 객체에 어떤 것들이 있는지, 그들 간의 관계를 식별하고 정의한다.

UML의 클래스 다이어그램으로 표현한다.

UML 정의 클래스 - 클래스 명, 클래스 속성, 클래스 연산

식별 방식

1. 문장 분석 - 유스 케이스 설명서의 정상 시나리오와 예외 처리 시나리오에 나타난 문장들의 구성요소를 분석한다.
2. 일반 객체 목록 - 도메인별로 나타나는 일반적인 용어들을 식별하여 목록화한 후에, 이를 구조 모델의 클래스로 사용한다면 객체 식별이 용
이해짐
3. 브레인 스토밍 - 사전에 검토하지 않은 채, 프로젝트 관련자들의 생각으로부터 객체를 추출한다.
4. 패턴 적용 - 빈번히 발생하는 실세계의 문제를 해결하기 위하여 체계적으로 작성된 클래스와 이들 간의 협업 관계를 정의한다.

클래스 명세

CRC 카드를 이용하여 작성한다.

클래스 다이어그램 작성

식별된 모든 클래스를 CRC 카드에 명세한 후에 클래스 다이어그램을 작성한다.

일반화 관계

식별된 클래스들을 추상화하여 보다 보편적인 클래스로 나타냄
추상 클래스와 구체 클래스 간에 “Kind of” 관계 형성

집합 관계

구성 또는 조합 관계
집합체 클래스와 구성 클래스 간에는 “Part of” 관계 형성

연관 관계

두 클래스 간의 관계가 일반화 관계와 집합 관계에 속하지 않는 경우

클래스 다이어그램에 표현되는 추가 정보

속성 가시성: 클래스의 속성으로 정의되는 변수가 갖는 접근 범위 Public(+), Private(-), Protected(#)

관계 기수성: 두 클래스 간에 몇 개의 인스턴스, 즉 객체가 발생할 수 있는 가를 나타낸다.(1:N)

4. 행위 모델링

시스템의 내적인 동작을 표현하는 다이어그램, 객체 간에 어떤 상호작용이 발생하는 지를 나타낸다.

시스템의 내적 동작에 대한 모형화

순차 다이어그램

기능 모델링의 결과로 식별된 유스 케이스 별로 작성하는 다이어그램
유스 케이스의 개수와 순차 다이어그램의 개수는 일반적으로 동일하다.

순차 다이어그램의 구성 요소

객체 간의 상호작용, 즉 메시지 패싱에 대한 행위를 시간 축 중심으로 나타낸다.

참여 객체: 한 클래스의 객체가 여러 개 존재할 수 있음다. 참여 객체 간의 상호작용은 왼쪽에서 오른쪽으로 진행하도록 표현한다.

시간 축: 다이어그램의 세로축, 메시지 전달 순서는 위에서 아래 방향으로 나타내며 업무의 처리 흐름과 일치해야한다.

실행 사건: 실행 동작이 발생하는 기간을 길쭉한 사각형의 실행 사건으로 표현

메시지 전달: 객체 간의 상호작용을 나타내는 기본 요소, 객체 간의 함수 호출을 의미

제어 로직: 소프트웨어의 행위를 표현한다. 선택, 반복, 병렬 처리 등

순차 다이어그램 작성 (그닥?)

순차 다이어그램을 작성할 유스 케이스를 검토한다.

유스 케이스의 정상 시나리오 부분의 철저한 분석과 이해를 필요로 한다.

산출물 간 일관성 점검(0.5)

일관성 점검: 설계 과정에 들어가기 전에 분석 결과물에 대한 일관성 점검을 실시한다.

기능 모델과 구조 모델

기능 모델의 유스 케이스 설명서와 CRC 카드 간의 일관성을 점검한다.

모든 클래스의 이름은 반드시 유스 케이스 설명서의 어딘가에 나타나야한다.

기능 모델과 행위 모델

기능 모델의 유스 케이스는 각각 하나의 순차 다이어그램으로 표현
유스 케이스 다이어그램의 액터는 순차 다이어그램의 액터로 나타나야한다.

구조 모델과 행위 모델

상태 기계 다이어그램은 클래스의 인스턴스와 매핑
순차 다이어그램에서 객체 간에 전달되는 메시지는 반드시 클래스의 연산으로 정의되어야 한다.