

13주차 1차시 분석론 기초사항 고찰

【학습목표】

1. 객체의 개념 및 문제정의를 설명할 수 있으며, 객체의 식별과 결정을 파악할 수 있다.
2. 자료사전으로 표현되는 객체의 속성기술을 설명할 수 있다.

학습내용1 : 개요 / 문제정의

1. 개요

<실세계(Real world)의 내용을 객체모델(Object model)로 작성하는 과정이 객체지향분석(Object Oriented analysis)임>
- 객체모델을 효과적으로 유도하기 위해서는 다음과 같은 과정을 거쳐야 함

- ① 객체의 식별
- ② 객체 사이의 속성과 연산의 정의 및 기술(Description)
- ③ 객체 사이의 통신 정의

1) 다양한 객체분석 절차가 존재하지만 요든(Yourdon) · 코드(Coad)에 의하면 다음과 같은 5 단계로 구분됨

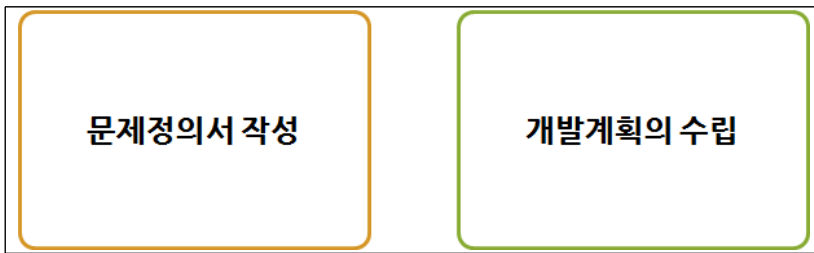
- ① 문제명세서 작성
- ② 객체 · 클래스 정의
- ③ 클래스 속성 정의
- ④ 클래스의 계층구조 결정
- ⑤ 메시지 정의

2. 문제 정의

1) 객체지향방법론으로 문제를 해결하기 위한 작업진행은

- ① 문제정의(Problem Statements)
- ② 객체 및 클래스 정의
- ③ 클래스의 속성 · 연산 정의
- ④ 관련된 클래스 · 객체들의 관계 설정
- ⑤ 각 클래스의 연산을 구체화한 알고리즘 결정 순으로 이루어짐

2) 개발계획단계에서는 다음 두 가지가 선결과제임



3) 문제 정의(Problem Statements)의 목적과 내용 및 수정

- ① 문제 내용을 구체적으로 정의하고 범위를 명확히 구분하기 위하여 작성함
- ② 대상업무를 개발하기 이전에 고객에 의해서 시스템의 기능, 사용자의 요구 성능을 구체적으로 명시하는 것임
- ③ 고객이 작성한 문제정의 내용이 부실할 경우 개발자가 수정보완 가능함

4) 문제정의 기술(Description) 과정의 유의사항

- ① 간단 · 명료한 문체로 기술해야 함
- ② 질문문 · 명령문 · 감탄문 등을 피하고 선언하는 문장으로 기술함
- ③ 필요한 모든 기능 · 요구되는 중요 정보 및 과정의 포함 여부를 확인해야 함
- ④ 각 기술문장이 중요도 · 상세도 측면에서 동일 수준 여부를 확인해야 함

* 예시 : 「문제 정의」내용

- 각자 참고서에 나타난 내용 수집하여 비교 · 검토하는 시간 을 가지도록 강의를 진행함

5) 문제정의서 작성 「문제정의」된 내용을 요약하여 간단하게 「문제정의서」를 작성해 두어야 함

- 문제정의를 완벽하게 해두면 그 내용에서 클래스 객체 · 함수를 찾아내는 일이 용이하기 때문에 문제정의서를 작성함

* 예시 : 문제정의서 작성

- 각자 사례를 수집하여 장단점을 비교·검토하도록 강의를 진행하도록 유도함

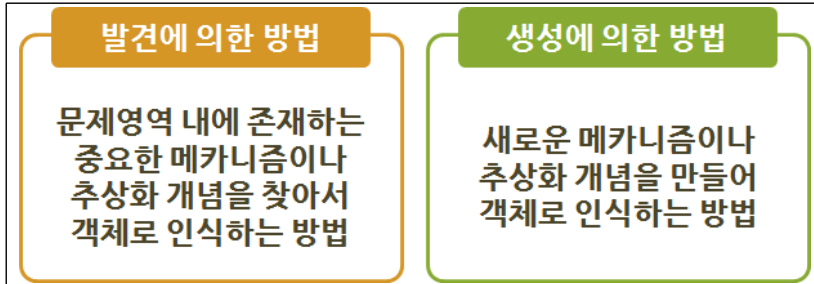
학습내용2 : 객체의 식별과 결정 / 구조결정

1. 객체의 식별과 결정

1) 객체의 식별

① 객체의 식별(Identification) 방법은 두 가지가 있음

- 이들은 모두「공통적 구조 · 행동을 나타내는 사항들을 집단으로 구분하는 분류(Classification)에 근거를 두고 있음



② 객체의 식별을 위해서 분류하는 과정에서 가장 핵심적인 문제는 「식별을 위하여 정확한 경계(Boundary)」를 설정하는 것임

* 이유 : 분류기준이 다를 경우 동일한 내용이라도 결과가 달라짐

- 분류 목적에 따라 그 결과가 달라지거나, 객체 식별을 위한 경계가 불명확하여 담당자의 견해 차이가 유발됨

③ 분류(classification) 문제를 해결하기 위한 최선의 방법은 각 추상화 개발방법에서

- 문제의 해결 대책을 제시

- 그 해결 대책을 체계적으로 이해한 후,이해된 해결 대책을 분석하여 코드화 하는 절차를 반복하다가 원하는 결과가 산출되면 종료하는 점진적으로 접근하는 것임

④ 「객체 혹은 클래스의 결정 방법」은 다음의 두 가지가 있음

* 전통적인 방법

- 공통적인 특성을 가진 객체들을 동일한 그룹(Group)으로 분류하는 방법으로서「객체가 특정한 특성을 소유했는지 여부에 따라서 결정하는 방식」임

- 특정한 시스템과 관련하여 상호작용이 이루어지는 외부개체(다른 시스템이나 디바이스, 사람, 기계)

- 다수의 하위 요소로 이루어진 구조 (전송수단, 교육수단, 운송수단)

- 시스템과 관련된 상호작용에서 사람의 수행역할 (사용자, 운영자)

- 사람이 소속한 조직 (교수부, 개발팀, 결재라인, 재무관리실)

- 문제에 대하여 정보영역 부분이 되는 사물 (팩스, 문서, 보고서, 편지)

* 객체지향 방법

- 문제영역을 구성하고 있는 객체와 클래스를 찾아서 문제를 모델화 하는 방법임

- 이 방법에서는 행위분석, 영역분석, 유즈케이스분석 등을 사용하여 객체를 결정함

- 객체지향분석에서는 객체의 결정은 행위분석, 영역 분석, 유즈케이스분석 등과 전통적 방법을 모두 사용함

⑤ 행위분석(Action Analysis)

- * 개념적 그룹화(Grouping)에서 파생된 행위분석 : 「동적인 행동을 가지고 있는 내용」을 「객체 혹은 클래스 후보」로 정하는 방식
- 기본적으로 「객체의 집합을 클래스로 규정」하고, 그 규정된 내용에 의거하여 객체들을 구분하는 방식임

⑥ 영역분석(Domain Analysis)

- * 영역분석 : 「문제를 가지고 있거나, 「개발대상인 업무의 전문가들이」 해당영역에서 중요한 것으로 판정되는 「관련(Association), 연산 (Operation), 객체(Object)」등을 모두 찾아내는 방법임
- 지정된 해당영역의 전문가에 의해서 일반적인 모델링(Modeling)의 정의
- 영역내의 모든 시스템을 조사하여 공통적인 방법으로 표현
- 영역 전문가 참여 하에 시스템 사이의 차이점과 유사점 구분
- 최초 정의모델을 기존 시스템에 적용 가능하게 재정의

⑦ 유즈케이스분석(Use-Case Analysis)

* 유즈케이스

분석 : 시스템에서 필요한 연산에 요구되는 시나리오(Scenario)를 기술하고, 그 시나리오를 스토리보드(Storyboard)로 작성하여 분석하는 방식임

- 시나리오를 분석함으로써 다음 사항을 명확히 파악 가능함
 - 연산에 관계되는 객체
 - 각 객체의 역할
 - 각 객체가 어떤 연산을 사용해서 다른 객체를 호출
 - 어떻게 다른 객체와 공동작업을 하는지
- 분석 중에 예외사항 · 보조 시스템 행위 등을 반영해야하기 때문에 최초의 시나리오는 확장 · 변형은 필연적임
- 그러므로 새로운 추상화가 만들어져 이미 정의된 추상화의 역할 변경 · 추가 · 첨삭을 통해서 분석이 이루어짐

2) 객체의 선정

① 「객체후보」를 결정하기 위해서는 대상업무에 대하여「문제점」이나「활동」을 기술한 요구명세서, 시스템정의서, 문제정의 등과 같은 각종 문서를 읽어 가면서 「명사(Noun)」혹은「명사절」을 골라내서 「객체 리스트(Object list)」를 작성하면 됨

② 객체후보를 선정할 때 객체의 집합이「클래스」이고, 객체는 클래스의 「인스턴스(Instance)」라는 사실을 염두에 두고, 다음과 같이 분석함

- 문제정의에 나타난「명사」에 줄을 그음
- 문제정의에 나타난 주요「동사(verb)」를 차례대로 별도의 종이에 적음
- 밑줄 그어진「명사」가 객체가 될 수 있는지 여부를 판정함 (객체의 속성에 해당하는「명사」는 제외되어야 함)
- 후보객체가 문제영역(problem space), 해답영역(solution space)중에 어디에 속하는지 결정함
 - 그 이유는 문제영역은 해답영역을 포함하는 상위개념으로서 시스템의 범위를 나타내기 때문임
 - 「해답영역」은 「문제영역의 부분집합으로」시스템을 구현한 내용이기 때문에 해답영역에 속하는 객체는 문제영역에도 나타남
 - 문제영역에 속하는 객체는 해답영역에 나타나지 않음
- 해답영역에 속하는 객체들에 대해서「이름」을 부여하며, 문제영역에 속하는 객체는 무시함

③ 원칙에 의거한「객체후보를 결정하는 과정」 요약

- 문제정의에 나타난「명사」들을 차례대로 종이 위에 기재하는「객체후보 리스트」를 작성함 이「명사」들이「객체」가 될 수 있는지 여부를 검토함
- 「명사」중에서 객체의 속성을 나타내는「명사」(「명사」중에서 값을 가지는 명사(주민등록번호, 총수령액))는 객체후보가 될 수 없기 때문에 지우며, 이 제외 된「명사」들은 별도의 종이에 적어둠
- 객체후보가 문제영역과 해답영역 중에서 어디에 속하는지 판정하며, 그 이유는 문제영역의 객체들은 문제기술(Problem Description)에 유용하지만 해답영역에는 필요치 않기 때문임
- 이 과정까지 객체후보 리스트에 남아 있는「명사」중에서
 - 시스템 영역 외에 존재하는 명사
 - 무의미한 명사
 - 중복되는 명사 등을 제외하고, 「필수객체(Essential Object)」만 이용하여 최종적인 「객체 리스트(Object List)」를 작성함
- 원칙에 의거하여 작성된 최종적인「객체 리스트」에 따라서「객체 모델(object model)」이 작성됨

④ 요든(yourdon)과 코드(coad) 등이 제시한 필수객체를 선택하는 기준

- 속성이 다른 객체에도 적용되는 다중속성을 가진 객체후보는 유용함
- 정의된 속성이 모든 객체에 적용되는 공통속성을 가진 객체후보는 유용함
- 시스템이 기능을 나타내기 위한 정보를 기억하는 보유정보 기능을 가진 객체후보는 유용함
- 정의된 연산이 출현하는 모든 객체에 적용되는 공통연산 기능을 가지는 객체후보는 유용함
- 연산에 반드시 요구되는 필수요구를 포함한 객체후보는 유용함
- 객체가 자신의 속성 값을 변경할 수 있는 몇 가지 연산이 가능한 필요한 서비스 (needs service) 기능을 가진 객체후보는 유용함

2. 객체의 구조결정

1) 객체는 독립적으로 정의됨

- 업무수행에서는 상호 유기적인 관계를 이루며 진행됨
- 객체 사이에 존재하는 관계를 감안하여 관련 있는 클래스들을 서로 연결하여「객체구조」를 결정해야 함

2) 객체구조를 결정하는 방법 두 가지

- 상속 기능을 이용하는 분류구조(Classification Structure).
- 조립(Aggregation) 기능을 이용하는 전체-부분구조(Whole-Part Structure)

3) 객체구조를 결정하는 방법 두 가지

① 분류구조

- 상위클래스의 속성은 하위클래스에 상속되기 때문에 공통적으로 사용되는 속성은 상위클래스에서 정의하고 하위클래스는 상위클래스에서 정의하지 않은 개별적 속성만을 추가적으로 정의해서, 새로운 클래스를 정의해 가는 원리를 이용하는 기법임

② 전체-부분구조

- 속성의 상속을 나타내지는 않지만 「개발대상 시스템의 경계를 설정하거나,」 「시스템의 목적을 명확히 할 수 있다」는 장점 때문에 이 구조를 이 용하여「객체구조」를 결정하기도 함
- 분석과정에서 선정된 클래스를 특수화(Specialization)에 의해서 「하위클래스를 생성하거나」 「이전의 정의에서 누락되었던 클래스를 새로 작성하여 연결함으로써」 「객체구조」를 결정하는 방법이「전체-부분구조」임
- 객체구조 결정과정에서 대규모, 복잡한 시스템인 경우 해당 시스템을 주제 (Subject)별로 분할하는 것이 효과적임

학습내용3 : 속성의 기술 / 연산의 파악

1. 속성의 기술

1) 객체의 속성(Attribute)

- ① 객체가 연산(Operation)을 하는데 필요한 자료(Data)를 의미 하며, 그러므로 「객체의 속성후보」는「객체를 설명하는 다양한 정보」로서 다음과 같은 것들임
- ② 논리적 특성 정보 : 성명, 번호, 우선순위, 단위
- ③ 물리적 특성 정보 : 중량, 길이, 성분
- ④ 객체에 저장된 정보 : 원시자료, 계측 값
- ⑤ 상태 정보 정지와 움직임, 켜짐과 꺼짐
- ⑥ 이들 중에 유용하지 않은 정보도 포함되어 있으므로 시스템에 무의미한 정보는 제외 시켜야 함

2) 객체를 설명하는 정보 중에 클래스에 필요한 속성으로 규정되는 정보의 유형

- ① 원시자료(source data)
- ② 단일항목 정보(복합정보보다 단일항목 정보)
- ③ 문제해결에 필수적인 정보

3) 객체의 속성을 정의하는데 유의사항

- ① 객체에 공통적으로 적용되는 속성은 상위클래스로 옮겨야 함
- ② 특정한 클래스에 관련되는 객체를 구분하기 위한 「식별항목에 대한 속성」은 설계과정에서 고려해야 함

4) 객체의 속성은 자료사전으로 나타냄

분야	Thing
성명 주소 전화번호 신용카드타입 신용카드번호	성명 = 성 + 이름 /* 고객의 성과 이름 */ 주소 = /* 고객의 주소 */ 전화번호 = 지역번호 + 국번호 + 고유번호 신용카드타입 = (직불카드 + 후불카드) 신용카드번호 = /* 고객의 신용카드번호 */

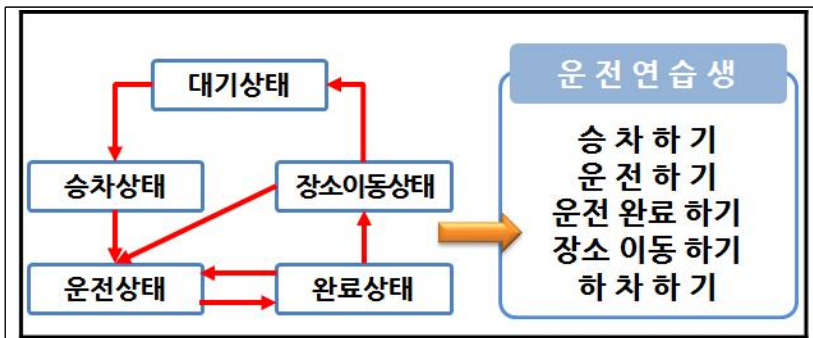
5) 자료사전으로 표현되는 객체의 속성기술

- ① 속성이름
- ② 속성의 용도
- ③ 표현의 단위
- ④ 자료타입
- ⑤ 기본 값
- ⑥ 값의 허용 범위 등에 대한 정보가 포함됨

2. 연산의 파악

- 1) 주어진 사건(Event)을 해결하기 위하여 수행하는 활동으로 인해서 자신의 속성은 다른 값으로 변함

- 이처럼 객체가 속성 값의 변화로 자신의 상태를 변화시키는 일련의 활동을 「객체의 연산(Operation)」이라고 함
- 2) 연산(Operation)을 파악하는 두 가지 방법
 - ① 계획단계에서 정의된 문제정의서, 수집한 각종 문서를 읽어 가면서 객체와 연결된 「동사(Verb)」들을 찾아서 그 객체의 연산으로 결정하는 방법
 - ② 요든과 코드가 제안한 정형화된 방법
- 3) 동사를 찾아 연산으로 결정하는 방법
 - 「문제정의서」에 나타난 모든「동사」를 찾음
 - 별도의 종이에 동사가 나타나는 순서대로 옮겨 적음
 - 옮겨 적은 동사가 연산이 될 수 있는지 여부를 판정함 (연산이 될 수 없는 동사는 제외시킴)
 - 연산이 된다고 판정된 동사들은 문제영역과 해답영역 중에서 어디에 해당하는지 구분하여「연산후보 리스트」를 작성함
 - 연산후보 리스트에서 「해답영역에 해당하면서 중복되지 않는 동사를 골라서」연산으로 선정함
 - 객체 중에서 연산에 의해서 자료가 읽혀지거나 변경되는 것에 연산을 할당함
- 4) 동사를 찾아서 연산으로 결정 과정에서의 유의사항
 - 「문제정의서에 나타난 동사」가 연산이 될 수 있는지 여부를 판단하는 기준은 「동사의 표현 의미가 시스템에 제공되어야 하는 것이면」 연산으로 규정함
 - 그러므로 「선택하다 · 반납하다」와 같이 시스템에서 구현해야 할 내용 등은 제외시켜야 함
 - 문제영역에 포함되는 연산은 실제 구현과 관계가 없기 때문에 제외시킴
 - 문제영역의 연산은 문제를 설명하는 데만 필요하기 때문임
 - 식별과 이해를 용이하게 하기 위해서 원래 연산이 뜻하는 내용을 나타내는 「연산 이름」을 부여해야 함
 - 정의된 연산들을 「클래스」와「객체」로 나누어 분류함
 - 「코드와 요든이 제안한 정형화된 방법」으로 연산을 정의하는 과정은 다음과 같음
 - 객체가 가질 수 있는 상태를 파악한 다음에 객체의 상태(State)를 바꾸게 하는 활동이 어떤 것인지를 규명한다면 연산을 정의 할 수 있다는 견해임
 - 주어진 객체상태를 객체상태도 (Object State Diagram)를 작성하고 그 객체상태도에서 연산을 유도한 내용임



- 객체상태도는 객체의 최초상태에서부터 그 이후에 파생되는 상태를 그림으로 표현함으로써 객체의 상태 변화 내용을 일목요연하게 정리하기 때문에 매우 유용한 도구임
- 객체상태도에서 객체는「사각형」으로 나타내고, 특정상태에서 다른 상태로 옮겨가는 것은「화살표」로 나타냄.
- 객체상태도에 나타난 모든「화살표」에 대해서는 반드시 연산 내용을 정의해야함
 - 그 이유는 객체의 상태는 연산에 의해서 변화되기 때문임

- 객체가 수행하는 연산에는 다음과 같은 두 가지가 있음
 - 기본연산 : create : /*객체생성 및 초기화 연산 */,
 - access : /* 객체 속성 값 확보 연산 */,
 - connect : /* 다른 객체와 연결 연산 */,
 - release : /* 다른 객체와 연결 연산 */
 - 복합연산 : compute : /* 계산 수행 연산 */,
 - monitor : /* 사건발생 감시연산 */,
 - manipulate : /* 자료 조작 연산 */

3. 메시지

1) 객체 사이의 연결

- 메시지(Message)에 의해서 이루어짐
- 그러므로 이를 통해서만 원하는 객체에게 활동을 요구할 수 있고, 응답을 받을 수 있음

2) 메시지의 구성 세 부분

- ① 객체명(Object Name)
- ② 연산명(Operation Name)
- ③ 인수(Argument)

【학습정리】

1. 객체 개요 / 문제정의를 알아본다.
2. 객체의 식별과 결정 /구조결정을 이해한다.
3. 속성의 기술 / 연산의 파악을 파악한다.