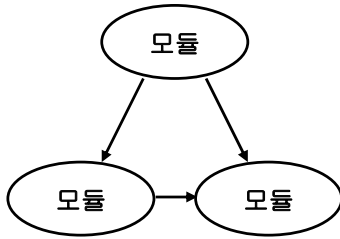


1. 구조적 개발 VS 객체지향 개발

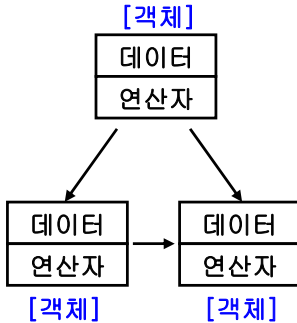


[장점]

구조 단순 -> 이해 O, 수정 O, 정확 O (C언어)

[단점]

소프트웨어 재사용, 유지보수 어려움 -> 소프트웨어 위기 해결 안됨



[장점]

현실 세계를 프로그램에 반영

소프트웨어 재사용, 유지보수 향상 -> 소프트웨어 위기 해결 방안

[관련 용어]

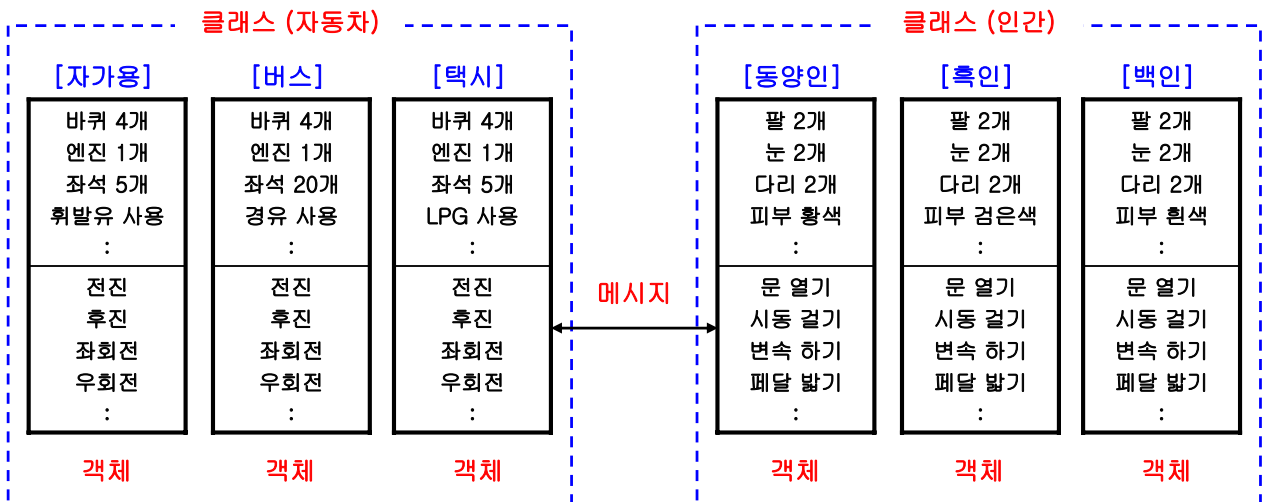
기본 : 객체, 클래스, 메시지

원칙 : 캡슐화, 정보 은폐, 추상화, 상속성, 다형성

데이터 = 상태, 속성(Attribute), 변수, 자료구조

연산자 = 행위, 메소드(Method), 동작(Operation)

2. 객체, 클래스, 메시지 - 개념 이해하기



1) 객체 (Object)

- 현실 세계의 개체며 객체들 간의 상호작용은 메시지를 통해 이루어짐

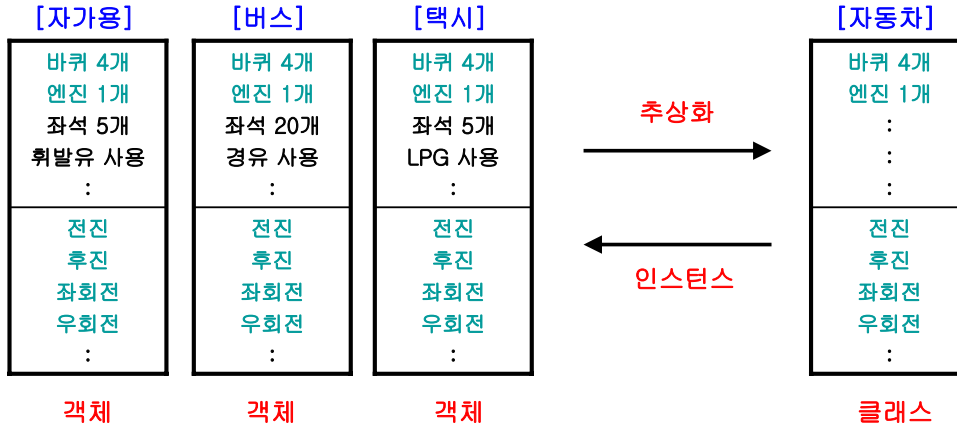
① 데이터 : 객체가 가지고 있는 **상태** (속성, Attribute, 변수, 자료구조)

② 연산자 : 객체의 데이터를 처리하는 **행위** (메소드, Method, 동작, Operation, 함수, 프로시저)

[SE 4강]-객체지향 개발 방법론

2) 클래스 (Class)

- 하나 이상의 유사한 객체들을 묶어 공통된 특성을 표현한 데이터 추상화(모델링)를 의미
- 공통된 속성과 연산을 갖는 객체의 집합 (객체의 일반적인 타입)
- 인스턴스 (Instance) : 클래스에 속한 각각의 객체 (객체는 클래스의 인스턴스)



[SE 4강]-객체지향 개발 방법론

3) 메시지 (Message)

- 객체들 간에 상호작용을 하는데 사용되는 수단
- 객체에서 객체로 메시지가 전달되면 메소드(행위)를 시작함

4) 메소드(method)

- 객체지향 시스템에서 전통적 시스템의 함수(function) 또는 프로시저(procedure)에 해당하는 연산기능
- 객체지향 개념에서 객체가 메시지를 받아 실행해야 할 객체의 구체적인 연산

3. 캡슐화 (Encapsulation) ★★★★★

- 자료 부분과 연산(또는 함수) 부분 등 정보처리에 필요한 기능을 한 테두리로 묶는 것
- 왜? 정보 은폐 -> 외부에서 변경 X -> 프로그램 변경에 대한 오류의 파급효과가 적다 (결합도 낮아짐)
- > 재사용 용이, 객체간의 인터페이스 단순화, 응집도 향상

4. 정보은폐 (Information Hiding) ★★☆☆☆

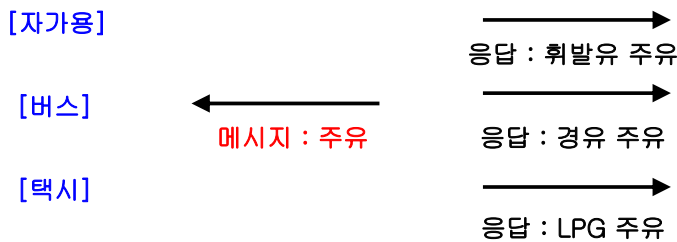
- 객체는 다른 객체로부터 자신의 자료를 숨기고 자신의 연산만을 통하여 접근을 허용하는 것
- 왜? 고려되지 않은 영향들을 최소화하기 위해

5. 상속 (Inheritance) ★★☆☆☆☆

- 상위 클래스의 메소드와 속성을 하위 클래스가 물려받는 것
- 다중 상속은 한 클래스가 여러 상위 클래스로부터 상속 받는 것 (상위 클래스 : 비행기, 자동차) -> 재사용

6. 다형성 (Polymorphism) ★☆☆☆☆

- 한 메시지가 객체에 따라 다른 방법으로 응답할 수 있는 것
- 많은 상이한 클래스들이 동일한 메소드명을 이용하는 능력



1. 객체지향 프로그램 특징 ★☆☆☆☆

- 재사용율이 높다 -> 자연적인 모델링이 가능
- 유지보수성이 향상된다.
- 사용자 중심, 대화식 프로그램 등 대형 프로젝트 개발에 적합
- 실행 속도가 빨라진다. (X)

2. 객체지향 개발 단계 ★☆☆☆☆

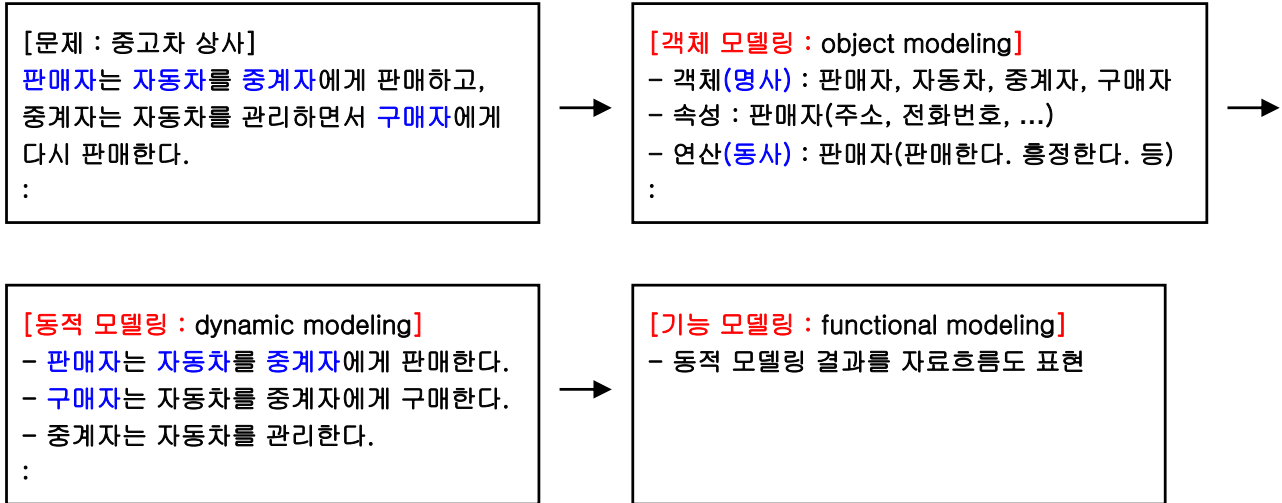
- : 계획 -> 분석 -> 설계 -> 구현 -> 테스트 및 검증
- 분석과 설계, 구현 작업이 폭포수 모형과 같이 순서가 뚜렷하게 구별되지 않음

3. 객체지향 분석 (OOA : Object Oriented Analysis) ★☆☆☆☆

- 모델링 구성요소인 클래스, 객체, 속성, 연산들을 표현해서 문제를 모형화 (ERD)
- 모형화 표기법의 전후관계에서 객체의 분류, 속성들의 상속, 그리고 메시지의 통신 등을 결합한 것이다.
- 주요 목적 : 객체는 클래스로부터 인스턴스화 되고, 이 클래스를 식별
- 기법 : Rumbaugh(럼바우)

[SE 4강]-객체지향 개발 방법론

4. Rumbaugh 분석 기법



* 절차 : 객체 모형 -> 동적 모형 -> 기능 모형

- 객체 모델링 : 시스템에서 요구되는 객체를 찾아내어 객체들의 특성을 규명
- 동적 모델링 : 상태도를 이용하여 객체들의 제어 흐름, 상호 반응 연산 순서를 나타내주는 과정
- 기능 모델링 : 자료흐름도(DFD)를 이용하여 각 객체에서 수행되는 동작들을 기술

[SE 4강]-객체지향 개발 방법론

5. 객체지향 설계 ★☆☆☆☆

- 객체지향 설계에 있어 가장 중요한 문제는 시스템을 구성하는 객체와 속성, 연산을 인식하는 것이다.
- 객체지향 설계를 문서화할 때 객체와 그들의 부객체(sub-object)의 계층적 구조를 보여주는 계층차트를 그리면 유용하다.
- 객체는 순차적으로(Sequentially) 또는 동시적으로(Concurrently) 구현될 수 있다.
- 객체의 속성과 자료구조를 표현한다.
- 서브 클래스와 메시지 특성을 세분화하여 세부사항을 정제화한다.

6. 객체지향 구현

- 객체지향 프로그래밍 언어에는 Smalltalk, C++ 등이 있다.
- 절차 중심 프로그래밍 기법이다. (X)

7. 테스트 ★☆☆☆☆

- 테스트 순서 : 단위(class) 테스트 -> 통합 테스트 -> 검증과 시스템 테스트

[SE 4강]-객체지향 개발 방법론

1. 객체지향 기법에서 오브젝트의 상태는 무엇을 파악함으로써 알 수 있는가?

- 가. 오퍼레이션(Operation)
- 나. 어트리뷰트(Attribute)
- 다. 트리거(Trigger)
- 라. 메소드(Method)

2. 객체지향 시스템에서 전통적 시스템의 함수(Function) 또는 프로시저(Procedure)에 해당하는 연산 기능을 무엇이라고 하는가?

- 가. 메소드(Method)
- 나. 메시지(Message)
- 다. 모듈(Module)
- 라. 패키지(Package)

3. 객체지향 개념에 관한 설명 중 잘못된 것은?

- 가. 객체들 간의 상호작용은 메시지를 통해 이루어진다.
- 나. 클래스는 인스턴스(Instance)들이 갖는 변수들과 인스턴스들이 사용할 메소드(Method)를 갖는다.
- 다. 다중 상속(Multiple Inheritance)은 두 개 이상의 클래스가 한 클래스로부터 상속 받는 것을 말한다.
- 라. 객체가 갖는 데이터를 처리하는 알고리즘을 연산(Operation) 또는 메소드(Method)라 한다.

[정답] 1.나 2.가 3.다 4.나 5.나 6.나

4. 모든 객체들은 더 큰()의 멤버이고, 그()에 대하여 이미 정의된 개별 자료 구조와 연산이 상속된다. 그 때문에 개별 객체는()의 인스턴스가 된다. 다음 중()안에 공통으로 들어갈 내용은?

- 가. 메시지
- 나. 클래스
- 다. 상속성
- 라. 정보

5. 객체지향 개념 중 하나 이상의 유사한 객체들을 묶어 공통된 특성을 표현한 데이터 추상화를 의미하는 것은?

- 가. 메소드(Method)
- 나. 클래스(Class)
- 다. 상속성(Inheritance)
- 라. 추상화(Abstraction)

6. 객체지향 프로그램의 장점으로 거리가 먼 것은?

- 가. 자연적인 모델링이 가능하다.
- 나. 실행 속도가 빨라진다.
- 다. 소프트웨어의 재사용 율이 높아진다.
- 라. 소프트웨어의 유지보수성이 향상된다.



[SE 4강]-객체지향 개발 방법론

7. 기존의 소프트웨어 공학 기법들과 차별화될 수 있는 객체지향 개념이 아닌 것은 어느 것인가?

- 가. 캡슐화(Encapsulation)
- 나. 상속성(Inheritance)
- 다. 다형성(Polymorphism)
- 라. 모듈화(Modularity)

8. 객체지향의 기본 원리인 정보은폐와 가장 밀접한 관계가 있는 것은?

- 가. 캡슐화(Encapsulation)
- 나. 클래스(Class)
- 다. 메시지(Message)
- 라. 상속성(Inheritance)

9. 객체는 다른 객체로부터 자신의 자료를 숨기고 자신의 연산만을 통하여 접근을 허용하는 것을 무엇이라 하는가?

- 가. Abstraction
- 나. Information Hiding
- 다. Modularity
- 라. Typing

10. 객체지향 기법에서 상속(Inheritance)의 결과로서 얻을 수 있는 가장 주요한 이점은?

- 가. 모듈 라이브러리의 재이용
- 나. 객체지향 DB를 사용할 수 있는 능력
- 다. 클래스와 오브젝트들을 재사용할 수 있는 능력
- 라. 프로젝트들을 보다 효과적으로 관리할 수 있는 능력

11. 객체지향 기술에서 다형성(Polymorphism)의 의미로 가장 적절한 것은?

- 가. 다중 메시지를 수행하기 위하여 이용되는 기술
- 나. 동일한 일을 수행하기 위하여 상이한 메소드 이름을 이용하는 능력
- 다. 상이한 일을 수행하기 위하여 동일한 메시지 형태를 이용하는 능력
- 라. 많은 상이한 클래스들이 동일한 메소드명을 이용하는 능력

[정답] 7.라 8.가 9.나 10.다 11.라



12. 객체지향 소프트웨어 개발 모형의 개발 단계로 옳은 것은?

- | | | |
|-----|-----------|-----|
| ㉠설계 | ㉡구현 | ㉢계획 |
| ㉣분석 | ㉤테스트 및 검증 | |

- 가. ㉠→㉡→㉢→㉣→㉤
 나. ㉠→㉢→㉣→㉡→㉤
 다. ㉠→㉣→㉢→㉡→㉤
 라. ㉠→㉢→㉡→㉣→㉤

13. 객체지향 분석 기법의 하나로 객체 모형, 동적 모형, 기능 모형의 3개 모형을 생성하는 방법은?

- 가. Wirfs-block Method
 나. Rumbaugh Method
 다. Booch Method
 라. Jacobson Method

14. 객체지향 분석에 대한 설명으로 옳지 않은 것은?

- 가. 분석가에게 주요한 모델링 구성 요소인 클래스, 객체, 속성, 연산들을 표현해서 문제를 모형화 시킬 수 있게 해준다.
 나. 객체지향 관점은 모형화 표기법의 전후 관계에서 객체의 분류, 속성들의 상속, 그리고 메시지의 통신 등을 결합한 것이다.
 다. 객체는 클래스로부터 인스턴스화 되고, 이 클래스를 식별하는 것이 객체지향 분석의 주요한 목적이다.
 라. E-R 다이어그램은 객체지향 분석의 표기법으로는 적합하지 않다.

15. 럼바우(Rumbaugh)의 객체 지향 분석절차를 바르게 나열한 것은?

- 가. 객체 모형 → 동적 모형 → 기능 모형
 나. 객체 모형 → 기능 모형 → 동적 모형
 다. 기능 모형 → 동적 모형 → 객체 모형
 라. 기능 모형 → 객체 모형 → 동적 모형

[정답] 12.라 13.나 14.라 15.가



16. 객체지향 설계 방법론에 대한 설명 중 옳지 않은 것은?

- 가. 구체적인 절차를 표현한다.
 나. 형식적인 전략으로 기술한다.
 다. 객체의 속성과 자료 구조를 표현한다.
 라. 서브 클래스의 메시지 특성을 세분화하여 세부 사항을 정제화 한다.

18. 럼바우의 객체 지향 분석에서 분석활동의 모델링과 관계없는 것은?

- 가. 객체(object) 모델링 나. 절차(procedure) 모델링
 다. 동적(dynamic) 모델링 라. 기능(functional) 모델링

17. 객체지향 프로그래밍(OOP, Object-Oriented Programming)개발 기법에 대한 설명으로 옳지 않은 것은?

- 가. 절차 중심 프로그래밍 기법이다.
 나. 객체지향 프로그래밍 언어에는 Smalltalk, C++등이 있다.
 다. 객체 모델 주요 요소는 추상화, 캡슐화, 모듈화, 계층 등이다.
 라. 설계 시 자료와 자료에 가해지는 프로세스를 묶어 정의하고 관계를 규명한다.

[정답] 16.나 17.가 18.나

