

PC용 객체지향 구조해석 프로그램의 개발 Development of Object-Oriented Structural Analysis Program for PC

신 영식* 서 진국** 박 영식** 최 희욱**
Shin, Young-Shik Suh, Jin-Kook Park, Young-Shik Choi, Hee-Wook

ABSTRACT

A computer program for matrix structural analysis by object-oriented programming technique using C++ language has been developed. Object, class, method and inheritance which are used in object-oriented programming are illustrated using a graphical representation. The matrix operations and the structural analysis by matrix displacement method were satisfactorily performed by the proposed program. Numerical tests demonstrate that this proposed program can be widely used for the matrix structural analysis for personal computers.

1. 서 론

대표적인 수치해석 방법인 유한요소법은 컴퓨터의 급속한 발전과 더불어 여러 공학분야에서 광범위하게 사용되어 이에 근거한 컴퓨터 프로그램의 개발로 보다 정확한 구조해석이 가능하게 되었다. 지금까지 많은 구조해석 프로그램들이 개발되어 널리 사용되고 있는데 대부분이 범용프로그램(General purpose program)이라 불리는 대형 컴퓨터용이었다. Bathe에 의해 개발된 SAP[1]과 같은 범용 구조해석 프로그램은 지금까지 많이 사용되어 왔으나 비교적 다양하고 정확한 구조해석기능에 비해 실행시간이 길며 자료의 입출력이 불편할 뿐 아니라 소형구조해석 및 부분적인 실행의 경우에도 프로그램 전체를 실행시켜야 하며 부분적인 프로그램의 수정이 거의 불가능하였다. 그러나 최근 10여년간 개인용 소형컴퓨터(Personal computer)의 폭넓은 보급과 기억용량및 처리속도의 향상은 이러한 범용프로그램의 단점을 개선한 PC용 프로그램의 개발을 촉진시키고 있다.

지금까지 개발된 PC용 구조해석 프로그램들은 범용프로그램이 가지고 있는 여러가지 구조해석의 기능은 가지고 있지만 전공지식이 충분치 않은 경우에 프로그램 이용이 곤란하며 사용자가 Macro command등을 잘못 사용했을 경우와 Error Message가 나타났을 경우에는 더 이상의 해석이 불가능하게 되어 있다. 일반적으로 PC용 구조해석 프로그램은 다양한 메뉴선택과 해석기능을 가지고 있어야 하나 FORTRAN 및 BASIC언어로 작성된 기존의 프로그램들은 단

* 영남대학교 토목공학과 교수

** 영남대학교 토목공학과 대학원

지 정확하고 신속한 처리에만 주력하고 있을 뿐 다양한 메뉴의 선택, 부분적인 프로그램의 추가 및 수정 등이 용이하지 않다. 최근 이러한 단점을 보완하기 위해 프로그램이 간편하고 확장 및 수정이 매우 간단하며 프로그램의 신뢰성 및 재사용성이 뛰어난 객체지향 프로그래밍(Object-oriented programming) 기법[2]이 제안되었다.

객체지향 프로그래밍 기법의 주된 목적은 객체(Object)라 불리는 독립적이고 항상 사용 가능한 소프트웨어를 구성하는 것인데, 이는 대규모의 응용 프로그램을 개발하고자 할 때 각 소프트웨어의 구성요소를 병렬응용범위에서 설계할 수 있고, 비슷한 응용에 대해 기존의 Code(또는 객체)를 재사용할 수 있으며 Prototype을 효과적으로 형성할 수 있음을 의미한다. 이것은 대형 응용프로그램의 유지, 보수와 개발을 신속히 할 수 있게 한다. 자료흐름 방식이나 수학적 논리를 기초로 하는 기존의 절차적 언어(Procedural languages)의 프로그래밍 방식과는 달리 객체지향 프로그래밍은 응용하고자 하는 대상을 직접 모델링할 수 있으며 특히 객체지향적 개념에 의해 특별히 개발된 언어인 Smalltalk[3]과 같은 언어와는 달리 본 연구에서 사용된 C++언어[4]는 기존의 C언어의 효율성과 객체지향적 개념을 함께 포함하고 있는 Hybrid 언어이기 때문에 C로 작성된 기존의 프로그램을 그대로 이용할 수 있다는 장점이 있다.

이러한 객체지향 프로그래밍 기법에 대한 연구는 1986년경 전산공학 분야에서 제안되어 구조해석분야에서도 1990년 Forde 등[5]이 객체지향 유한요소해석에 관한 연구를 발표한 이래 이에 관한 많은 연구가 활발히 진행되고 있다. Fenves[6], Zimmermann 등[7]의 연구에서 Smalltalk과 같은 객체지향적 언어가 유한요소해석을 다룰 수 있음을 보여 주었고, C++언어를 사용한 Mackie[8], Scholz[9] 등의 연구가 있으나 아직은 유한요소해석 프로그램에 대한 피상적인 개념 설명에 불과하다고 할 수 있다.

따라서 본 연구에서는 이러한 객체지향 프로그래밍 기법으로 매트릭스의 연산과 평면뼈대 구조해석이 가능한 PC용 프로그램을 개발하여 앞으로의 객체지향 유한요소 프로그램 개발에 활용하고자 한다.

2. 객체지향 프로그래밍의 개념

객체지향 프로그래밍은 함수들과 함수의 처리대상이 되는 자료(Data)들을 주어진 문제의 범위에 있는 객체로 취급하여 이 객체의 성격을 파악하고 그 객체를 구현하는 과정이 프로그램의 작성과정이 된다. 자료구조와 제어구조로 된 기존의 프로그래밍 기법과는 달리 객체지향 프로그래밍은 객체(Object), 클래스(Class), 처리방식(Methods), 메시지(Message), 상속성(Inheritance), 다형성(Polymorphism) 등의 여러가지 개념으로 구성되며, 이러한 객체와 클래스, 처리방식 사이의 관계에 대한 개념적 모델을 나타낸 것이 그림1, 그림2이다.

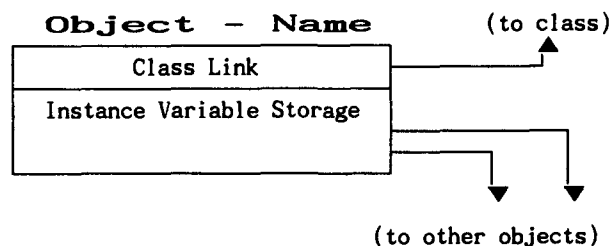


그림 1. Graphical representation of an object

은 이름의 처리방식을 가질 수 있다. 이와 같이 각기 다른 객체들이 동일한 메시지에 대해 각각 다른 반응을 취하는 것을 다형성이라 하며, 이것은 시스템내에 이미 존재하는 메시지에 대해 응답할 수 있는 새로운 객체를 시스템에 삽입하는 것이 용이하다는 것을 의미한다.

3. 객체지향 구조해석 프로그램의 구성

본 객체지향 구조해석 프로그램은 그림 3과 같이 주 클래스(Main class), 매트릭스 연산 및 벡터 클래스(Matrix & Vector class), 구조해석 클래스(STRU class)로 구성되어 있으며, 이 중 주 클래스와 구조해석 클래스는 추상적 클래스이다.

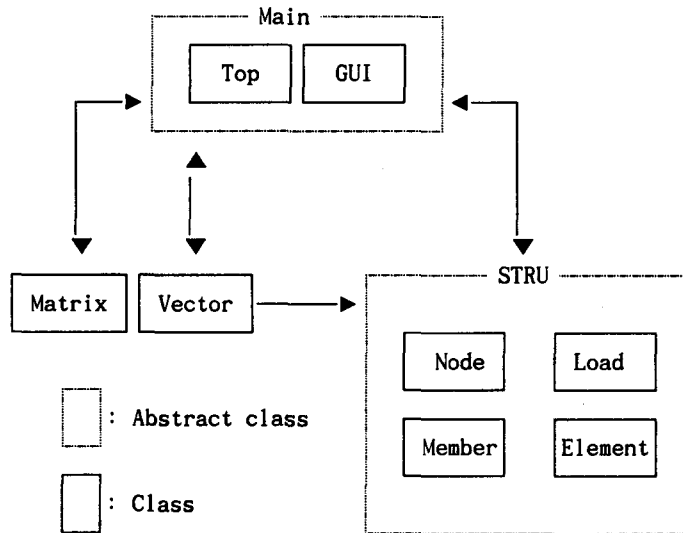


그림 3. Structure of object-oriented structural analysis program

3.1 주 클래스

주 클래스는 아래의 매트릭스 연산 클래스, 구조해석 클래스들을 총괄하는 입출력 제어, 명령의 전달 등을 수행하는 최상위 클래스(Top class)와 자료의 입출력과 저장을 담당하는 GUI(Graphic user interface)클래스를 포함하는 추상적 클래스이다. 그림 4는 최상위 클래스의 생성자가 수행됨과 동시에 프로그램의 변화를 감지하여 그에 해당하는 명령Icon의 Instance를 수행시키는 Event클래스의 Member함수가 Event Loop안에서 수행되는 주 클래스의 내부 수행체계를 나타낸다. 그림 5는 GUI클래스중 Window의 구성과 다른 클래스와의 관계를 도식화한 것이다.

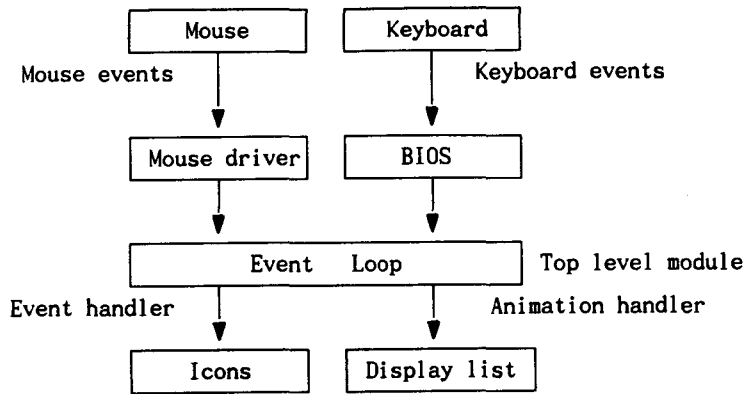


그림 4. Connection between user and Icons in main class

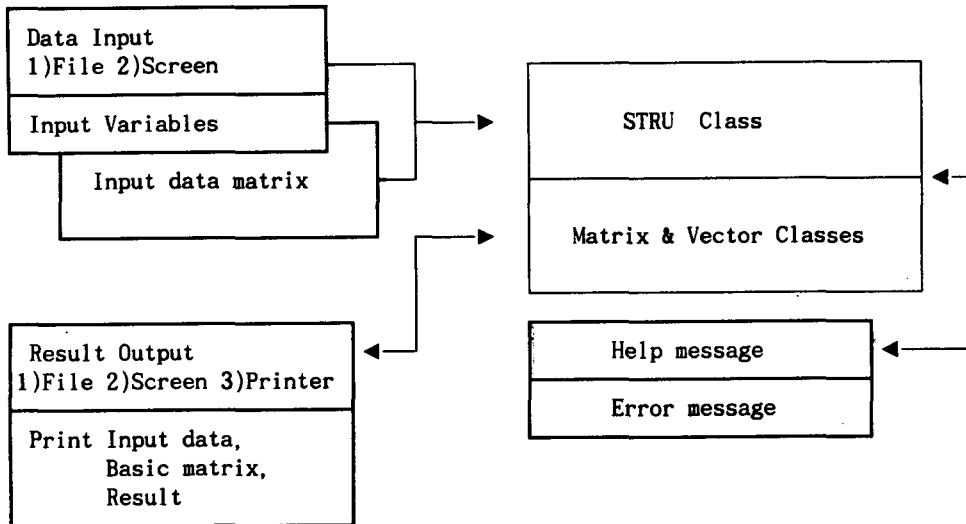


그림 5. Structure of a user interface class

3.2 매트릭스 연산 및 벡터 클래스

이 클래스는 구조해석시 최소한 한번이상 이용되는 클래스로 벡터, 행렬의 연산과 고유값 문제의 해석을 수행한다. 행렬의 연산과 벡터의 연산은 동일한 형태의 연산자로 처리되며 하나의 함수가 두개이상의 클래스에 대하여 연산을 수행하도록 "friend"명령어를 사용한다. 하나의 클래스내에 동일한 함수명으로 서로 다른 인수를 가지는 함수를 중복하여 선언하는 C++언어의 고유한 초과연산자의 특징과 다형성을 잘 나타낼 수 있다. 또한 프로그램상에서 매트릭스와 벡터를 하나의 변수형태로하여 스칼라변수와와의 연산이 가능하며 각 클래스의 구성은 그림 6과 같다. 그림의 왼쪽은 함수, 오른쪽은 변수를 나타낸다.

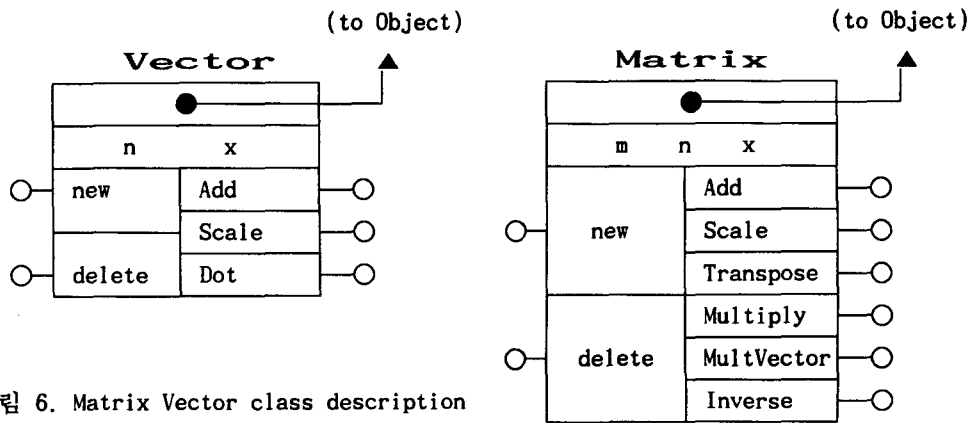


그림 6. Matrix Vector class description

3.3 구조해석 클래스

구조해석 클래스는 그림 7, 그림 8과 같은 클래스들을 포함하는 추상적 클래스이다. 이 클래스에서는 해석하고자 하는 구조물의 기하학적, 재료학적 자료 및 경계조건이 Node클래스 및, Member클래스에, 하중에 대한 자료가 Load클래스에 각각 입력되어 Element클래스에 의해 요소 강성도 매트릭스를 구성한 후, 전체 구조물에 대한 강성도 매트릭스를 조합하여 변위와 응력을 구하는 매트릭스 변위법으로 평면 뼈대 구조물을 해석한다.

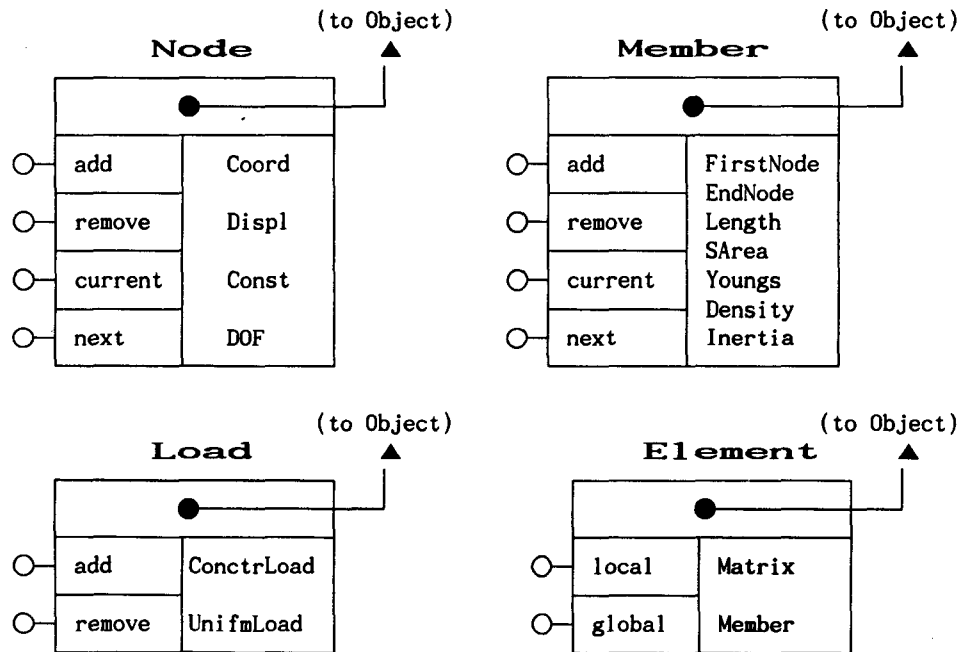


그림 7. STRU class description

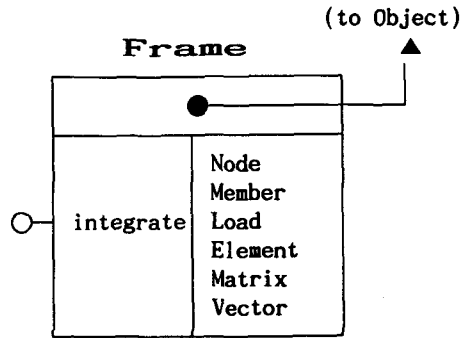


그림 8. Description of Frame class inherited from STRU class

그림 8은 실제 구조물을 해석하기 위한 클래스로 그림 7의 Node, Member, Load, Element 클래스들과 그림 6의 Matrix, Vector클래스를 내부변수로 사용하며, 해석하고자 하는 구조형태에 따라 Frame클래스와 같은 이미 생성된 기본 클래스들을 포함하는 구조해석용 클래스들을 필요에 따라 만들 수 있다.

앞에서 언급한 클래스외에 각 Node 및 Member의 객체를 연결하여 해석하고자 하는 구조형태를 구성해 주는 NodeList클래스와 MemberList클래스가 있다.

4. 고찰 및 결론

본 연구에서는 C++언어를 이용한 객체지향프로그래밍 기법으로 매트릭스의 연산과 평면 뼈대 구조물의 구조해석이 가능한 PC용 구조해석 프로그램을 개발 하였다. 이 객체지향 프로그램은 자료뿐 아니라 Code도 메모리(Memory)에 동적으로 생성되고 또한 동적으로 소멸되는 Dynamic Memory Allocation특성을 가지므로 작은 기억용량을 사용하더라도 프로그램이 정적으로 고정되어 있는 절차적언어 프로그램보다 처리속도가 빠르며 상속성및 다형성으로 인하여 프로그램 Size도 동일한 내용의 절차적 프로그램보다 축소된다. 본 연구에서 개발된 프로그램의 효율성을 예제해석을 통하여 검증한 결과 만족할만한 처리속도및 결과의 정확도를 고찰 할 수 있었다.

따라서 객체지향 프로그래밍 기법의 특징인 프로그램의 확장성과 재사용성및 다양한 GUI의 구현 가능성 등 여러가지 장점때문에, 앞으로 객체지향적 프로그래밍 기법을 이용한 다양한 구조해석 프로그램의 개발이 본격화 될 것으로 판단된다.

참고문헌

1. K.J.Bathe, E.L.Wilson and F.E.Peterson, " SAPV-A structural analysis program for static and dynamic response of linear systems ",Report No. EERC 73-11, Earthquake Engineering Research Center, Univ.of California, Berkeley (1973)
2. R.S.Wiener, and L.J.Pinson, *An Introduction to Object-Oriented Programming and C++*, Addison-Wesley, MA (1989)

3. A. Goldberg, and D. Robson, *Smalltalk-80, The Implementation*, Addison-Wesley, MA (1983)
4. B. Stroustrup, *The C++ Programming Language*, Addison-Wesley, MA (1986).
5. B. W. R. Forde, O. F. Ricardo and F. S. Siegfried, "Object oriented finite element analysis", *Computers & Structures*, Vol. 34, No. 3, pp. 355-374 (1990)
6. G. L. Fenves, "Object-oriented programming for engineering software development", *Engng. Comput.*, Vol. 6, pp. 1-15 (1990)
7. T. Zimmermann, Y. Dubois-Pelerin and P. Bomme, "Object-oriented finite element programming: I. Governing Principles", *Comput. Meth. Appl. Mech. Engng.*, Vol. 98, pp. 291-303 (1992)
8. R. I. Mackie, "Object oriented programming of the finite element method", *Int. J. Numer. Meth. Engng.*, Vol. 35, pp. 425-436 (1992)
9. S. P. Scholz, "Elements of an object-oriented FEM++ program in C++", *Computers & Structures*, Vol. 43, No. 3, pp. 517-529 (1992)
10. G. J. Miller, "An object-oriented approach to structural analysis and design", *Computers & Structures*, Vol. 40, No. 1, pp. 75-82 (1991)
11. B. W. Kernighan, and D. M. Ritchie, *The C Programming Language*, 2nd ed. Prentice-Hall (1988).
12. S. M. Holzer, *Computer Analysis of Structures*, Elsevier, New York (1985).