

---

# 구조적 프로그래밍을 위한 언어 학습 시스템

박경우<sup>\*</sup> · 류남훈<sup>\*\*</sup> · 김응곤<sup>\*\*\*</sup>

Language Education System with Structured Programming

Kyoung-wook Park<sup>\*</sup> · Nam-hoon Ryu<sup>\*\*</sup> · Eung-kon Kim<sup>\*\*\*</sup>

## 요약

컴퓨터 프로그램은 기계, 우주, 항공, 의학을 비롯한 사회 전 분야에서 필요로 한다. 하지만 프로그래밍 교육과정은 수많은 교재와 동영상 강의에도 불구하고 어려워하고 있다. 프로그래밍 언어는 매우 다양하지만 대부분의 언어들은 동일한 구조를 사용하며, 표현 방식에 차이가 있을 뿐이다. 그래서 하나의 프로그래밍 언어를 습득하면 또 다른 언어를 배우는데 많은 시간과 노력을 기울이지 않아도 된다. 대부분의 프로그래밍 언어들은 일반적으로 순서, 선택, 반복 구조를 사용한다. 프로그래밍을 배우는 사람에게 중요한 것은 프로그램의 문법이 아니라 프로그램의 구조나 알고리즘이다. 본 논문에서는 순서도를 활용해 구조화된 프로그래밍을 학습할 수 있는 언어 학습 시스템을 설계하고 구현하였다.

## ABSTRACT

Computer programs are required from all areas in society including machine, space, aviation, and medicine. However, the programming curriculum is getting hard despite a lot of teaching materials and video lessons. Programming languages are very diverse, but most of them use the same structure, and they only have different expression methods. Therefore, if one learns one programming language, then it doesn't need to spend a lot of time and efforts to learn another programming language. Most programming languages use the structure of sequence, selection, and repetition in general. The important thing for programming learners is the structure or algorithm of programming not the grammar of program. This study designed and implemented the language learning system to learn structured programming by using a flowchart.

## 키워드

Flowchart, Algorithm, Structured Programming, Programming Language

## I. 서론

컴퓨터 프로그램은 기계, 우주, 항공, 의학을 비롯한 사회 전 분야에서 활용되고 있다. 하지만 프로그래밍 교육과정은 컴퓨터 관련학과를 비롯한 이공계 학과 대부분의 학생들이 어려워하고 있다. 프로그래밍

언어는 매우 다양하지만 대부분의 언어들은 동일한 구조를 사용하며, 표현 방식에 차이가 있을 뿐이다.

그래서 하나의 프로그래밍 언어를 습득하면 큰 어려움 없이 또 다른 언어를 배울 수 있다. 대부분의 프로그래밍 언어들은 순서 구조, 선택 구조, 반복 구조로 구성되어 있다. 일반적으로 프로그램

\* 순천대학교 컴퓨터공학과(xelfiria@nate.com)

\*\* 순천대학교 대학원 컴퓨터과학과(nhryu@sunchon.ac.kr)

\*\*\* 교신저자 : 순천대학교 컴퓨터공학과 교수(kek@sunchon.ac.kr)

접수일자 : 2010. 09. 10

심사(수정)일자 : 2010. 09. 28

제재 확정일자 : 2010. 10. 15

의 문법은 쉽게 습득하지만 작성하고자 하는 문제를 어떻게 구조화 시키는지, 어떤 흐름으로 작성해야 하는지 알지 못해서 많이 어려워한다. 그래서 알고리즘의 기본 개념 및 구조화된 프로그래밍을 학습할 수 있는 보조 프로그램의 필요성이 절실히 대두되고 있다. 본 논문에서는 작성하고자 하는 프로그램을 어떻게 구조화시키며, 알고리즘을 어떻게 작성해야 하는지 학습할 수 있도록 순서도를 활용한 언어 학습 시스템을 설계하고 구현하였다.

## II. 알고리즘 교육 시스템

### 2.1 알고리즘 교육 시스템 구성도

본 시스템은 프로그래밍 언어와 알고리즘의 개념 정립에 활용할 수 있는 순서도를 활용한 프로그래밍 교육 시스템이다. 알고리즘의 진행과정을 시각화하여 순서도의 실행과정 확인 및 스택, 변수의 데이터 흐름을 실시간으로 확인할 수 있도록 한다[1-2]. 프로그램을 시각화하고자 할 때에 사용하는 방법으로는 선정의(Predefinition) 방법[3], 함수 호출(Annotating) 방법[4], 선언(Declaration) 방법[5], 조정(Manipulation) 방법[6] 등이 있는데, 이중 함수 호출 방법이 가장 많이 사용된다. 본 시스템은 프로그램 원리에 대한 이해와 학습의 용이성 향상을 위해 순서도 기초 선택 모듈, 순서도 배치 모듈, 진행과정 표현 모듈, 명령어 해석 및 소스코드 실행 과정 표현 모듈로 구성된다. 그림 1은 알고리즘 교육 시스템의 구성도를 나타낸다.

#### 2.1.1 순서도 기호 선택 모듈

GUI 환경에 적합한 직관적 사용자 인터페이스로 제작하여 프로그램 원리 학습 시 알고리즘 교육 시스템 사용에 있어 어려움 없이 시스템에 대한 접근이 가능하므로 알고리즘 교육의 효율성을 보장한다. 순서도 기호 선택 시 도움말 및 마우스 이동에 따른 순서도 기호 그림 표현으로 순서도 배치 영역에 손쉽게 배치할 수 있도록 하여 알고리즘 작성에 대한 기본적인 학습이 가능하도록 한다.

#### 2.1.2 순서도 배치 모듈

학습자가 선택한 순서도 기호를 순서도 배치 영역

에 나타내주는 모듈이다. 명령입력 모듈, 변수선언 모듈, 조건입력 모듈, 분기방향 입력 모듈로 구성된다. 순서도 기호에 따른 알고리즘을 작성하기 위한 입력 대화상자, 화면에 표현되는 배치 기호 및 순서도의 흐름을 나타내기 위한 연결선이 표현되며, 알고리즘에 대한 전체적인 윤곽을 설정한다. 입력 대화상자의 경우 각 순서도의 처리 기호에 필요한 변수 정보 및 조건을 학습자가 입력할 수 있도록 하고, 이에 해당하는 설명을 포함하여 학습 시 발생할 수 있는 시행착오를 최소화 할 수 있게 한다.



그림 1. 알고리즘 교육 시스템의 구성도  
Fig. 1 Diagram of algorithm education system

#### 2.1.3 진행과정 표현 모듈

순서도의 실행과정을 통해 알고리즘의 흐름을 시각화한다. 학습자가 가장 어려워하는 알고리즘의 핵심 원리인 변수 간의 관계 및 변수 값 변화 과정을 단계별로 확인할 수 있도록 하여 원리 파악을 통한 알고리즘 학습이 가능하도록 한다. 순서도 실행 과정, 메모리뷰, 결과 창 표현 모듈로 구성된다. 메모리뷰 모듈에서 변수와 스택에서의 자료 흐름 및 자료 구조에 대한 처리 과정을 실시간으로 표현한다. 결과 창은 학습자로부터 변수 값을 직접 입력받도록 하여 본 시스템과 학습자간의 상호작용이 가능하도록 하고, 알고리즘 흐름에 따라 적절한 시점의 결과 값을 출력한다.

#### 2.1.4 명령어 해석 모듈

토큰을 분리하여 어휘 및 구문을 분석하는 모듈이다. 알고리즘 교육 시스템에서 학습자가 순서도를 작성하고, 변수를 입력하였을 경우 이에 대한 처리 과정을 위해 필요한 핵심 기술이다. 어휘 분석 단계에서 원시 프로그램의 어휘 단어를 식별자, 특수어, 연산자, 구분자 기호 등으로 분류하고, 구문 분석 단계에서는 어휘 분석기로부터 전달받은 어휘 단어를 사용하여 파스 트리(Parse Tree)라 불리는 계층적 구조를 생성한다.

### 2.1.5 소스코드 실행 과정 표현 모듈

학습자가 생성한 순서도를 진행하였을 경우 알고리즘 흐름에 맞는 소스코드를 확인할 수 있도록 하는 모듈이다. 프로그래밍 언어에 대한 기본 개념을 습득할 수 있도록 하여 알고리즘에 대해 깊이 있는 학습이 가능하도록 한다.

## 2.2 순서도 인터페이스 개발

작성하고자 하는 알고리즘에 맞게 순서도를 배치하기 위해 도구상자의 순서도 기호 목록에서 원하는 순서도 기호를 선택한 후 마우스를 이용하여 배치할 수 있도록 순서도 인터페이스를 구성하였다. 순서도 인터페이스는 작업창에 표시되는 순서도 기호와 순서도의 흐름을 나타내는 연결선으로 구성된다. 그림 2는 순서도 인터페이스 구현 화면이다.

순서도 기호 목록 중 원하는 기호를 선택하여 작업창을 클릭하여 배치한다. 배치 시에는 해당 기호에 대한 입력 예시 및 설명을 참조하면서 순서도 기호별 해당 내용을 대화상자를 통해 입력한다. 순서도 기호는 입력 기능, 처리 기능, 조건 기능과 반복 기능을 통하여 작성하고자 하는 알고리즘은 모두 작성이 가능하도록 하였다.

### 2.2.1 입력 기능

사용자로부터 데이터를 입력받는 과정으로 데이터 입력 시 출력할 메시지와 저장될 변수를 입력받는다.

### 2.2.2 처리 기능

각종 연산 과정 및 데이터 저장 등의 일을 수행하며, 정의 탭에서는 입력 대화상자에 변수의 초기값을

정의하며, 산술연산 탭에서는 사칙연산 및 논리연산을 수행하고, 문자연산 탭에서는 문자열의 연산을 수행한다.

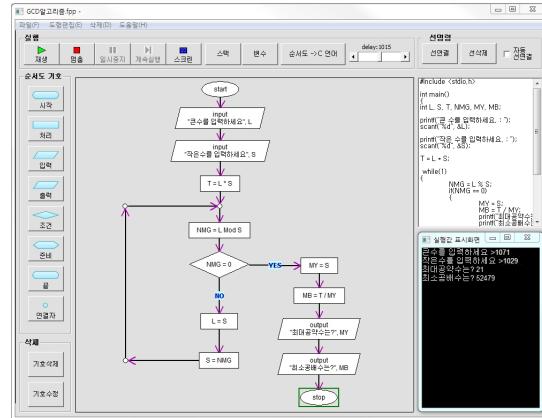


그림 2. 순서도 인터페이스

Fig. 2 Flowchart interface

### 2.2.3 조건 기능

알고리즘의 진행 과정 중 조건에 따라 분기하는 일을 수행하며, 두 개의 항 및 비교연산자를 입력 받아 분기한다.

### 2.2.4 연결선 기능

순서도 기호를 작업창에 배치시킨 후 작업의 순서 및 흐름을 결정하기 위해 연결선으로 순서도 기호를 연결한다.

### 2.2.5 반복 기능

For Loop나 While Loop 등의 반복문 처리 시 사용하며, 조건에 따라 해당 과정을 반복하게 된다.

## 2.3 소스 코드 인터페이스 개발

사용자가 배치한 순서도의 내용을 분석하여 입력, 처리, 조건 기호에 대해 자동으로 소스 코드를 생성해내고, 소스코드 영역에 표시하였다. 그림 3은 작성된 순서도에 따른 소스코드 인터페이스를 나타내는 화면이다.

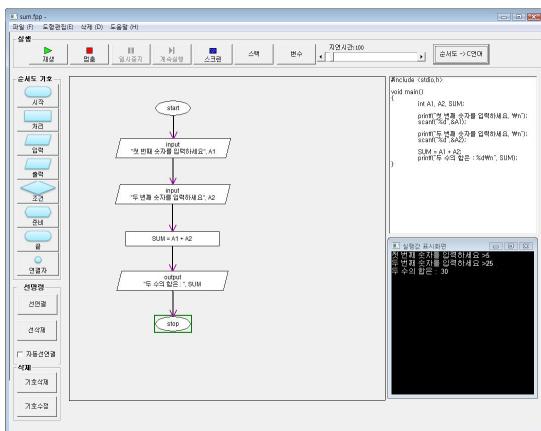


그림 3. 소스 코드 인터페이스  
Fig. 3 Source code interface

### 2.3.1 입력 기호

프로그램 실행 시 사용자에게 입력을 받기 위한 안내 메시지를 먼저 출력하고, 키보드를 통해 입력되는 값을 저장하기 위한 변수를 지정하는 소스 코드를 표기한다.

### 2.3.2 처리 기호

두 수의 사칙연산이나 문자 연산의 결과를 표기한다.

### 2.3.3 조건 기호

If문을 통한 비교 판단문에 대한 소스 코드를 표기 한다.

## 2.4 순서도 동적 뷰 개발

순서도를 통한 데이터 처리 과정을 동적으로 표시하여 알고리즘의 흐름을 이해하기 쉽도록 현재 단계의 순서도 기호에 대해 마킹 처리하고, 진행과정에 대해 한 단계씩 진행하거나 진행속도를 사용자가 자유롭게 설정할 수 있도록 한다.

그림 4는 1071과 1029의 최대공약수와 최소공배수를 구하는 예를 들어 나타낸다. (a)는 순서도를 실행시키면 알고리즘 교육 시스템은 결과 창을 통해 학습자에게 큰 수의 입력을 요청하게 되고, 학습자가 1071을 입력하면 변수  $I$ 에 1071이 대입되는 과정을 나타낸다.

낸다. (b)와 같이 작은 수의 입력을 요청하게 되면, 학습자가 1029를 입력 시 변수 S에 1029가 대입되는 과정이 진행되는 것을 알 수 있다. 변수 T에 1071과 1029를 곱한 값을 대입하게 되며, 변수 L을 변수 S로 나눈 나머지 42를 변수 NMG에 대입하게 되는 단계를 마킹 처리를 통해 보여주는 과정이다.

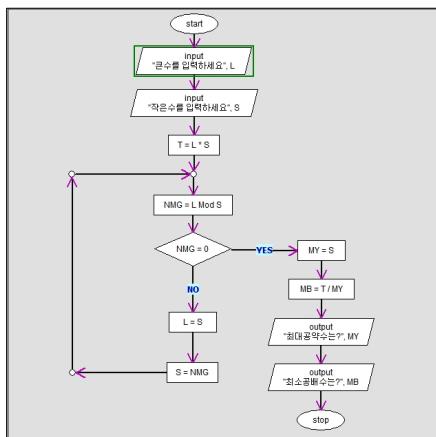
NMG가 0과 같은지를 판단하게 되며, NMG의 값이 42이므로 YES에 해당하는 과정을 진행하게 된다. 변수 S의 값인 1029를 변수 L에 대입하며, 변수 NMG의 값인 42를 S에 대입한다. 변수 L을 변수 S로 나눈 나머지 21을 변수 NMG에 대입하게 된다. NMG가 0과 다른지를 판단하게 되며, NMG의 값이 21이므로 YES에 해당하는 과정을 진행하게 된다. 변수 S의 값인 42를 변수 L에 대입하며, 변수 NMG의 값인 21을 S에 대입한다. 변수 L을 변수 S로 나눈 나머지 0을 변수 NMG에 대입하게 된다. NMG가 0과 같은지를 판단하게 되며, NMG의 값이 0이므로 NO에 해당하는 과정을 진행하게 된다. 변수 S의 값 21을 변수 MY에 대입하며, 변수 T의 값 1,102,059를 변수 MY의 값인 21로 나눈 값 52,479를 변수 MB에 대입한다. 이러한 과정을 동적 뷰에서 확인이 가능하며, (c)에서는 최대공약수에 변수 MY의 값인 21을 출력하고, 최소공배수에 변수 MB의 값인 52,479를 출력한 값을 보여주는 단계를 마킹 처리하여 나타낸다.

## 2.5 소스코드 동적 뷰 개발

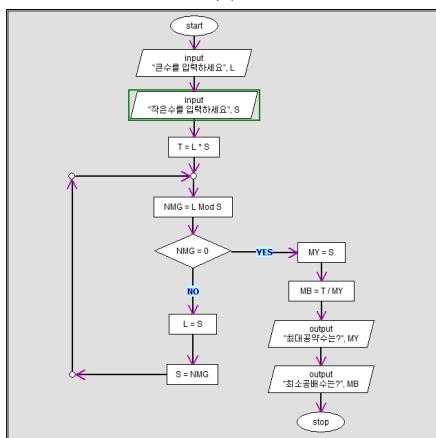
프로그램의 진행과정을 사용자가 이해하기 쉽도록  
현 진행단계의 소스에 대해 마킹 처리하여 알고리즘  
의 이해를 높일 수 있도록 하였다. 그림 5는 최대공약  
수와 최소공배수를 구하는 과정에 대해 소스코드의  
동적 변화 과정을 나타낸다.

III. 결 론

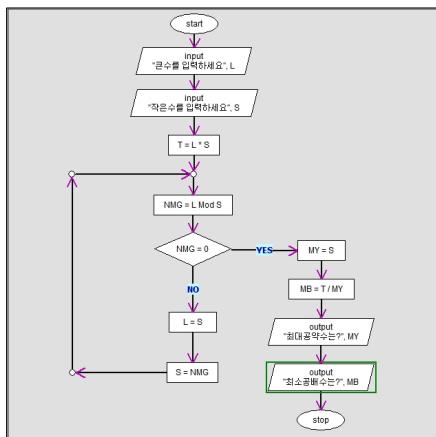
컴퓨터 프로그래밍 교육과정은 테이터의 구조에 대한 사전 지식을 필요로 하며, 수학적 사고 능력을 필요로 하므로 대부분의 학생들이 어려워하고 있으며, 실습 위주의 학습이 가능한 보조 프로그램을 필요로 하고 있다.



(a)



(b)



(c)

그림 4. 순서도 동적 뷰  
Fig. 4 Flowchart dynamic view

```

#include <stdio.h>
int main()
{
int L, S, T, NMG, MY, MB;
printf("값 수를 입력하세요, :");
scanf("%d", &L);
printf("작은 수를 입력하세요, :");
scanf("%d", &S);
T = L * S;
while(1)
{
    NMG = L % S;
    if(NMG == 0)
    {
        MY = S;
        MB = T / MY;
        printf("최대공약수는:");
        printf("최소공배수는:");
    }
}

```

```

#include <stdio.h>
int main()
{
int L, S, T, NMG, MY, MB;
printf("값 수를 입력하세요, :");
scanf("%d", &L);
printf("작은 수를 입력하세요, :");
scanf("%d", &S);
T = L * S;
while(1)
{
    NMG = L % S;
    if(NMG == 0)
    {
        MY = S;
        MB = T / MY;
        printf("최대공약수는:");
        printf("최소공배수는:");
    }
}

```

```

#include <stdio.h>
int main()
{
int L, S, T, NMG, MY, MB;
printf("값 수를 입력하세요, :");
scanf("%d", &L);
printf("작은 수를 입력하세요, :");
scanf("%d", &S);
T = L * S;
while(1)
{
    NMG = L % S;
    if(NMG == 0)
    {
        MY = S;
        MB = T / MY;
        printf("최대공약수는:");
        printf("최소공배수는:");
    }
}

```

```

#include <stdio.h>
int main()
{
int L, S, T, NMG, MY, MB;
printf("값 수를 입력하세요, :");
scanf("%d", &L);
printf("작은 수를 입력하세요, :");
scanf("%d", &S);
T = L * S;
while(1)
{
    NMG = L % S;
    if(NMG == 0)
    {
        MY = S;
        MB = T / MY;
        printf("최대공약수는:");
        printf("최소공배수는:");
    }
}

```

그림 5. 소스 코드동적 뷰  
Fig. 5 Source code dynamic view

본 논문에서는 프로그래밍 입문자들이 알고리즘의 개념 및 프로그래밍 언어에 대한 기본 지식을 습득할 수 있으며, 구조적인 프로그래밍이 가능하도록 순서도를 활용한 언어 학습 시스템을 설계하고 구현하였다. 이 시스템은 순서도를 작성하여 프로그램의 진행과정 및 데이터의 변화 과정을 쉽게 이해할 수 있다. 프로그래밍 교육과정은 프로그래밍 언어의 문법을 학습하는 것보다 프로그램을 어떻게 구조화시키는지가 더욱 중요하다고 할 수 있으므로 순서도를 활용하여 쉽게 데이터의 흐름을 파악할 수 있으며, 어떻게 구조화시켜 나가는지를 쉽게 학습할 수 있도록 하였다.

### 감사의 글

이 논문 또는 저서는 2008년 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (KRF-2008-313-D00001)

## 참고 문헌

- [1] Marc H. Brown, "Techniques for Algorithm Animation," IEEE Computer, Vol.2, No.1, pp.28-39, Jan. 1985.
- [2] John T. Stasko, "Tango: A Framework and System for Algorithm Animation," IEEE computer, Vol.23, No.9, pp.27-39, Sep. 1990.
- [3] Marc H. Brown, "Exploring Algorithms Using Balsa-II," IEEE Computer, Vol.21, No.5, pp.14-36, May 1988.
- [4] Marc H. Brown, "MacBALSA Version Alpha.3," Manuscript, 1989.
- [5] Marc H. Brown, "Zeus: A System for Algorithm Animation and Multi-View Editing," DEC. SRC research report 75, Feb. 1992.
- [6] Jone T. Stasko, "The Path-Transition Paradigm: A Practical Methodology for Adding Animation to Program Interfaces," Manuscript, College of Computing, Georgia Institute of Technology, June 1991.



김응곤(Eung-kon Kim)

1980년 2월 : 조선대학교 전자공학과 (공학사)

1986년 2월 : 한양대학교 컴퓨터공학과 (공학석사)

1992년 2월 : 조선대학교 컴퓨터공학과 (공학박사)

1993년 3월 ~ 현재 : 순천대학교 컴퓨터공학과 교수

※ 주 관심분야 : 영상처리, 컴퓨터 그래픽스, 멀티 미디어, HCI

## 저자 소개



박경욱(Kyoung-wook Park)

1996년 8월 : 순천대학교 전자계산학과 (이학사)

1999년 8월 : 전남대학교 전산통계학과 (이학석사)

2004년 8월 : 전남대학교 전산학과 (이학박사)

※ 주 관심분야 : 병렬 및 분산처리, 그래프 이론, 알고리즘



류남훈(Nam-hoon Ryu)

2007년 2월 : 한국방송통신대학교 컴퓨터과학과 (이학사)

2009년 2월 : 순천대학교 컴퓨터과학과 (이학석사)

2009년 3월 ~ : 순천대학교 컴퓨터과학과 박사과정 재학 중

※ 주 관심분야 : 영상처리, 컴퓨터 그래픽스, 알고리즘