



# 01

강

## 프로그래밍의 개요

### 학습목표

▶ 프로그램과 프로그래밍의 기본 개념에 대하여 이해할 수 있다.

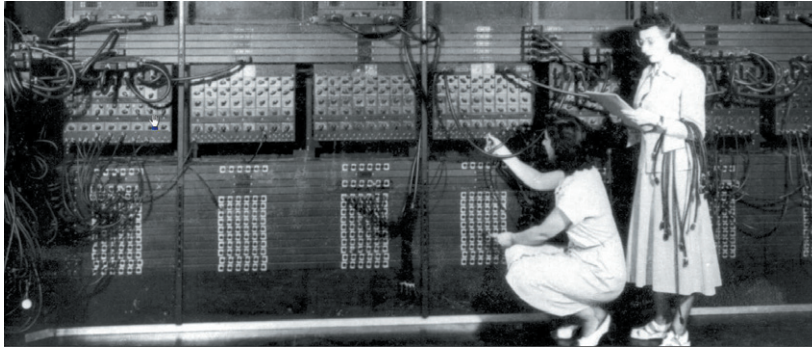


EBS tip

### 1 프로그램과 프로그래밍

#### (1) 프로그래밍의 역사

- ① 1940년대 이전 : 전선을 연결하는 방법으로 프로그램을 수행했으며, 필요한 작업을 위해 컴퓨터 내부에 있는 스위치 설정을 변경하였다.



▲ 전선을 연결하여 작업하는 에니악

- ② 1940년대 이후 : 폰 노이만의 프로그램 내장 방식 등장으로 전선 연결이 아닌 중앙 처리 장치가 수행해야 될 작업들을 일련의 명령 코드로 작성해서 컴퓨터 내부로 저장하고 프로그램을 실행하여, 수행해야 할 작업을 처리한다.

#### (2) 프로그래밍의 개념

- ① 컴퓨터를 통하여 문제를 해결하려면 문제 해결을 위한 작업의 순서를 지시해 주어야 한다. 이처럼 처리할 업무의 작업 수행 과정을 컴퓨터가 수행할 수 있는 명령문들로 차례로 나열한 것을 프로그램(program)이라 한다.
- ② 프로그래밍(programming)은 프로그램을 만드는 작업이나 과정이다. 즉, 컴퓨터에 어떤 문제를 처리할 수 있도록 일련의 명령문들을 작성하는 과정이나 절차를 말한다.
- ③ 프로그래머(programmer)는 프로그래밍을 하는 사람이다.
- ④ 동일한 문제라도 프로그램을 어떻게 작성하느냐에 따라서 수행 과정이나 처리 과정이 달라질 수 있고 프로그램의 효율이 달라진다.
- ⑤ 프로그램은 간결하고, 읽기 쉽고, 쓰기 쉽게 작성한다.

일상생활에서 행사의 일정이나 절차, 순서를 나타내는 것도 프로그램이라고 하는데 이는 컴퓨터 프로그램과 구별된다.

### 2 프로그래밍 언어(Programming Language)

#### (1) 프로그래밍 언어의 개념

- ① 프로그래밍 언어 : 컴퓨터로 프로그램을 작성할 때 사용하는 언어로 사람들이 대화할 때 사용하는 자연 언어와 구별된다.
- ② 프로그래밍 언어는 기계가 실행할 수 있고 사람이 이해하고 작성할 수 있는 형식으로 명령문을 기술하는 표현 체계이다.

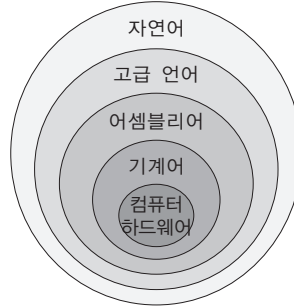
## (2) 프로그래밍 언어의 분류

누가 이해하기 쉬우냐에 따라 저급 언어와 고급 언어로 분류할 수 있다.

- ① 저급 언어 : 사람보다는 컴퓨터가 이해하기 쉬운 언어로서 하드웨어 의존도가 높은 프로그래밍 언어이다.
- ② 고급 언어 : 기계보다는 사람이 이해하기 쉬운 언어로서 사람이 일상생활에서 사용하는 자연어와 가까운 구조를 가진 프로그래밍 언어이다.



▲ 프로그래밍 언어의 분류



▲ 프로그래밍 언어의 계층 구조

- 절차 지향 언어 : 코볼, 포트란, C, 파스칼
- 객체 지향 언어 : C++, 자바

## 3 기계어(Machine Language)

- ① CPU가 직접 해독하고 실행할 수 있는 비트 단위로 쓰인 프로그래밍 언어이다.
- ② 컴퓨터는 on과 off의 전기 신호로 동작하므로 컴퓨터 기계에서 직접 실행하기 위해서는 on을 1로 off를 0으로 하여 명령어 코드를 작성하여야 하고, 이 언어가 기계어이다. 따라서 기계어는 0과 1의 2진수로 이루어져 있다.
- ③ 컴퓨터 기종에 따라 각기 다른 명령어 체계를 가지므로 기종이 다르면 사용하는 기계어도 달라진다.
- ④ CPU에 내장된 명령들을 이용하여 직접 작성하므로 프로그램 작성이 어렵고 복잡하다.
- ⑤ 컴퓨터가 바로 이해할 수 있으므로 수행 속도가 빠르다.

### [기계어 코드]

```

10101001 00000001 10100000 00000000 10011001
00000000 10000000 10011001 00000000 10000001
10011001 10000010 10011001 00000000 10000011
11001000 11010000 11110001 01100000
  
```

### [비주얼베이직 코드]

```

For I = 1 To 1000
  Print "A";
Next I
  
```

### [C 코드]

```

for (i=0; i<=999; i++) printf("A");
  
```

▲ 프로그래밍 언어별 코드 비교

## 4 어셈블리어(Assembly Language)

### (1) 어셈블리어의 특징

- ① 기계어는 실제로 컴퓨터 CPU가 읽어서 실행할 수 있는 0과 1로 이루어진 명령어의 조합이다. 이러한 각 명령어에 대해 사람이 알아보기 쉬운 니모닉 기호(mnemonic symbol)를 정해 사람이 좀 더 쉽게 컴퓨터의 행동을 제어할 수 있도록 한 언어가 어셈블리어이다. 즉, 어셈블리어는 0과 1로 이루어진 기계어에 1:1로 대응하는 기호로 이루어진 언어이다.
- ② 기계어보다는 쉽게 작성할 수 있다.
- ③ 기계어로 번역해 주는 번역기가 필요하다.
- ④ 기계 중심의 기호 언어로 작성되므로 프로그래머가 이해하기 어렵다.
- ⑤ 프로그램의 작성과 수정, 보완이 기계어보다는 쉽지만 고급 언어보다는 어렵다.
- ⑥ 처리 속도가 빠르며, 메모리를 효율적으로 관리할 수 있다.

### (2) 어셈블리어의 예

[x86 계열 CPU의 기계어]  
10110000 01100001

[어셈블리어]  
MOV AL, 061H

- ① 명령어 mov는 영어 move를 변형한 니모닉이며, al은 CPU 안에 있는 변수를 저장하는 레지스터의 하나이다. 그리고, 061h는 16진수 61 (즉, 십진수 97, 이진수 01100001)이다. 이 한 줄의 뜻은 16진수 61을 al 레지스터에 넣으라는 뜻이다.
- ② 1과 0의 반복인 기계어보다 사람이 혼동없이 이해하기가 한결 쉽다.

#### ▶ 어셈블리어 프로그램의 예

어셈블리 언어 프로그램	의미
MOV A, 55H	16진수 55를 레지스터 A에 넣는다.
MOV B, 3EH	16진수 3E를 레지스터 B에 넣는다.
ADD A, B	레지스터 A, B에 들어 있는 두 수를 더하라

## 5 고급 언어(High Level Language)

- ① 인간이 일상생활에서 사용하는 자연어에 가까운 표현을 사용하여 작성하는 언어이며, 기계어로 번역해 주는 번역기가 필요하다.
- ② 고급 언어의 특징
  - 자연어와 비슷한 구조로 되어 있어 프로그래머가 이해하기가 쉬우므로 배우기가 쉽고 프로그래밍하기가 용이하다.
  - 컴퓨터 기종과 운영 체제가 달라도 하나의 프로그램으로 여러 기종과 다른 운영체제에서 실행이 가능하기 때문에 프로그램의 이식성이 좋다.

- 고급 언어는 번역하는 방법에 따라 컴파일러 언어와 인터프리터 언어로 구별할 수 있다.

### ● 고급 언어의 종류

- 포트란(FORTRAN)
  - 과학 기술 계산용 언어
  - 복잡한 계산이나 수식을 처리하는 과학 계산 분야에 널리 사용되었음.
- 코볼(COBOL)
  - 주로 사무 처리에 사용
  - 일상 생활에서 사용하는 영어에 가까운 표현으로 명령문이 구성되어 있음.
- C 언어
  - 고급 언어이면서도 저급 언어 수준의 컴퓨터 하드웨어 제어가 가능한 언어
  - 운영 체제를 쉽게 구현하기 위해 개발된 언어
  - 간결한 표현, 유연한 제어 구조, 함수로 구성된 프로그램 등의 특징을 가지고 있음.
  - 시스템 프로그램을 작성하는 데 적합

- 프로그래머가 작성한 프로그램이 기계어로 번역되기 때문에 기계어 또는 기계어와 1대 1로 대응되는 어셈블리어로 작성된 프로그램보다는 처리 속도가 느리다.
- 고급 언어로 작업 시 개발 기간이 단축되며, 프로그램의 유지 보수가 쉽다.

구분	내용
절차 지향 언어	초기 고급 언어로서, 문제 해결을 위해 정해진 문법에 따라 일련의 처리 절차를 차례대로 기술하는 언어를 말한다. 예) 포트란, 코볼, C, 베이직
객체 지향 언어	각각의 소프트웨어 모듈을 하나의 객체(object)로 만들어 보관한다. 데이터와 이 데이터에 특정 조작을 가하는 메소드의 집합체인 클래스(class)를 지원한다. 예) 자바, C++, 델파이
비절차 언어	일상생활에서 사용하는 자연 언어와 유사한 언어이며, 대화 형식의 사용자 중심 언어로 보통 4세대 언어(4GL : 4th Generation Language)라고 한다. 예) 질의어(SQL : Structured Query Language)
함수 언어 (논리 언어)	컴퓨터가 인간 두뇌의 기능을 모방할 수 있게 해주는 응용 기술인 인공지능의 구현이나 전문가 시스템을 만드는 데 사용하며, 5세대 언어(5GL)라고 한다. 예) 리스프(LISP), 프롤로그(PROLOG)



#### ● 인공 지능 언어

- 리스프(LISP)
- 프롤로그(PROLOG)

## 6 저급 언어와 고급 언어의 비교

항목	저급 언어	고급 언어
처리 속도	빠르다.	느리다.
호환성	적다.	많다.
배우기	어렵다.	쉽다.
개발 중심	컴퓨터 H/W 중심	개발자 중심
유지 보수	어렵다.	쉽다.

#### ● 고급 언어의 특징

- 프로그래밍이 쉽다
- 이식성이 높다.
- 저급 언어보다 처리 속도가 느리다.
- 프로그램의 유지보수가 쉽다.

## 7 프로그래밍 언어가 갖추어야 할 조건

- ① 프로그램의 작성을 쉽게 하고, 읽고 이해하기 쉬우며, 프로그램의 오류를 최소화하기 위해서는 프로그래밍 언어의 개념이 단순 명료하고 통일성을 가져야 한다. 프로그래밍 언어가 복잡하면 프로그래머가 명령어 사용에서 혼동을 일으킬 수 있고, 애매모호하면 프로그래머의 의도와는 다른 명령어로 사용될 수 있다.
- ② 프로그래밍 언어의 구조가 체계적이어야 한다. 프로그래머의 사고 체계와 유사하여야 한다.
- ③ 응용 문제에 자연스럽게 적용할 수 있어야 한다. 즉 주어진 문제의 해결을 위한 적절한 자료 구조, 연산, 제어 등의 문법을 제공해야 한다.
- ④ 언어의 확장이 용이하여야 한다. 새로운 기능을 추가할 때 쉽게 추가와 수정을 할 수 있는 구조를 가져야 한다.



- ⑤ 적은 기억 장소를 사용하고, 짧은 시간 내에 실행할 수 있게 효율적이어야 한다.
- ⑥ 필요한 경우 운영 체제의 지원을 받을 수 있어야 하고, 다른 프로그램과 쉽게 연결할 수 있도록 외부적인 지원이 가능해야 한다.
- ⑦ 프로그래머의 아이디어를 자연스럽게 표현할 수 있는 문법 구조를 가져야 한다.
- ⑧ 프로그램을 작성한 후 반드시 테스트와 검증을 거쳐야 한다. 작성된 프로그램이 정상적으로 실행되는가 모든 조건에서 에러 없이 실행되는가 등을 테스트하고 검증하여야 한다. 프로그래밍의 생산성과 프로그램의 정확성을 높이기 위해서 프로그래밍 언어는 이러한 테스트와 검증이 용이한 툴을 제공하여야 한다.

## 8 프로그래밍 언어의 선정 기준

- ① 프로그래머가 이해하고 사용하기 쉬운 언어이어야 한다.
- ② 응용 목적에 맞아야 한다.
- ③ 프로그래밍에 효율적인 언어이어야 한다.



# 기출 모의고사

정답 및 해설 p. 1

1

2005학년도 대수능

컴퓨터 프로그램과 관련된 설명으로 옳은 것을 <보기>에서 고른 것은?

**보기**

- ㄱ. 컴퓨터는 프로그램이 없어도 스스로 일을 처리한다.
- ㄴ. 컴퓨터 프로그램을 작성하는 사람을 프로그래머라 한다.
- ㄷ. 코딩(coding)이란 프로그래밍 과정에서 생성되는 오류를 수정하는 것이다.
- ㄹ. 프로그램은 컴퓨터가 수행해야 할 일의 처리 방법과 순서를 지시하는 명령문의 집합이다.

- ① ㄱ, ㄴ      ② ㄱ, ㄷ      ③ ㄴ, ㄷ
- ④ ㄴ, ㄹ      ⑤ ㄷ, ㄹ

2

2011학년도 대수능

다음 대화에 나타난 프로그래밍 언어에 대한 설명으로 옳은 것만을 <보기>에서 있는 대로 고른 것은?

- 철수 : 오늘 학교에서 C 언어로 로봇을 제어하는 프로그램을 배웠는데 재미있었어.
- 영희 : 그렇구나! 우리는 비주얼베이직 언어로 성적 처리하는 프로그램을 배우고 있는데.

**보기**

- ㄱ. 비주얼베이직 언어는 고급 언어로 분류된다.
- ㄴ. C 언어는 컴파일러를 언어 번역기로 사용한다.
- ㄷ. 어셈블리어에 비해 프로그램 작성과 수정이 쉽다.

- ① ㄱ      ② ㄴ      ③ ㄱ, ㄷ
- ④ ㄴ, ㄷ      ⑤ ㄱ, ㄴ, ㄷ

3

2010학년도 대수능

다음 대화 내용을 모두 만족하는 프로그래밍 언어로 옳은 것을 <보기>에서 고른 것은?



**보기**

- ㄱ. 자바      ㄴ. C++
- ㄷ. 기계어      ㄹ. 포트란

- ① ㄱ, ㄴ      ② ㄱ, ㄷ      ③ ㄴ, ㄷ
- ④ ㄴ, ㄹ      ⑤ ㄷ, ㄹ

4

다음은 프로그램 개발자들 간의 대화이다. 모든 개발자를 만족시킬 수 있는 언어로 가장 적절한 것은?

- 강산 : 속도가 빨라야 하므로 컴파일 방식의 언어이어야 하는데.
- 철수 : 하드웨어를 직접 제어해야 할 부분이 많으므로 저급 언어적 특성도 갖는 프로그래밍 언어이어야 해.
- 영희 : 다양한 기종에서 사용하려면 호환성이 좋은 언어이어야 해.
- 우석 : 변화가 많은 업무환경에서 사용하려면 새로운 업무가 추가될 때마다 모듈로 삽입하여 쉽게 수정·개발할 수 있어야 해.

- ① C      ② COBOL      ③ LISP
- ④ 기계어      ⑤ 어셈블리어

## 5 다음 중 프로그래밍에 대해 바르게 말한 사람을 고른 것은?

우석 : 프로그래밍 언어 중에서 처음 사용한 것은 코볼이었어.  
 영희 : 포트란은 널리 사용했던 고급 언어 중에서는 제일 먼저 나온 것으로 과학 기술 분야에 많이 사용했지.  
 철수 : 프로그래머가 작성한 원시 프로그램은 인터프리터를 사용해야만 기계어로 번역할 수 있어.  
 순희 : 알고리즘은 문제를 해결하는 데 필요한 자료들과 논리적인 절차를 알기 쉽게 기술하는 것으로 프로그램 교육에 매우 필요한 거야.

- ① 우석, 영희      ② 우석, 철수  
 ③ 우석, 순희      ④ 영희, 철수  
 ⑤ 영희, 순희

## 6 다음은 프로그래밍에 대한 설명이다. <보기>에서 옳은 것을 고른 것은?

### 보기

- ㄱ. 기계어와 어셈블리어는 저급 언어에 속한다.  
 ㄴ. 알고리즘이란 문제 해결을 위한 방법 및 절차를 말한다.  
 ㄷ. 프로그램의 오류 수정 작업을 디버깅(debugging)이라 한다.  
 ㄹ. 컴파일러 방식은 프로그램을 명령문 단위로 번역하여 실행한다.  
 ㅁ. 저급 언어로 작성된 프로그램은 컴퓨터 기종 간의 이식성이 좋다.

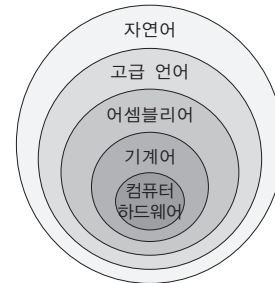
- ① ㄱ, ㄴ, ㄷ      ② ㄱ, ㄷ, ㄹ  
 ③ ㄱ, ㄷ, ㅁ      ④ ㄴ, ㄷ, ㄹ  
 ⑤ ㄴ, ㄷ, ㅁ

## 7 영희네 조에서 프로그래밍에 대하여 발표할 자료를 만들기 위해 대화한 내용의 일부이다. 옳지 않게 말한 사람은?

강수 : 복잡한 프로그램은 내용을 알기 쉽도록 주석을 붙여 주는 것이 좋아.  
 난희 : 다양한 프로그래밍 언어를 혼합하여 프로그래밍해야 프로그램의 유지 보수가 쉬워.  
 박연 : 변수 이름은 의미를 가질 수 있도록 붙이는 것이 프로그램을 해석하는 데 도움이 되지.  
 은수 : 프로그래밍하는 과정에서 생성되는 오류를 수정하는 일을 디버깅(debugging)이라고 해.  
 화자 : 언어 번역 프로그램은 컴퓨터가 처리할 수 없는 프로그램을 컴퓨터가 직접 처리할 수 있는 기계어로 만들어 줘.

- ① 강수      ② 난희      ③ 박연  
 ④ 은수      ⑤ 화자

## 8 다음 그림은 프로그래밍 언어의 일반적인 계층 구조이다. 고급 언어에 해당하는 언어의 특징들을 <보기>에서 고른 것은?



### 보기

- ㄱ. 번역 과정 없이 바로 실행된다.  
 ㄴ. 자연어와 비슷한 구조로 되어 있어 배우기 쉽다.  
 ㄷ. 기계어를 대신하여 이해하기 쉬운 기호로 명령을 만든 기호 언어이다.  
 ㄹ. 컴퓨터 기종이 달라도 프로그램의 이식성이 좋다.  
 ㅁ. 컴퓨터 기종에 따라 사용하는 언어가 다르다.

- ① ㄱ, ㄹ      ② ㄱ, ㅁ      ③ ㄴ, ㄷ  
 ④ ㄴ, ㄹ      ⑤ ㄷ, ㄹ