



이현진 교수님

컴퓨터개론

7주. 프로그래밍 언어

사용서체: 나눔글꼴



숭실사이버대학교

숭실사이버대학교의 강의콘텐츠는
저작권법에 의하여 보호를 받는바,
무단 전재, 복제, 배포, 전송, 대여 등을
금합니다.

지난 학습 내용



데이터 처리 원리

데이터의 표현 비트, 바이트, 워드, 아스키코드, 유니코드

진법변환 10진법, 2진법, 8진법, 16진법

중앙처리장치 기계사이클(인출, 해독, 실행), 파이프라이닝



학습목차

1

프로그래밍 언어

2

프로그래밍

3

코딩

1. 프로그래밍 언어



프로그래밍 언어 개요

▶ 프로그래밍 언어의 필요성

- 사람이 컴퓨터에게 지시할 명령어를 기술하기 위함

▶ 고급언어와 저급언어

- 고급언어
 - ✓ 인간이 이해하기 쉽고 친근한 언어
- 저급 언어
 - ✓ 컴퓨터가 이해하기 쉬운 언어

저급언어

➤ 기계어

- 기계어란 0과 1로 표현
- 컴퓨터가 직접 이해할 수 있는 유일한 언어
- 연산 코드(Operation Code)와 피연산자(Operand)로 구성
- 프로그래밍하기가 상당히 어려움

➤ 어셈블리어

- 복잡한 기계어를 간략하게 기호화(symbolize)함
- 기계어의 연산코드와 피연산자를 프로그래머가 이해하기 쉬운 기호형태로 일대일 대응시킨 언어
- 연산 코드를 기호화한 것을 니모닉(mnemonic)이라 함
- 연산 코드의 예로 LDA(Load Address), ADD(ADD), STA(STore Address) 등이 사용
- 하드웨어 장치에 대한 강력한 통제 가능

기계어와 어셈블리어 비교

순서	기계어	어셈블리어	의미
명령어1	0101000000000100	LDA A	메모리 A의 내용을 누산 레지스터(AC)에 저장
명령어2	0111000000000110	ADD B	메모리 B의 내용과 누산 레지스터(AC)의 값을 더하여 다시 누산 레지스터(AC)에 저장
명령어3	0100000000000111	STA C	누산 레지스터(AC)의 값을 메모리 C에 저장
명령어4	0011000000000000	HLT	프로그램 종료

저급언어 (Low Level Language)

- 컴퓨터의 주기억장치, 레지스터, 마이크로프로세서, 입출력 포트 등의 하드웨어를 직접 통제 가능
- 저급 언어를 사용하기 위해서는 하드웨어에 대한 충분한 지식 필요
- 고급 언어(High Level Language)에 비하여 언어 자체가 어렵기 때문에 전문가라 하더라도 프로그램의 생산성이 낮음

고급 언어(High Level Language)

- ▶ 1950년 중반부터 인간의 자연 언어와 비슷한 포트란, 코볼, 베이직 등의 고급 언어가 개발되어 프로그램 개발에 사용



프로그래밍 언어의 세대별 분류

세대	시기	기능
1세대	1945년	컴퓨터가 이해하는 유일한 언어인 기계어만을 이용한 세대이며 현재에도 기계어는 이용되고 있다.
2세대	1950년 중반	어셈블리어가 개발되고 어셈블러가 개발되어 프로그램 개발의 생산성이 높아진 세대이나 시스템마다 어셈블리어는 다르므로 시스템 호환 문제가 계속 남아 있었던 세대이다.
3세대	1960년 초반	포트란, 알골, 베이직, 파스칼 같은 고급 언어와 컴파일러가 개발되었고 시스템에 독립적인 프로그램을 개발하여 프로그램 개발의 생산성이 매우 높아진 세대이다.
4세대	1970년 초반 이후	비절차 중심의 언어로 보고서 생성기와 데이터베이스 질의 언어(query language) 또는 비주얼 베이직과 같은 비주얼 프로그래밍 언어이다.
5세대	현재와 미래	영어, 한국어와 같은 자연 언어로 아직 진정한 의미의 자연 언어는 없으며, 컴퓨터에 대한 기초지식이 없는 일반인도 코드 없이 프로그램을 만들 수 있는 블록 비주얼 프로그래밍 언어가 소개되어 교육용 프로그래밍 언어로 많이 활용되고 있다.

비절차적 언어

➤ 데이터베이스 질의 언어(query language)

- 비절차적 중심 언어
- 프로그래머가 원하는 결과를 얻기 위해서 하나 하나의 절차를 따를 필요가 없다는 의미

➤ 비주얼 프로그래밍 언어(Visual Programming Language)

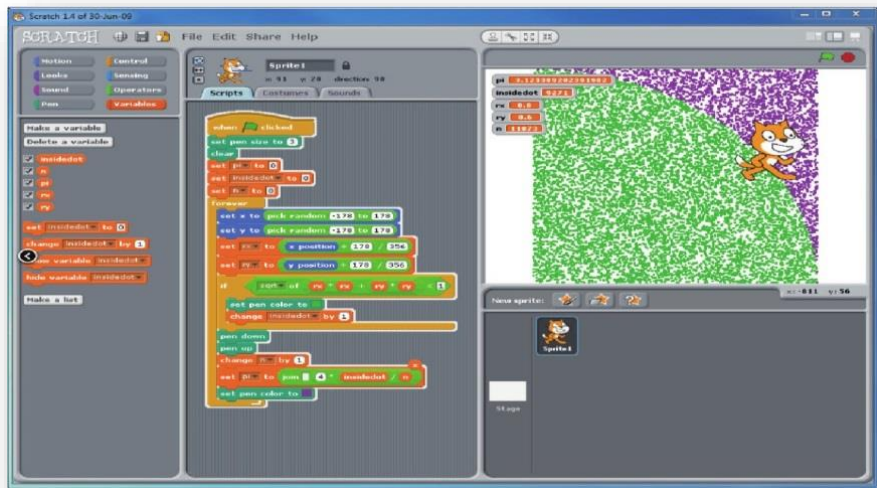
- 비주얼 베이직
- 델파이



비절차적 언어

▶ 블록 프로그래밍

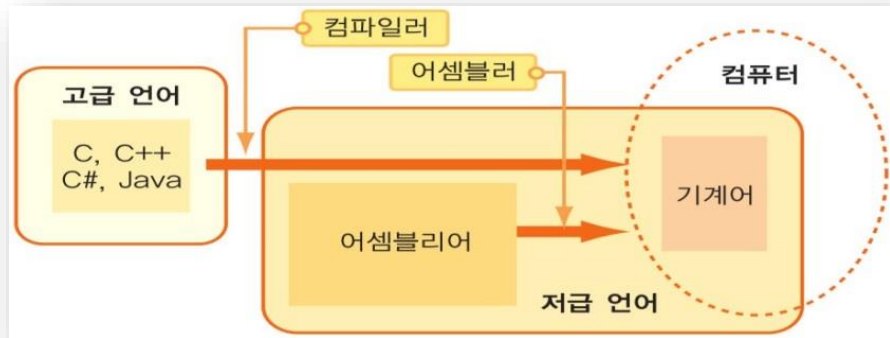
- 코드 없는 정해진 블록으로 코딩
- 진보된 비주얼 프로그래밍 언어
- MIT에서 개발한 스크래치



컴파일러와 어셈블리어

▶ 어셈블리어

- 어셈블리 언어의 프로그램을 기계어로 변환



컴파일러와 인터프리터(1)

▶ 인터프리터(Interpreter)

- 고급 언어를 기계어로 번역해주는 역할을 수행
- 원시 코드를 한 줄씩 읽어 들여 목적 코드로 바꾸어줌
- 컴파일러에 비해 번역 속도가 느릴 수 밖에 없지만, 프로그램을 작성할 때 보다 융통성을 가질 수 있음.

▶ 컴파일러(Compiler)

- 원시 코드 전체를 읽은 다음 이를 기계어로 번역
- 컴파일러는 한 번 컴파일 한 후에는 수정이 없다면 매번 컴파일 할 필요 없이 빠른 시간 내에 프로그램 실행이 가능

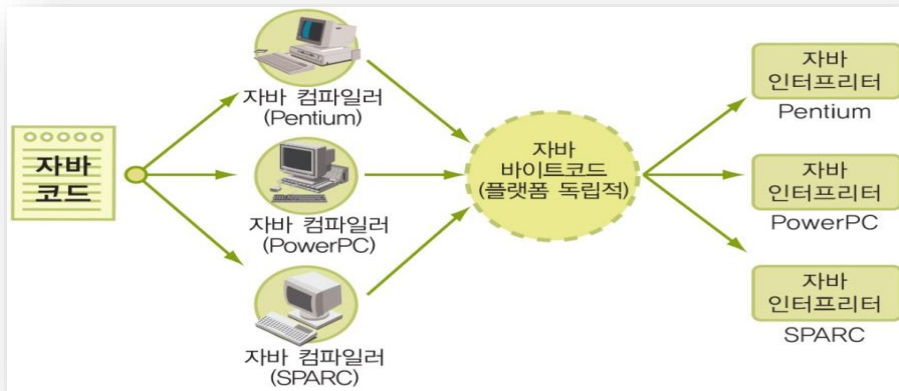
컴파일러와 인터프리터(2)

특징 \ 방식	컴파일러	인터프리터
번역 방법	프로그램 전체 번역	실행되는 줄(라인) 단위 번역
장점	한 번 컴파일한 후에는 매번 빠른 시간 내에 전체 실행 가능	번역 과정이 비교적 간단하고 대화형 언어에 편리함
단점	프로그램의 일부를 수정하는 경우에도 전체 프로그램을 다시 컴파일해야 함	실행할 때마다 매번 기계어로 바꾸는 과정을 다시 수행해야 하므로 항상 인터프리터가 필요함
출력물	목적 코드	즉시 실행
언어 종류	FORTTRAN, COBOL, C 등	BASIC, PYTHON 등

컴파일러와 인터프리터 중간 방식

▶ 컴파일러와 인터프리터의 특징을 모두 갖는 방식

- 자바 언어와 C# 언어
- 이들 언어는 컴파일러가 존재하여 컴파일 과정이 필요
- 컴파일된 실행 파일을 실행할 때 인터프리터 방식과 같이 인터프리터가 필요
- 모든 시스템에서 독립적인 프로그램 언어를 개발 위해



2. 프로그래밍



프로그래밍 개요

➤ 프로그램(program)

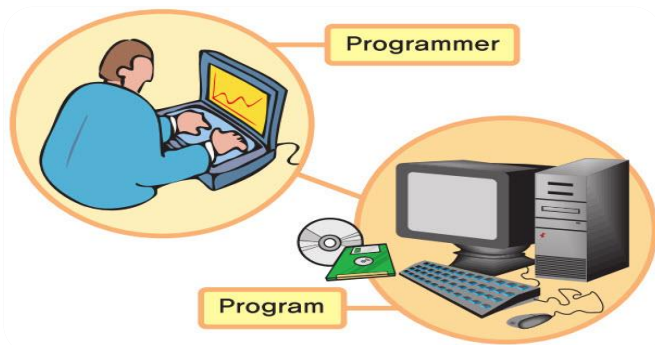
- 컴퓨터에서 특정 목적의 작업을 수행하기 위해 관련된 명령어와 자료를 모아 놓은 것
- 컴퓨터에게 지시할 일련의 처리 작업 내용

➤ “프로그래밍한다”

- 컴퓨터에서 특정 목적의 일을 수행하는 프로그램을 만드는 과정

➤ 프로그래머(Programmer)

- 프로그램을 만드는 사람



프로그램 개발 환경

➤ 개발 도구(development tools)

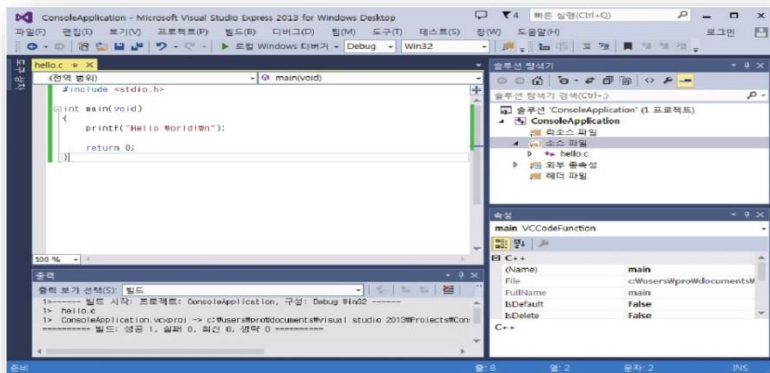
- 선택한 언어를 이용하여 프로그램을 개발하기 위한 환경 필요

➤ 개발 도구

- 편집기(Editor)
 - ✓ 프로그램 명령어인 프로그래밍 언어의 내용을 편집
- 컴파일러(Compiler)
 - ✓ 개발 도구에서 가장 중요한 것은 작성한 고급 프로그래밍 언어를 컴퓨터가 이해할 수 있는 기계어로 변환
- 디버거 (debugger)
 - ✓ 작성된 프로그램에서 발생하는 프로그램 오류를 쉽게 찾아 수정 할 수 있도록 도와주는 프로그램
- 링커(linker)
 - ✓ 여러 목적 파일을 하나의 실행 파일로 만들어 주는 기능

통합개발환경(Integrated Development Environments)

- ▶ 프로그램을 개발하는데 필요한 컴파일러, 디버거, 링커, 에디터 등을 통합적으로 제공하는 개발 환경
- ▶ 마이크로소프트사의 '비주얼 스튜디오'
 - <https://www.visualstudio.com/ko/>



프로그램 구현 과정(1)

➔ 소스 작성

- 프로그램 언어를 이용하여 원하는 작업을 기술한 내용을 소스 코드(source code) 또는 간단히 코드(code)라 함

[illegible]

프로그램 구현 과정(2)

▶ 컴파일

- 소스(source) 파일(원시 파일)을 목적 파일(Object file)로 변환 과정
- 컴파일러
 - ✓ 컴파일을 해주는 프로그램



프로그램 구현 과정(3) - 링크

➤ 링커

- 목적 파일을 실행 가능한 실행 파일(Eexecute File)로 만들어 주는 프로그램
- 여러 개의 프로그램 각각을 컴파일하여 목적 파일을 만들게 되는데, 링커는 이러한 여러 개의 목적 파일들을 라이브러리 함수와 연결해서 하나의 파일로 합치는 작업을 수행

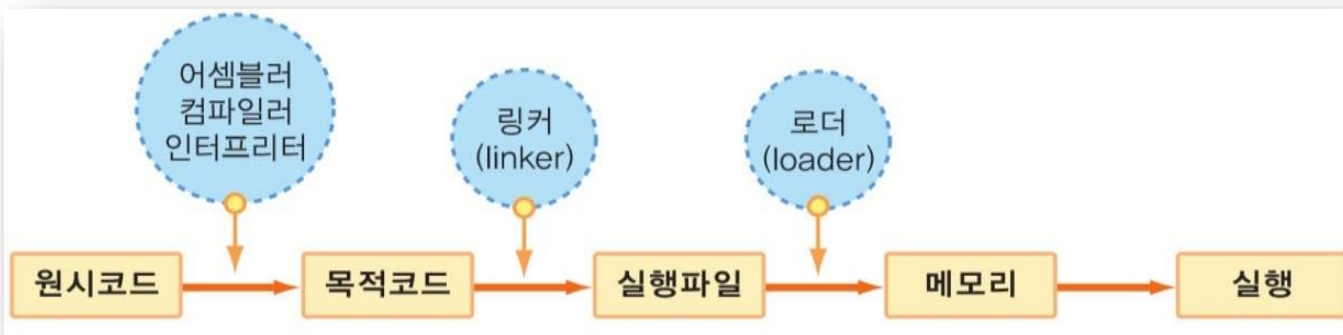
➤ 링킹(linking)

- 과정

프로그램 구현 과정(4) - 실행

▶ 실행 파일(executable program)

- 링킹 작업 결과 생성되는 프로그램.exe나 .com의 확장자 파일



프로그램 구현 과정(5) - 디버깅

▶ 디버깅(debugging)과 디버거

- 컴파일 에러나 실행 에러를 수정하는 과정
- 프로그램의 명령을 수행함에 있어 컴퓨터의 상태를 보여주거나 오류(또는 에러) 발생시 오류를 쉽게 찾을 수 있도록 도와주는 프로그램
- 처음부터 에러가 없는 프로그램을 작성하기란 거의 불가능
- 디버깅 과정에서는 많은 경험으로 에러를 쉽게 찾아 수정

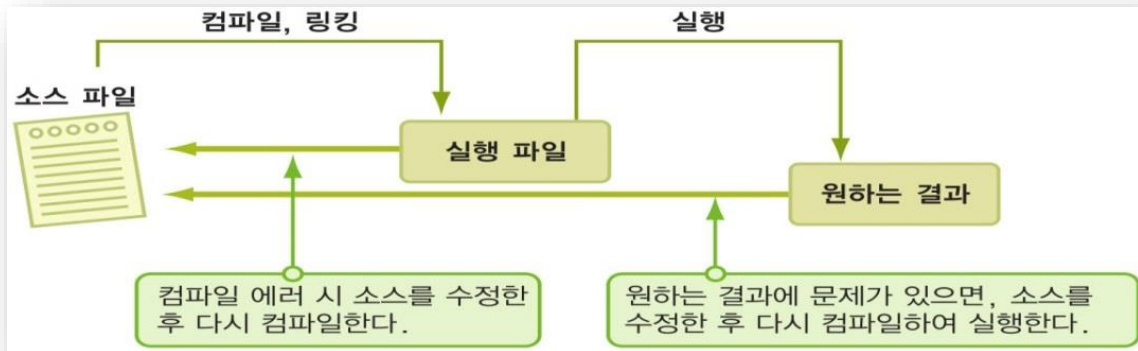
프로그램 구현 과정(6) - 디버깅

▶ 에러 또는 오류

- 컴파일 (시간) 에러
- 실행 (시간) 에러

▶ ‘디버그’란 용어의 유래

- 원래 버그는 나방 벌레
- 버그에 의해 컴퓨터가 오작동된 이유에서 유래



3. 코딩



컴퓨터과학의 4단계

➤ 정보의 표현

- 컴퓨터가 수, 그림, 글자, 컬러 등의 정보를 디지털화하여 표현하는 방법을 배움

➤ 자료 구조

- 이러한 정보를 저장장치와 메모리에 어떻게 저장하고 불러오는지 배움

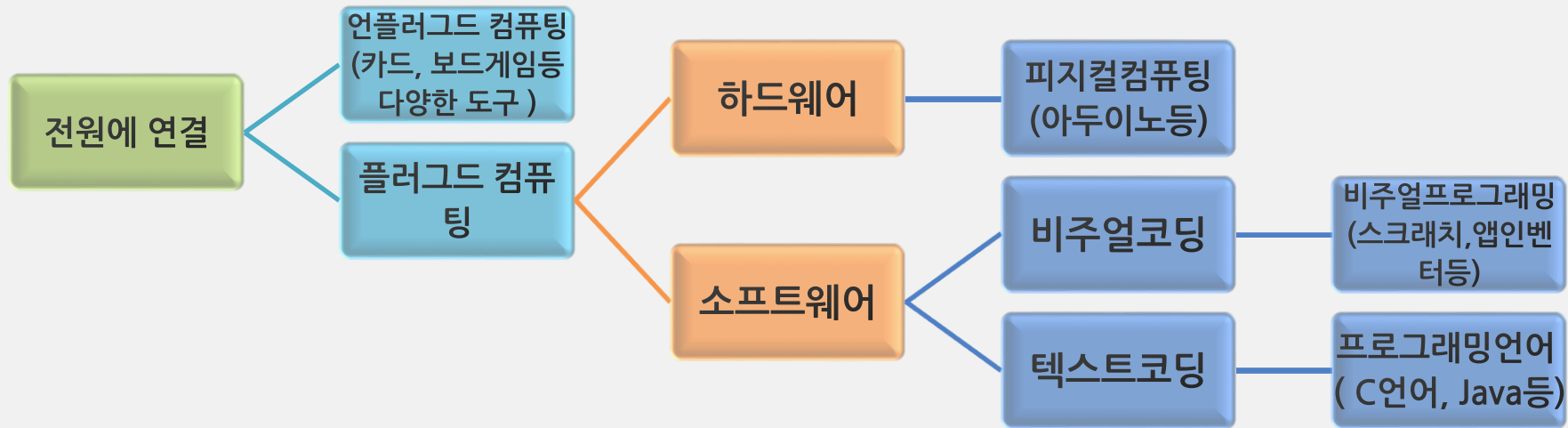
➤ 알고리즘

- 여러 가지 정보를 가공하여 문제를 해결하거나 새로운 정보를 만들어내는 절차와 방법을 배움

➤ 프로그래밍

- 자료구조와 알고리즘을 이용하여 실제로 프로그램을 만들어 봄

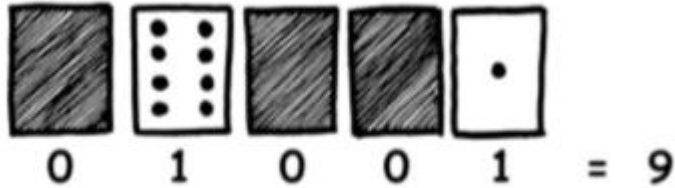
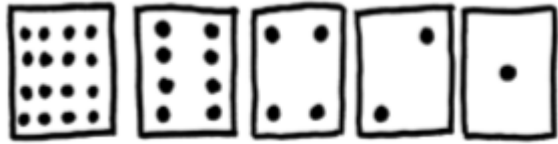
코딩의 유형



언플러그드 컴퓨팅(unplugged computing)

- ▶ 컴퓨터 없이 컴퓨터 과학적 사고를 향상시킬 수 있는 학습 활동
- ▶ 뉴질랜드의 Tim Bell교수가 컴퓨터가 없는 환경에서 컴퓨터 과학 원리를 학습할 수 있는 놀이를 개발한 교수학습방법으로 놀이를 통해 컴퓨터원리를 학습하여 문제해결에 있어서 컴퓨터원리를 적용할 수 있는 능력을 신장
 - <https://csunplugged.org/en/>

이진수 카드 세기



<http://computing.or.kr/12024/%ec%a0%90%ec%9d%98-%ec%88%98-%ec%84%b8%ea%b8%b0-%ec%9d%b4%ec%a7%84%ec%88%98csunplugged-v3-1/>

피지컬 컴퓨팅

- ▶ 현실 세계 속 데이터를 디지털 기기로 내려받아 소프트웨어의 형태로 처리한 후, 그 결과를 모니터나 LED 또는 로봇 같은 여러 가지 장치로 출력
- ▶ 피지컬 컴퓨팅은 사물인터넷 분야의 기초

로코드, 노코드

- ▶ 로코드(low-code), 노코드(no-code) 플랫폼 개발로 코드와 노코드 툴을 통해 일반적인 비즈니스 사용자도 빠르고 쉽게 새로운 앱을 만들거나 기능을 추가가능
- ▶ 장점
 - 앱 제공 속도가 높아지고, 작업자들이 더 나은 제품을 만들도록 유도

<http://www.itworld.co.kr/news/107328#csidx0c26e5fc73196cf28743f472e0ce5>

파이썬

- 1990년 네덜란드의 개발자인 귀도 반 로섬(Guido Van Rossum)이 개발해 발표한 프로그램 언어
- CWI(Centrum voor Wiskunde en Informatica)라는 국립 수학 및 컴퓨터 과학 연구기관에 근무하면서 인터프리터 언어를 개선하는 프로젝트를 맡고 분산시스템을 연구한 경험을 토대로 개발한 언어
- 파이썬이 인기를 누리는 이유

- 높은 가독성과 생산성

<http://www.software.kr/um/um03/um0305/um030503/um030503View.do?postId=29848>

Python	JAVA	C
<pre>print("Hello, World");</pre>	<pre>public class Main { public static void Main (String[] args) { System.out.println("Hello, World"); } }</pre>	<pre>#include <stdio.h> int main() { printf("Hello, World"); return 0; }</pre>



학습정리

1

프로그래밍언어 고급언어,저급언어, 컴파일러, 인터프리터

2

프로그래밍 편집기,컴파일러, 링커 로더, 디버거

3

코딩 언플러그드 컴퓨팅, 피지컬 컴퓨팅

7주. 프로그래밍 언어



이번 강의를 마칩니다.
수고하셨습니다.