

■ 프로그래밍 언어란?

컴퓨터 시스템을 동작시키기 위한 프로그램 언어 명령어들의 조합, 이진법으로 작성되거나 변환되어 컴퓨터가 이해할 수 있도록 한다.

-Bit

Binary Digit의 약칭, 이진법을 이용, 1byte = 8bit

-Byte

1B = 8bit, 1bit = 0.125B

-Kilobyte

1KB = 8,192bit, 1bit = 0.000122KB

-Megabyte

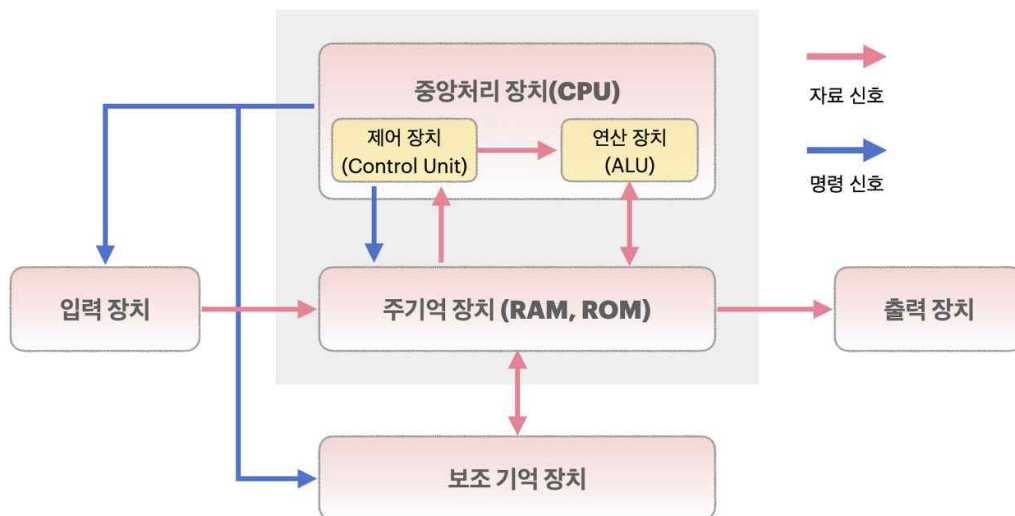
1MB = 8,388,608bit, 1bit = 1.1921e-7MB

-Gigabyte

1GB = 8.5899e+9bit, 1bit = 1.1642e-10GB

-Terabyte

1TB = 8.7961e+12bit, 1bit = 1.1369e-13TB



■ 프로그램의 구성 요소

■ 자료구조

자료구조는 컴퓨터에 데이터를 삽입, 삭제, 수정하게 해주는 논리적인 공간 구조를 의미한다.

-단순 구조Simple

프로그래밍 언어에서 제공하는 기본 데이터 타입 (Int, float, double, char 등)

-선형구조Liner

자료들 사이의 선후 관계가 일대일인 구조 (연결 자료 구조, 순차 자료 구조, 스택, 큐, 텍)

-비선형구조Non-Liner

자료들 사이의 선후 관계 계층 또는 그물 형태를 가지는 구조 (트리, 그래프 등)

-파일 구조File

보조 기억 장치에 데이터 값이 실제로 기록되는 자료구조 (순차파일, 색인파일 등)

■ 알고리즘

넓은 의미에서 자료 구조와 함께 프로그램을 구성하는 요소, 좁은 의미에서는 어떤 문제에 대한 답을 찾는 해법 이다.

알고리즘의 표현은 자연어, 순서도, 의사코드, 프로그래밍 언어를 이용하는 방법이 있다.

-입력

외부로부터 입력되는 자료가 0개 이상이어야 함.

-출력

출력되는 결과가 1개 이상이어야 함.

-명확성

각 명령어의 의미가 명확해야 함.

-유한성

정해진 단계를 지나면 종료되어야 함.

-유효성

모든 명령은 실행이 가능한 연산들이어야 함.

■ 구성요소

-변수

어떤 값을 주기억 장치에 기억하기 위해서 사용하는 공간

-식별자

프로그램의 구성 요소를 구별하기 위한 기준, 변수명이 식별자에 속함.

-바인딩

변수와 변수에 관련된 속성을 연결하는 과정으로, 정적 바인딩과 동적 바인딩으로 구분1)정적 바인딩(Static

binding) : 프로그램 실행 시간 전에 속성을 연결하는 방식2)동적 바인딩(Dynamic binding) : 프로그램 실행시

간에 속성을 연결하는 방식

-선언

변수에 이름, 데이터타입 등의 속성을 부여하는 작업으로, 명시적 선언과 묵시적 선언으로 구분1)명시적 선언 : 선언문을 이용하여 변수 이름을 나열하고 속성을 부여하는 방식2)묵시적 선언 : 별도의 선언문 없이 디폴트 규칙에 의해 속성이 부여되는 방식

-할당

변수에 메모리 공간을 바인딩(정적/동적) 하는 작업

■ 데이터 타입의 유형

-Boolean 타입

True or False 조건이 참인지 거짓인지 판단하고자 할 때 사용

-문자 타입

문자 하나 (예 : 'A')를 저장하고자 할 때 사용

-문자열 타입

나열된 여러개의 문자를 저장하고자 할 때 사용

-Integer 타입

정수값을 저장하고자 할 때 사용

-float 타입

소수점을 포함하는 실수값을 저장하고자 할 때 사용

-배열 타입

여러 데이터를 하나로 묶어서 저장하고자 할 때 사용

■ Java 의 정수타입

데이터형	바이트 수	데이터 범위
byte	1byte	-128 ~ 127
short	2byte	-32768 ~ 32767
int	4byte	-214783648 ~ 214783647
long	8byte	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807

■ 연산자 종류

-산술연산자

+양쪽의 값을 더한다.

-왼쪽에서 오른쪽 값을 뺀다.

*두 개의 값을 곱한다.

/왼쪽값을 오른쪽 값으로 나눈다.

%왼쪽값을 오른쪽 값으로 나눈 나머지를 계산한다.

-시프트 연산자(비트를 이동시키는 연산자)

<< 왼쪽값을 오른쪽 값만큼 비트를 왼쪽으로 이동시킨다.

>> 왼쪽값에 오른쪽 값만큼의 부호 비트를 채우면서 왼쪽으로 이동시킨다.

-관계연산자

>왼쪽 값이 오른쪽 값보다 크면 true, 아니면 false

<왼쪽 값이 오른쪽 값보다 작으면 true, 아니면 false

>=왼쪽 값이 오른쪽 값보다 크거나 같으면 true, 아니면 false

<=왼쪽 값이 오른쪽 값보다 작거나 같으면 true, 아니면 false

==양쪽 값이 같으면 true

!=양쪽값이 다르면 true

-논리연산자(두 연산자 사이의 논리적인 관계를 정의하는 연산자)

&& And 두개의 값이 모두 참이면 true, 아니면 false

|| or 두개의 값 중 하나가 참이면 true, 아니면 false

-비트연산자(0과 1의 각 자리에 대한 연산을 수행하며, 0 또는 1의 결과값을 가짐)

&비트로 연산후 모두 참이면 true, 아니면 false

|비트로 연산후 하나가 참이면 true, 아니면 false

^비트로 연산후 서로 다르면 true, 아니면 false

-조건문

if

조건이 참인지 거짓인지에 따라 경로를 선택

case/switch

조건에 따라 여러 개의 경로 중 하나를 취하고자 할때 사용한다.

-반복문

while

수식이 거짓이 될 때 까지 해당 문장을 반복해서 실행

for

시작과 종료 조건을 지정하여 참인 동안 해당 문장을 반복하여 실행

do

초기값, 최종값, 중간값을 지정하여 반복 실행

■ 사용자 정의 자료형 활용

-열거체

괄호 안에 연속적인 값이 들어가는 자료형, 특정 값을 넣어주지 않으면 1씩 늘어나기 때문에 상수 배열이라고 부르기도 한다.

-구조체

괄호 안에 멤버 변수를 사용하는 자료형

-공용체

구조체와 거의 유사하나 조금 더 범주가 크다.

-추상화

복잡한 문제의 본질을 이해하기 위해 세부 사항은 배제하고 중요한 부분을 중심으로 간략화 하는 기법기능 추상화: 입력 자료를 출력 자료로 변환하는 과정자료 추상화: 자료와 자료에 적용할 수 있는 오퍼레이션을 함께 정의제어 추상화: 외부 이벤트에 대한 반응을 추상화 하는 방법

-상속

상위 수준 그룹의 모든 특성을 하위 수준 그룹이 이어받아 재사용/확장 (단일 상속과 다중 상속이 있다.)상위 수준의 그룹이 단하나만 존재할 때 이를 단일속성이라고 부른다.

-구체화

하위 수준 그룹이 상위 수준 그룹의 추상적인 부분을 구체화

■ 프로그래밍 언어의 유형 분류

개발 편의성에 따라 저급언어와 고급언어로 분류된다.

-저급 언어

기계가 이해할 수 있도록 만들어진 기계어 (예 : 어셈블리어)

-고급 언어

개발자가 소스코드를 작성할 때 쉽게 이해할 수 있도록 작성된 언어 (예 : C, C++, Java 등)

-명령형 언어/절차형 언어

명령어들이 순차적으로 실행되는 프로그래밍 언어 (예 : FORTRAN, COBOL, PASCAL,C)

-함수형 언어

수학적 수식과 같은 함수들로 프로그래밍을 구성해 호출 (예 : LISP)

-논리형 언어

규칙에 대한 활성화 조건이 만족되면 연관된 규칙이 실행 (예 : PROLOG)

-객체지향 언어

객체 간의 메시지 통신을 이용해 프로그래밍 (예 : JAVA, C++)

-컴파일 언어

고급언어 -> 기계어로 번역, 실행 속도가 높은 장점 (예 : FORTRAN, PASCAL, C, C++)

-인터프리터 언어

고급언어 명령문을 하나씩 번역하고 실행하는 방식의 언어, 프로그램 실행시 계산 (예 : BASIC, PROLOG, LISP, SNOBOL)

-혼합형 언어

고급어를 컴파일 하여 중간언어로 변환한 후 인터프리터에 의해 번역을 실행하는 방식의 언어 (예 : JAVA)

■ 프로그래밍 언어의 역사

-FORTRAN

1954~1957년 사이에 IBM의 존 배커스(John Backus)가 중심이 되어 만든 과학 응용 목적 언어최초의 고급 언어 반복문, 배열, 분기 개념을 처음으로 도입.

-COBOL

1959~1960년 사이에 미 국방성에서 해군그레이스 호퍼(Grace Hopper)가 이끈 팀에 의해서은행이나 대규모 회사들의 사무처리 목적으로 만든 언어

-PASCAL

1971년 개발되어 1980년대 말까지 많이 사용된 교육적인 목적의 구조적 프로그래밍 언어. 작고 간결하고 효율적, 분리 컴파일 기능, 문자열 조작, 입출력 기능의 효율화 등 고기능성을 갖춘.

-C

1972년 벨연구소의 데니스 리치(Dennis Ritchie)에 의해 개발UNIX 운영 체제 구현에 사용되는 언어.

-C++

1980년부터 비야네 스트루스트룹이(Bjarne Stroustrup)이 개발C언어를 확장하여 객체지향프로그래밍 개념을 도입.

-JAVA

객체지향 개념에 기반을 두지만 C++의 많은 구조와 복잡성을 포기,구조가 간결하고 단순하다.JVM(자바가상머신) 위에서 동작하기 때문에 다양한 응용분야에 적합하고. 이식성이 좋은 언어로 평가되고 있다.

-JAVASCRIPT

1995년에 개발된 객체 지향 스크립트 언어, 웹 페이지 동작을 구현한다.(HTML 에 함께 작성되면서 소스코드가 외부로 공개되며, 이 과정에서 보안 취약점이 발생한다.)

-PERL

텍스트 처리용 인터프리터 언어 (CGI용으로 주로 사용, 변수 명시적 선언 필요 X)

-PYTHON

배우기 쉽고 이식성이 좋다. 최근 트렌드 다양한 함수를 제공, 스타트업 및 글로벌 기업에서 많이 사용 인터프리터 언어이며 객체 지향, 스크립트 언어

-C#

00년 마이크로소프트는 .NET 의 주력 언어인 C#을 개발(C++과 JAVA의 비주얼 베이직의 장점을 합쳐서 만든 언어)

-DART

2011년 구글이 javascript를 대체하기 위해 개발한 언어 (java랑 유사한 문법)

-SWIFT

2014년 애플이 자사 앱 개발을 하기 위해 개발한 프로그래밍 언어 (빠르고, 현대적이고, 안전하다)

■ 절차지향 & 객체지향 프로그래밍

-절차지향 프로그래밍

객체라는 개념이 등장하기 이전에 모듈, 변수, 함수를 사용하여 개발

프로그램을 순차적으로 수행시키는 방법

자료구조와 명령 중심으로 프로그램을 구성

-객체지향 프로그래밍

객체와 객체 간의 통신을 통해 프로그램이 구현됨

- 객체지향 프로그래밍의 구성요소 3가지 -

객체(Object)

개체, 속성, 메소드로 구성된 클래스의 인스턴스를 의미1. 개체(Entity) : 현실 세계에 보이는 본질2. 속성

(Attribute) : 자료 저장소 역할, 절차지향에서는 변수와 대응3. 메소드(Method) : 호출단위 의미, 절차지

향프로그래밍 함수와 대응

클래스(Class)

객체의 타입을 정의하고 객체를 생성하는 틀

메시지(Message)

객체 간의 통신을 의미

■ 라이브러리 란?

-라이브러리

필요할 때 찾아서 사용할수 있도록 모듈화 되어 제공되는 프로그램(구성 : 도움말, 설치파일, 샘플코드)

-표준 라이브러리

프로그래밍 언어가 기본적으로 가지고 있는 라이브러리

-외부 라이브러리

별도의 파일을 설치해야 함, 누구나 개발하여 설치,공유 가능

-모듈

개별 파일에서 기능을 제공

-패키지

여러 개의 모듈을 한 개의 폴더에 묶어서 제공