

## 1.컴퓨터 언어 종류

### 절차지향언어 (대표적인 언어 : C언어)

절차지향 프로그래밍에 쓰이는 언어이다.

**절차 위주**, **개체를 순차적으로 처리**하여 프로그램이 유기적으로 연결되어야 한다.

명령형 프로그래밍과 동의어로 쓰이기도 하지만, 프로시저 호출의 개념을 바탕으로 하고 있는 프로그래밍 패러다임을 의미하기도 한다.

#### 장점

컴퓨터의 처리구조와 유사해 **실행속도가 빠름**

#### 단점

**유지보수가 어려움**

**실행 순서가 정해져 있으므로** 코드의 순서가 바뀌면 동일한 결과를 보장하기 어려움

**디버깅이 어려움** (디버깅 : 컴퓨터 프로그램 개발 단계 중에 발생하는 시스템의 논리적인 오류나 비정상적 연산(버그)을 찾아내고 그 원인을 밝히고 수정하는 작업 과정)

### 객체지향언어 (대표적인 언어 : Java, Python, C# 등)

프로그램을 **다수의 객체**로 만들고 **서로 상호작용할 수 있도록** 하는 프로그래밍 언어이다.

컴퓨터 프로그램을 명령어의 목록으로 보는 시각에서 벗어나 여러 개의 독립된 단위,

즉 **객체**들의 모임으로 파악하고자 하는 것이다.

(객체 : 구체적 사물, 사람, 추상적 개념 등)

#### 장점

코드의 **재활용성이 높음**

코딩이 절차지향보다 간편함

**디버깅이 쉬움** (디버깅 : 컴퓨터 프로그램 개발 단계 중에 발생하는 시스템의 논리적인 오류나 비정상적 연산(버그)을 찾아내고 그 원인을 밝히고 수정하는 작업 과정)

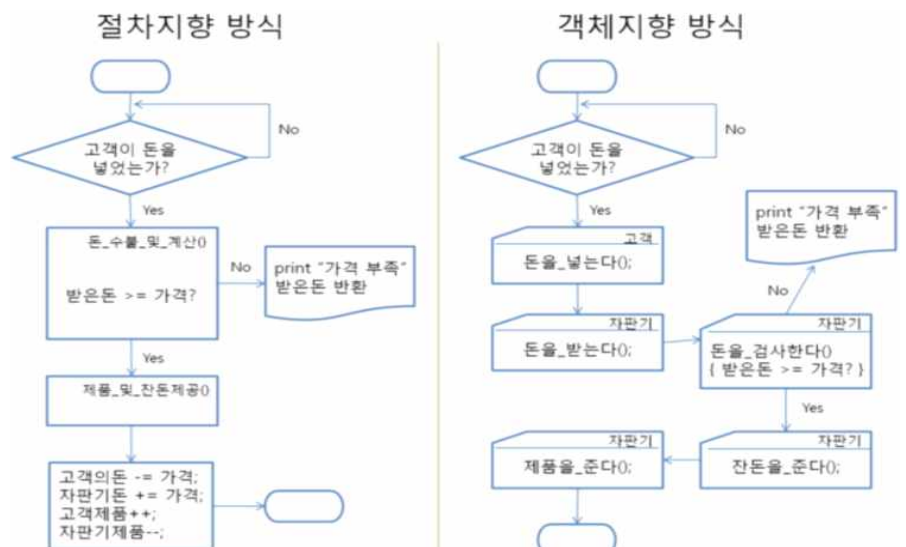
#### 단점

**처리속도가** 절차지향보다 **느림**

설계에 많은 시간소요가 들어감

### 차이점

절차지향은 **데이터 중심**, 객체지향은 **기능 중심**으로 메서드를 구현한다.



## 07.06 노트정리

### 언어

#### 절차지향언어

##### 정의

- 절차지향 프로그래밍에 쓰이는 언어이다.
- 절차 위주, 개체를 순차적으로 처리하여 프로그램이 유기적으로 연결되어야 한다.

##### 예시

c언어

##### 장점

- 컴퓨터의 처리구조와 유사해 실행속도가 빠름

##### 단점

- 유지보수가 어려움
- 실행 순서가 정해져 있으므로 코드의 순서가 바뀌면 동일한 결과를 보장하기 어려움
- 디버깅이 어려움

#### 객체지향언어

##### 정의

- 프로그램을 다수의 객체로 만들고 서로 상호작용할 수 있도록 하는 프로그래밍 언어이다.

##### 예시

Java, Python, C# 등

##### 장점

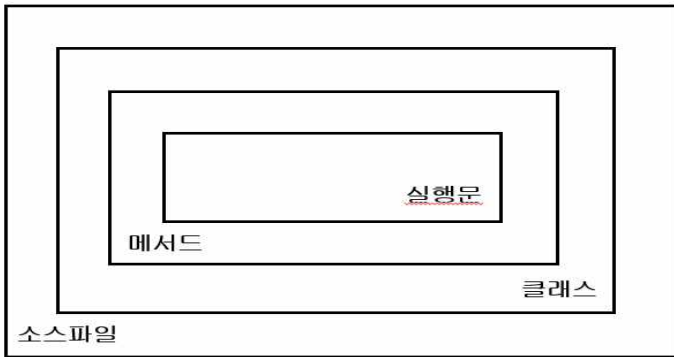
- 코드의 재사용성이 높음
- 코딩이 절차지향보다 간편함
- 디버깅이 쉬움

##### 단점

- 처리속도가 절차지향보다 느림
- 설계에 많은 시간소요가 들어감

## 2. 자바 프로그래밍의 구조

소스파일->클래스->메소드->실행문 순의 구조로 되어있다.



### 소스파일

확장명 .java 이다.

`public class` 이름과 같아야 한다.

자바는 중괄호( { } ) 블록 형태이다.

### 클래스

프로그램을 개발하는 단위이다.

유사한 특징을 지닌 객체들의 속성을 묶어 놓은 집합체이다.

적어도 하나의 클래스가 있어야 한다.

클래스 이름은 대문자로 시작하는 것이 관례이다.

### 메서드

수행할 작업을 나열한 코드의 모음이다.

자바 어플리케이션은 main으로 실행한다.

### 실행문

작업을 지시하는 변수 선언, 값 저장, 메소드 호출 등을 실행하기 위해 적는 코드이다.

#### • 출력 문자

`System.out.print( );` -> `print` (이어서 출력)

`System.out.println( );` -> `println` (줄바꿈)

`System.out.printf( );` -> `printf`

- ( " ")는 문자열을 출력할 때 사용
- ( )는 변수를 출력할 때 사용

#### • 주석문

프로그램에 덧붙이는 설명이고 프로그램 실행에 관여 안한다.

##### 주석적용

// -> 행 주석 한줄 쓸 때 사용

/\* \*/-> 여러줄 쓸 때 사용

/\*\* \*/-> 자바의 문서를 만들 때 사용

## 07.06 노트정리

### • 변수

계속 변하는 값을 저장할 수 있는 공간이다.

#### 변수적용

```
|Int a; //변수 선언  
|a=5; //변수에 정수 할당
```

-> `int a=5; //변수 선언 및 할당` 로 줄일 수 있다.

#### 변수이름 형태

카멜 : 단어와 단어 사이에 **대문자**로 구분한다. ex)string**A**rray

스네이크 : 단어+단어->**단어\_단어**로 표시한다. ex)string\_**a**rray

### • 상수

변하지 않는 **항상 같은 값**을 가지는 수

#### 변수적용

```
|final int b; //상수 선언  
|b = 20; //상수 할당
```

-> `final int b = 20; // 상수 선언 및 할당` 로 줄일 수 있다.

### 3. 자료형

변수를 선언할 때 변수명 앞에 붙는 것이 바로 변수의 데이터 타입이고 값과 값을 다룰 수 있는 연산의 집합이다.

#### 정수형

기본적으로 int타입을 사용한다.

같은 정수를 저장할 수 있지만, 값의 범위에 따라 byte, short, long 사용한다.

타입	범위	기본값
byte	-128 ~ 128	0
short	-32,768 ~ +32,767	0
int	-2,147,483,648 ~ +2,147,483,647	0
long	-9,223,372,036,854,775,808 ~ -9,223,372,036,854,775,807	0

long(수가 크다고 하면 이거 씀, 정수형은 L이나 l 접미사 붙여 표현)

#### 실수형

기본적으로 double타입을 사용한다.

값의 범위에 따라 float 타입 사용하기도 하고 실수형은 F나 f 접미사를 붙여 표현한다.

타입	범위	기본값
float	$1.4 \times 10^{-45} \sim 3.4 \times 10^{38}$ (양수)	0.0
double	$4.9 \times 10^{-324} \sim 1.8 \times 10^{308}$ (양수)	0.0

#### 문자형

문자값을 가질 수 있는 자료형이다.

작은따옴표(")로 하나의 문자 감싸서 표현하고 문자 여러 개가 아닌 1개만 저장할 수 있다.

타입	범위	기본값
char	₩u0000 ~ ₩uFFFF	₩u0000

일종의 정수 타입이기에 숫자 대입 가능, 유니코드 값 대입 가능, 문자열 불가능하다.

• String 타입 문자열 (문자와 문자열은 다르다!!)

문자형 char과는 달리, 문자열은 기본형이 아니다.

문자들의 집합이고 큰 따옴표(")로 여러 개의 문자를 감싼 문자열이다.

#### 논리형

논리 값은 boolean 타입 사용한다. 논리형은 두 가지 상태 참(True) / 거짓(False)로 표현한다.

타입	범위	기본값
boolean	true / false	false

## 자료형 정리

종류/크기	1byte	2byte	4byte	8byte
논리형	Boolean			
문자형		Char		
정수형	Byte	Short	Int	Long
실수형			Float	double

## 4. 사칙연산

$a+b$  -> 더하기

$a-b$  -> 빼기

$a*b$  -> 곱하기

$a/b$  -> 나누기 (나누기하고 나온 몫)

$a\%b$  -> 나머지 (나누기하고 나온 나머지)

## 5. 자바 주의사항

1. ;을 꼭 붙인다.
2. 클래스 이름 첫글자는 무조건 대문자로 쓴다.
3. 변수 이름은 알아볼 수 있게 작성하는 것 중요!