

## 2-1. 컴퓨터 구조

- CPU: CPU는 크게 CU(Control Unit), Register, ALU(Arithmetic and logical unit) 3개로 구성되어 있다. CU는 CPU를 제어하는 역할, Register는 CPU에서 처리할 명령어를 저장하는 역할, ALU는 연산, 비교, 판단을 담당하는 역할이다.
- RAM: 현재 실행 중인 프로그램의 명령어와 데이터를 저장하는 공간이다.
- 보조기억장치: 실행 중이지 않은 프로그램이 저장되어 있는 위치이다.

## 2-2. 변수

- 변수: 프로그래밍에서 값을 저장하는 공간을 뜻한다. 변수는 변수 이름, 타입, 값으로 구성된다.
- 변수 이름 규칙
  - ① 특수문자는 '\$', '\_'만 사용 가능
  - ② 동일한 이름의 변수 사용 불가
  - ③ 변수 첫 글자로 숫자 사용 불가
  - ④ 대소문자 구분
  - ⑤ 예약어 사용 불가
- 변수 선언: 변수 타입과 변수 이름을 작성하여 데이터 타입과 이름을 결정한다.
- 변수의 초기화: 변수를 사용하기 위해 변수에 값을 할당(최초 값 할당)한다.
- (지역)변수 범위: 변수는 중괄호{} 내에서 생성되었을 때 중괄호 내에서만 사용 가능하다. 밖에서 변수 선언 후 밑에 중괄호를 생성하였다면 중괄호 내에서도 사용 가능하다.

## 2-3. 변수의 타입(기본 타입)

- 변수의 타입: 변수의 크기 및 성질을 나타내고 크게 기본형과 참조형으로 나눌 수 있다.
- 기본형 변수: 실제 값이 저장되어 있는 변수이며 크게 정수, 실수, 논리 타입으로 나뉜다.
- 정수 타입: 정수 타입 변수로는 byte, char, short, int, long이 있다.
- 실수 타입: 실수 타입 변수로는 float, double이 있다.
- 논리 타입: 논리 타입 변수는 boolean 한 가지이다.

## 2-4. 타입 변환(Type Conversion)

- 타입 변환: 데이터의 타입을 다른 데이터 타입으로 변환하는 것을 말한다.
- 자동(묵시적) 타입 변환: 자동으로 타입 변환되는 것을 말하며, 작은 타입에서 큰 타입으로만 가능한 타입 변환이다.

#### 4-1. 조건문

- if 문: if 문의 소괄호 ( ) 안의 논리값에 따라 중괄호 { } 안의 내용을 실행할지 결정하는 조건식
- if-else 문: if 문에서 else 문이 추가된 형식이며, if 문의 조건이 false일 때 else 문을 실행한다.
- if-else if-else 문: if-else 문 중간에 else if 문이 추가된 형식이며, if 문의 조건이 false일 때 else if 문의 조건을 검사하고, else if 문의 조건이 false일 때 else 문을 실행한다.
- switch 문: if 문과 같이 조건문이지만, if 문과 다르게 인자로 boolean 외 타입의 값도 받을 수 있고, 하나의 블록 내에서 여러 조건을 설정할 수 있다.

#### 4-2. 반복문

- for 문: 코드 반복 실행을 위한 제어문으로서, 초기식, 조건식, 증감식으로 이루어져 있다.
- while 문: for 문과 마찬가지로 코드 반복 실행을 위한 제어문이고, for 문이 반복 횟수에 따라 조건식을 정한다면 while 문은 실행 및 종료시키고자 하는 조건에 따라 조건식을 정한다.
- do-while 문: while 문과 다르게 먼저 코드를 실행 후 조건식을 판단한다.

### 5-1. 참조 타입

- 참조 타입: 주소 값을 참조하는 타입을 뜻하며 배열, 열거, 클래스, 인터페이스 등이 있다.

### 5-2. 배열

- 배열: 같은 타입의 데이터가 연속적으로 나열된 자료구조이다.
- 배열의 초기화: 기본 타입과 마찬가지로 선언 및 초기화하여 사용한다.
  - 타입명[] 변수명 = new 타입명[배열 길이]
- Arrays: java.util 패키지에 포함된 클래스로 배열을 다루기 위한 여러 메소드를 가진 클래스이다.
- n차원 배열( $n \geq 2$ ): 1차원 이상의 배열을 뜻하며, 배열의 요소는  $n-1$ 차원의 배열이다.

### 6-1. 객체 지향 프로그램

- 클래스: 객체(클래스의 인스턴스)를 만들기 위한 설계도
- 객체: 클래스(설계도)를 통해 만들어진 결과물
- 인스턴스: 클래스에 의해 만들어진 객체
- 멤버: 객체를 이루는 구성 요소
- 메소드: 특정 작업을 수행하기 위한 명령어의 집합
- 재귀 함수: 자기 자신을 호출하는 함수

### 6-2. 오버로딩과 생성자

- 오버로딩: 메소드의 이름이 같지만, 매개 변수의 타입, 개수, 순서가 다른 메소드
- 생성자: 클래스로부터 인스턴스를 생성할 때 호출되는 멤버들을 초기화하는 메소드

### 8-1. 추상 클래스

- 추상 메소드: 메소드 앞에 'abstract' 키워드를 붙인 메소드로 선언부만 있고, 구현은 되어 있지 않은 메소드이다.
- 추상 클래스: 추상 메소드가 선언된 클래스이다.

### 8-2. 인터페이스

- 인터페이스: '클래스 및 메소드의 가이드라인'이며, 객체 및 메소드를 호출 및 사용하는 코드와 호출되는 객체 사이의 접점 역할을 한다.
- 인터페이스의 다형성: 인터페이스도 클래스, 추상 클래스와 마찬가지로, 인터페이스 타입 참조 변수로 인터페이스를 구현한 클래스 인스턴스를 참조할 수 있다. 이를 통해, 인터페이스로 다형성을 구현할 수 있다.