

1주차 마무리

월요일

- 자바의 장점
 - 운영체제에 관계 없이 동일 코드로 동작하여 이식성이 높다
 - 객체 지향성 프로그래밍이다 (object oriented programming)
 - 사용성이 쉽다
 - 자동 메모리관리 기능(Garbage Collection)
- JAVA Development Kit은 최상위 다이어그램이다.
- 벽을 느끼는 타임
 - 조건 / 반복 / 분기문(break/continue)
 - 배열 / 다차원 배열
 - 객체 지향 언어 - OOP
 - =====
 - JAVA API
 - Collection
 - Exception
 - IO(입출력) -> DB입출력을 위한 준비
 - Thread/Network

화요일

- 변수의 이해
 - 데이터를 변수에 저장 시킨다
 - 저장 시킨 변수를 메모리에 적재 시킨다
 - 변수에 기록된 값은 지속적으로 재 활용 가능하다.
- 입력 버퍼
 - 비트 2진수 값 단 한 개를 저장 (0 / 1) 최소 저장 단위
 - 비트 8개가 모여 만들어지면 하나의 바이트가 만들어진다
 - 하나의 바이트가 데이터를 처리한다 (-256~256)
- 변수 선언
 - **boolean** 논리형 참/거짓 만 저장 가능하다
 - **char** 문자'형'
 - **String** 문자'열'
- 기본 자료형 (7개)
 - byte (1)
 - short (2)
 - int (4byte 크기)
 - long (8)
 - 실수형 (정수가 아닌 유리수 0.123~~)
 - float (4)
 - double (8byte)
- 변수 명명 규칙
 - #카멜 표기법만 쓰자(이어지는 단어의 첫 글자만 대문자)
 - 숫자로 시작하면 안된다.
- 자료형 변경
 - 자동 형 변환
 - byte -> char(short) -> int > long->float->double 순으로
 - 컴파일러가 자동으로 변환 시켜 준다
- 강제 형 변환
 - (int)변수명 같이 선언하여 사용한다.
 - 불필요한 정보를 잘라낸다.

수요일

- 강제 형 변환
 - (int)변수명 같이 선언하여 사용한다.
 - 불필요한 정보를 잘라낸다.
 - 큰 데이터에서 작은 데이터로 변경시 손실 확률 높음
- 변수와 메모리 구조

- Static
- HEAP
- STACK
- 출력문 -
 - System.out.println(); -- 한 줄 출력 (줄 바꿈 효과가 일어난다.)
 - System.out.print(); -- 단순 출력.
 - System.out.printf() -- 포맷 출력
 - printf 의 활용
 - **%d 정수형 (데시머리?10진수라는 의미)**
 - `-%5d` : 5 칸을 확보하고 오른쪽 정렬.
 - `-%-5d` : 5 칸을 확보하고 왼쪽 정렬.
 - **%c (char) -문자**
 - **%f (float) - 실수 (실수들은 무조건 %f)**
 - `-%2f` : 소수점 아래 2 자리 까지만 표시
 - **%s 문자열**
 - **%b 논리형.**
- 탈출문 (escape)
- scanner
 - 스캐너 생성하고 입력 받기
 - ex) int input = sc.nextInt();
- \ % 등의 기호는 연달아 두 번 사용해 출력 가능하다.
 - 증감 연산자 (if / for 원톨 연산자)

- 전위 연산
- int ++num (다른 연산보다 먼저 실행됨.)

- 후위 연산
- int num++ (다른 연산 후 연산함.)
 - int a = 3;
 - int b = 5;
 - int c = a++ + --b;
 - 최종적으로 : a=4 b=4 c=7

- 비교 연산자
- 논리 연산자

- **&& == AND (TTA)**
- **|| == OR(FFF)**

목요일

- IF문
 - if문 안에 한번 if을 해서 중첩하여 사용한다.
- switch문
 -

금요일

- 반복문
- Switch문
 - 들어온 값에 맞는 case로 이동한다.
 - case에는 int형이 주로 쓰인다
 - 실수형은 사용불가.
 - 작성법
 - switch(식){
 -
 - case 결과 값 1 : 수행 코드1; break;
 - case 결과 값 2 : 수행 코드2; break;
 - case 결과 값 3 : 수행 코드3; break;
 -원하는 만큼 case생성 가능
 - default : 모든 case를 만족하지 못할 경우 수행하는 코드.
 - }

- 반복문
 - **for**
 - 목표가 정해진 경우
 - ex) 운동장 10바퀴
 - **do while**
 - 목표가 정확하지 않을 경우
 - ex) 운동장 지쳐 쓰러질 때 까지.
- 변수의 생명 주기
 - 선언된 범위에 따라 사용 가능 범위가 달라진다