

# 코테 준비

## 4장, 기본 자료형

# I. 체크 목록

- 기본 자료형의 크기, 구간, 부호 여부, 연산자에 익숙한가?
- Primitive와 Boxed Primitive(Wrapper class)의 차이를 아는가? Auto-Boxing, Unboxing이 무엇인가?
- java.lang.Math 클래스에서 제공하는 수학과 관련한 핵심 정적 메소드들이 익숙한가?

`Math.abs(), ceil(), floor(), min(), max(), pow(), sqrt(), random()` 등

- Random 라이브러리의 핵심 메소드가 익숙한가?

`nextInt(16), nextInt(), nextBoolean(), nextDouble()` 등

- 비트 연산자를 활용한 비트 연산자를 잘 다루는 가?
- Boxed Primitive의 메소드들을 활용하여
  - 최댓값, 최솟값을 다룰 수 있는가
  - Boxed Primitive의 인스턴스를 생성할 수 있는가
- 정수, 문자, 문자열간 변환 방법에 익숙한가?

## II. 학습 문제 복습과 구현 테스트

문제 4.1 – 패리티 계산하기

문제 4.2 – 비트 스왑하기

문제 4.3 – 비트 뒤집기

문제 4.4 – 같은 무게를 갖는 가장 가까운 양의 정수 찾기

## 문제 4.1 – 패리티 계산하기

Q1. 패리티(Parity) 란 무엇인가요?

Q2. 64비트로 이루어진 이진수에 대해, 패리티 계산하는 알고리즘을 구현해보세요.

Q3. Naïve 한 방식으로 구현해보기, 그리고 시간 복잡도는?

Q4. 더 향상시킬 순 없을까요?

Q5. lookup table을 사용하여 효율적으로 작성할 수도 있지 않을까요?

Q6. SHIFT 연산, XOR 연산의 어떤 특징을 이용하면,  
여러 비트 계산을 한번에 수행할 수도 있을 것 같아요. 설명해주세요. 이 때 시간 복잡도는?

## 문제 4.2 – 비트 스왑 하기

Q1. 64비트 정수가 주어졌을 때,  $i$ 번째 비트와  $j$ 번째 비트를 swap하는 알고리즘을 설계해보세요.

Q2. 효율적으로 구현할 수 있는 부분이 있을까요? 있다면 설명해주세요.

### 문제 4.3 – 비트 뒤집기(reverse)

Q1. 64비트의 정수형 값의 이진수 표현에 대하여, 역순으로 재구성하는 알고리즘을 설계해주세요.

Q2. 더 효율적으로 바꿀 수 있는 부분이 있을까요? 있다면 설명해주세요.