

# 자료구조와 코딩 테스트

충남대학교 데이터네트워크연구실  
이영석

# 코딩 테스트: 코딩 문제 해결 능력 점검

- IT/SW 기업 서류 전형+ 코딩 테스트 진행

- 카카오, Naver, NHN, SK, KT, 삼성 등

- 제한시간내 해결: 자료구조 활용

- 스택, 큐, 이진탐색, 트리, 그래프

- 코드 효율: 알고리즘 성능/복잡도

- 메모리 효율성: 배열 vs. 리스트, 변수 타입의 최대 크기

- 시간 효율성: 동적프로그래밍, 자료구조(해시), 알고리즘(병합정렬)



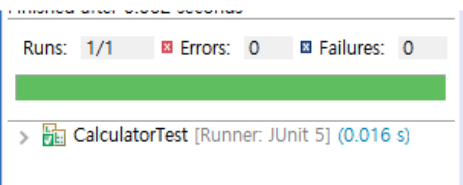
# 자료 구조와 코딩 테스트

- 자료구조 (Data structure): 데이터 구성, 관리, 저장에 관한 학습
  - 추상 데이터 타입 (Abstract Data Type), Java Interface, Collection Framework
- 코딩 테스트: 문제 해결 능력 평가
  - 문제 정의: 문제에 맞는 자료구조 활용(배열, 리스트, 해시, 셋, 맵 등)
  - 최적의 해결 방법: 메모리 요구사항(자료구조), 시간제한 요구사항(자료구조, 알고리즘)
- 코딩 테스트 문제를 풀어보며 자료 구조 학습
  - 온라인 코딩 테스트 시스템 활용 (ex. 프로그래머스, 백준, SW역량, LeetCode, HackerRank 등)

# 오프라인 코딩테스트

- 유닛 테스트(unit testing)
  - JUnit은 꼭 쓰자
  - assertEquals(예상 값, 실제 값)

```
1 package calculator;  
2  
3 public class Calculator {  
4     public int sum(int num1, int num2) {  
5         return num1 + num2;  
6     }  
7 }  
8
```



```
1 package calculatorTest;  
2  
3 import static org.junit.jupiter.api.Assertions.*;  
4  
5  
6  
7 class CalculatorTest {  
8  
9     @Test  
10    void test() {  
11        //fail("Not yet implemented");  
12        calculator.Calculator calculator = new calculator.Calculator();  
13        assertEquals(30, calculator.sum(10, 20));  
14    }  
15  
16 }
```

- Assert

- `assertArrayEquals(a,b)` : 배열 a와b가 일치함을 확인
- `assertEquals(a,b)` : 객체 a와b의 값이 같은지 확인
- `assertSame(a,b)` : 객체 a와b가 같은 객체임을 확인
- `assertTrue(a)` : a가 참인지 확인
- `assertNotNull(a)` : a객체가 null이 아님을 확인

# 자료구조, 코딩테스트와 자바 표준 라이브러리

- 자료구조 기초

- 배열 (Array)
- 연결 구조 (Linked List)
- 스택 (Stack)
- 큐 (Queue)
- 셋 (Set)
- 맵 (Map)

- 자료구조 핵심

- 추상 데이터 타입, STL
- 서로소 집합 (Disjoint set)
- 우선 순위 큐 (Priority Queue)
- 힙 (Heap)
- 정렬 (Sort)
- 문자열 (String)
- 트리 (Tree)
- 그래프 (Graph)

# 수업 방법) 코딩 테스트

- 기초 지식 학습 + 코딩테스트
  - 이론강의(+ 코딩테스트) + 실습강의(코딩테스트)
- 일정 시간 (기본 15분) 동안 각자 해결 및 답안 제출, 채점
- 예제 답안 or 제출된 답안을 함께 살펴보기

# ID / Password 확인

- 수업용 ID / Password 공지 !
  - <http://e-learn.cnu.ac.kr> 확인
- 코딩 테스트 사이트: <http://coding.cnu.ac.kr> 접속 및 로그인 확인



# 코딩 테스트 연습

# 코딩 테스트 입력과 출력

- 코딩 테스트에서는 주어진 입력을 읽고 정확한 출력!
  - 테스트 SW가 정답과 제출답안을 기계적으로 문자열 일치하는지 검사!
- ex) Hello, World! 를 출력하시오.
  - > 'hello, World!' (X)
  - > 'Hello, World!' (O)
- 표준 입력 / 표준 출력을 잘 활용해야 함
  - `system.out.println`, `system.in`

# 표준 입력 처리

- 자주 사용하는 Scanner 멤버 함수 기억
  - 참고) <https://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html>
  - nextInt, nextDouble 등의 멤버 함수가 개행 (\n)을 어떻게 처리하는지 잘 살펴볼 것
- String.Split(), Integer.parseInt(), Double.parseDouble() 잘 활용해야 함!
  - 주어진 입력을 적절한 자료형의 데이터로 변형

# 연습 문제) 문자열 반복 출력

- 주어진 문자열을 지정한 숫자만큼 출력하십시오.

[입력]

3 # 문자열 개수

4 2 3 # 각 문자열에 대한 반복 출력 개수

Hello, World!

Data Structure

Code test

[출력]

Hello, World!

Hello, World!

Hello, World!

Hello, World!

Data Structure

Data Structure

Code test

Code test

Code test

# 연습 문제 예제 답안) nextLine, Split, parseInt 활용

- 이것만 답안이 아님! 자신이 편한 표준 입력 처리 방법 익힐 것!
  - 특히 공백, 개행 문자 다루는 방법 미리 익혀두어야 함!

```
3
4 2 3
Hello, World!
Data Structure
Code test
harny@LuHa-X1C6 ~/Github
n.0 | java StringPrint
Hello, World!
Hello, World!
Hello, World!
Hello, World!
Data Structure
Data Structure
Code test
Code test
Code test
```

```
1 import java.util.Scanner;
2
3 class StringPrint {
4
5     private static final Scanner scanner = new Scanner(System.in);
6
7     public static void main(String[] args) {
8         String line = scanner.nextLine();
9         int str_cnt = Integer.parseInt(line);
10
11         line = scanner.nextLine();
12         String[] repeat = line.split(" ");
13
14         for (int i = 0; i < str_cnt; i++) {
15             int repeat_cnt = Integer.parseInt(repeat[i]);
16             line = scanner.nextLine();
17             for (int j = 0; j < repeat_cnt; j++) {
18                 System.out.println(line);
19             }
20         }
21     }
22 }
```