

## 4-5장 과제

---

- ▶ 다음 프로그램을 강의지원시스템에 5/16(일) 23:00까지 제출
  - ▶ 클래스 별로 적절하게 파일 분리하지 않으면 크게 감점
  - ▶ 각 과제별 폴더에 source code만 저장해서 파일 하나로 압축하여 제출
  - ▶ string class이외 자료구조 class는 사용 금지
- ▶ 트리 출력
  - ▶ tree를 보기 좋게 출력하는 방법은 검색을 통해 다양하게 찾을 수 있으니 원하는 방식대로 출력할 것.
    - ▶ 인터넷이나 선배들의 코드를 참조한 경우에는 comment로 어디서 참조했는지 반드시 기술할 것
    - ▶ 본인이 직접 구현한 경우에는 직접 구현했음을 기술할 것
      - 뒤 예제의 방식은 추가적인 postorder traversal을 통해, 현재 노드가 얼마크기의 width에 출력되어야 할지를 미리 결정하여 노드의 멤버 변수에 저장하여 사용
- ▶ 2-3번 과제는 레퍼런스 방식으로 child node를 연결할 것
- ▶ 도전과제: 공통 generic과 상속을 사용하여 구현해 볼 것

# 1. Complete Binary Tree

- ▶ 노드 개수를 입력받아 complete binary tree를 구성하여 출력하고, preorder, inorder, postorder, level order로 traversal한 결과 출력
  - ▶ tree representation에서 배열 구현 방식만 사용하시오.
  - ▶ 수업시간에 다룬 recursive 함수의 형태로 반드시 구현하시오.
    - ▶ void preorder(int nodeNum) / int getLeft(int nodeNum) 등 으로 구성
  - ▶ level order에서는 반드시 최초 크기가 4인 circular queue를 사용

노드 개수를 입력하세요 : 10

```

          1
        +-----+
       2       3
    +-----+ +---+
   4       5 6 7
 +---+ +---+
8 9 10 x
preorder   : 1 2 4 8 9 5 10 3 6 7
inorder    : 8 4 9 2 10 5 1 6 3 7
postorder  : 8 9 4 10 5 2 6 7 3 1
level order: 1 2 3 4 5 6 7 8 9 10
```

## 2. 계산식

---

- ▶ 3장의 계산기 과제를 Tree로 사용하여 해결하시오.
  - ▶ 입력받은 수식을 postfix notation으로 바꾸면, 이를 tree로 변환할 수 있음.
    - ▶ 계산 과정과 유사하게, postfix의 각 피연산자를 노드에 저장해서 stack에 넣고, 연산자가 나타나면 stack에서 두 개의 노드를 pop하여 subtree를 구성하고, 해당 root를 다시 stack에 push.
    - ▶ 이 과정을 반복하면 마지막에는 stack에 전체 계산식의 root가 남게 됨.
  - ▶ 적절한 class로 tree를 구성하여, 다음 페이지 예제와 같이 연산식에 대한 표현과 계산 과정을 tree에서 모두 표현 하시오.
  - ▶ 2-3번 과제는 Node leftChild, right child를 가지는 방식으로 구현하시오.
    - ▶ 앞페이지 과제에서 배열로 구현한 출력 method를 레퍼런스 형 tree에 맞게 수정.
  - ▶ 만들어진 tree를 preorder travers하여, 입력된 중위표기에 대한 전위표기 계산식을 tree 출력 전에 먼저 출력하시오.

## 2. 계산식

계산식을 입력하세요 (종료는 quit): 13 >> 2 & 1 + - - 9

--- 후위 계산식은 : 13 2 >> 1 9 m m + &

--- 전위 계산식은 : & >> 13 2 + 1 m m 9

```

                [2] &
      +-----+
[3] >>          [10] +
+-----+      +-----+
13   2         1   [9] m
                +-----+
                [-9] m   x
                +-----+
                9     x
```

---- 결과는 2

### 3. binary search tree

- ▶ 입력 문자열에 대해서 BST를 구성하고 출력하시오.
  - ▶ 입력된 순서에 따라 각기 다른 트리가 나오는 점 확인 할 것
    - ▶ 입력 문자열을 한글 3글자로 한정하지 마시오
  - ▶ right subtree는 존재하는데 left subtree가 null인 경우에 대한 출력은 -x

문자열들을 입력하세요 (마지막은 quit) : 김시관 이해연 김병만 김선명 오득환

```
      김시관
    +-----+
   김병만   이해연
+-----+   +---+
x 김선명   오득환 x
```

문자열들을 입력하세요 (마지막은 quit) : 김병만 김선명 이현아

```
      김병만
    +-----+
   x 김선명
      +-----+
      x 이현아
```

문자열들을 입력하세요 (마지막은 quit) : quit

종료합니다.