

## 1. 문제를 효율적으로 해결하기 위한 방법

이번 실습에서는 객체지향 프로그래밍의 상속과 다형성 기능을 활용하여 연결리스트와 스택 구조를 구현하는 프로그램을 작성하는 것이 목표이다.

이를 위해 기능 구현에 필요한 클래스가 무엇인지, 각 클래스에서 필요한 변수, 멤버함수는 무엇인지를 정리하여 차례대로 구현한다.

## 2. 단계별 수행 내용 및 자료구조

### 1) LinkedList의 Node 클래스 구현

연결 리스트 자료구조를 제공하기 위한 Node 클래스이다.

변수를 저장하는 data 필드와 다음 노드를 가리키는 link필드로 구성되어 있다

- int 형 변수 data
  - node 포인터형 변수 link
  - 맨 처음 노드를 가리키는 first 변수 (protected)
  - 리스트 노드의 개수를 저장하는 변수 current\_size (protected)
  - **Insert 함수** : 리스트 맨 앞에 노드 삽입 (queue의 형태)
  - **Delete 함수** : 리스트 맨 뒤에 노드 삭제 (queue의 형태)
  - **Print 함수** : 현재 리스트의 모습 출력
- + ) 저장되는 데이터 변수에 대해 템플릿 자료형을 사용하도록 변경

### 2) Stack 클래스 구현

1에서 작성한 클래스를 상속하여 스택 자료구조를 제공하는 클래스를 구현한다.

- 스택은 맨 앞의 데이터부터 삭제처리를 한다 : **Delete 함수** 재정의 필요
- >> 이때 재정의하는 함수가 virtual로 선언되어야 한다.

### 3) 기타 기능 구현

- Stack 자료구조를 사용할 것인지, Linked List 자료구조를 사용할 것인지를 사용자로부터 입력 받아 수행한다.
- 입력 받은 자료구조에서 삽입, 삭제, 출력, 종료 중 어느 작업을 수행할 것인지에 대한 것도 사용자로부터 입력 받아 수행한다.
- 화면에 결과를 알아보기 쉽게 출력되도록 내용의 배치를 조정한다