

컴퓨터공학실험1 4주차 예비보고서

전공: 컴퓨터공학과

학년: 2

학번: 20191559

이름: 강상원

문제 1. 다음을 명확하게 설명하라

- 다형성

변수, 상수, 함수, 오브젝트 등의 요소가 자료형에 관계없이 작동되는 성질을 의미한다. 다형성은 크게 서브타입 다형성(subtype polymorphism), 변수 다형성(parametric polymorphism), 임시 다형성(ad hoc polymorphism)이 있다.

- 캡슐화

객체의 일부를 private 처리하여 구현 내용 일부를 외부에서 접촉할 수 없게 만드는 것이다. 데이터와 그 함수를 연계하여 묶는다. 캡슐화의 이점은 다른 코드에서 필요한 부분만 빼서 편이하게 쓸 수 있고, 객체 내부 데이터가 임의로 수정되는 것을 방지할 수 있다는 것이다.

- 재정의

사우이 클래스에서 상속받은 함수 등 메서드를 하위 클래스에서 재정의하는 것을 의미한다.

문제 2. CPP-2의 최종 테스트 코드에서 서브타입 다형성이 적용되는 부분을 명시하고 그 이유를 기술하시오. (4주차 강의 PPT 참고)

```
void prnMenu()
{
    cout<<"*****"<<endl;
    cout<<"* 1. 삽입    2. 삭제    3. 출력    4. 종료 *"<<endl;
    cout<<"*****"<<endl;
    cout<<endl;
    cout<<"원하시는 메뉴를 골라주세요: ";
}
```

```
int main()
{
    // 스택 및 연결 리스트 테스트용 코드
    int mode, selectNumber, tmpItem;
    LinkedList<int> *p;
    bool flag = false;

    cout<<"자료구조 선택(1: Stack, Other: Linked List): ";
    cin>>mode;

    // 기반 클래스의 포인터를 사용하여 기반 클래스 뿐만 아니라
    // 파생 클래스의 인스턴스 또한 접근할 수 있다.
    if(mode == 1)
        p = new Stack<int>;           // 정수를 저장하는 스택
    else
        p = new LinkedList<int>;      // 정수를 저장하는 연결 리스트
}
```

mode가 1일 경우 서브타입 다형성이 구현된다. LinkedList 클래스를 Stack 클래스가 상속받기 위해서 기반 클래스의 포인터에 파생 클래스 인스턴스의 주소를 저장할 수 있게 한다.

문제 3. 주어진 문제 및 3-1의 문제 해결에 관한 내용을 이해하고 이 문제를 효율적으로 해결하기 위한 방법을 생각하여 이를 1쪽 이내로 요약하여 제출하시오. 문제 해결을 위한 간단한 단계별 수행 내용, 자료구조 등을 기술하시오. (4주차 교재 PDF 참고)

Print 부분이 완벽히 구현되어 있지 않은 LinkedList 클래스를 완성하고, LinkedList 함수를 템플릿 자료형으로 쓸 수 있도록 바꿔야 한다.

또한 스택을 이해하고 LinkedList에서 상속받는 Stack 클래스를 만들어야 한다.

- Print()
Linked List의 원소를 순회하는 형태로 값을 순서대로 출력한다.
- LinkedList 함수를 템플릿 자료형으로 바꾸기
클래스 선언부에 템플릿 클래스를 추가하고 데이터를 담던 자료의 자료형을 모두 그 템플릿 변수로 바꿔 준다.
- LinkedList에서 상속받는 Stack 클래스를 만들기
Delete() 함수를 재정의하여야 한다.
삭제 성공-> TRUE, 삭제 실패-> FALSE (스택이 빈 경우)
first..는 first->link가 되며, current_size는 1 감소한다.