**2024 Spring OOP Assignment Report**

과제 번호 : Assignment #6

학번 : 49004728

이름 : 배재호

Povis ID : bjh1749

**명예서약 (Honor Code)**

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

I completed this programming task without the improper help of others.

1. **프로그램 개요**

이 프로그램은 논리학의 삼단 논법(syllogism)을 다중 헤드 리스트(MultiHeadList) 자료 구조로 구현하고 조작하기 위해 만들어졌습니다. 다중 헤드 리스트는 여러 개의 독립된 연결 리스트를 동시에 관리할 수 있는 자료 구조로, 각 리스트는 개별적으로 접근, 수정, 삭제가 가능합니다. 본 프로그램은 이러한 다중 헤드 리스트에 항목을 추가하거나 삭제하고, 특정 조건에 따라 리스트를 병합하는 기능을 제공합니다. 또한, 리스트 병합 시 각 리스트의 노드들을 특정 규칙에 따라 결합하여 하나의 리스트로 만드는 기능을 포함하고 있습니다. 이 프로그램은 논리적인 관계를 표현하는 데이터 쌍을 리스트에 추가, 병합, 삭제하고 논리적 연관관계에 대한 질의에 응답할 수 있는 기능을 제공한다.

1. **프로그램의 구조 및 알고리즘**

**2.1 MultiHeadList**

이 프로그램의 핵심 클래스는 MultiHeadList<T>입니다. 이 클래스는 여러 개의 독립된 연결 리스트를 저장하고 각각의 리스트를 위한 헤드 포인터를 벡터로 관리합니다. 클래스는 템플릿으로 구현되어 있어 다양한 데이터 타입을 저장할 수 있습니다. 리스트의 각 노드는 Node<T> 구조체로 정의되며, 이는 데이터 항목과 이전 및 다음 노드를 가리키는 포인터를 포함합니다. headList 벡터는 각 리스트의 헤드 포인터를 저장하는 벡터로, 리스트를 관리하는 핵심 데이터 구조입니다. 또한, 리스트를 순회하기 위한 Iterator 클래스와 역방향으로 순회하기 위한 ReverseIterator 클래스가 정의되어 있습니다. MultiHeadList 클래스의 주요 기능은 다음과 같습니다. push\_back 함수는 특정 리스트의 끝에 새로운 노드를 추가하며, push\_front 함수는 특정 리스트의 시작에 새로운 노드를 추가합니다. insert 함수는 지정된 위치에 새로운 노드를 삽입하고, pop\_back와 pop\_front 함수는 각각 특정 리스트의 끝과 시작에서 노드를 제거합니다. merge 함수는 두 리스트를 병합하며, 병합 후 뒷 리스트는 제거됩니다. erase 함수는 주어진 데이터를 가진 노드나 주어진 이터레이터가 가리키는 노드를 제거합니다.

**2.2 Syllogism**

Syllogism 클래스는 논리적 명제와 삼단 논법을 처리하기 위해 MultiHeadList를 활용하는 응용 프로그램의 핵심 클래스입니다. 이 클래스는 논리적 명제 쌍을 관리하고, 특정 조건에 따라 리스트를 병합하거나 질의를 수행하는 기능을 다음 세 메소드를 활용하여 제공합니다.

* put : 주어진 명제 쌍을 리스트에 추가합니다. 기존 리스트를 순회하여 첫 번째 값이 쌍의 두 번째 값과 일치하는 노드를 찾고, 두 번째 값이 쌍의 첫 번째 값과 일치하는 노드를 찾습니다. 두 노드를 찾으면 두 리스트를 병합하고, 그렇지 않으면 적절한 리스트에 노드를 추가하거나 새로운 리스트를 생성합다. 이를 위해 firstHeadIdx, firstFound와 secondHeadIdx, secondFound라는 로컬 변수를 선언해 활용합니다.
* qna : 주어진 값에 대해 리스트를 순회하며 주어진 값이 리스트에 포함된 노드를 찾고, 해당 값으로부터 도출되는 논리적 결론을 출력합니다.
* print : 현재 저장된 모든 리스트와 그 내용을 출력하여, 리스트의 상태를 시각적으로 확인할 수 있게 합니다.

1. **토론 및 개선**

프로그램에서 발생할 수 있는 다양한 에러 상황에 대한 처리가 미흡합다. 예를 들어, 이미 있는 논리관계를 입력하거나 아예 잘못된 포맷의 입력 등의 상황이 벌어지지 않을 것이라고 가정했다. 프로그램의 안정성을 높이기 위해서는 입력 검증 과정이 반드시 필요하다.

일부 MultiHeadList의 메소드의 경우 모든 head를 순회하도록 작성되어 있다. 적절한 flag나 control value가 있다면 불필요한 순회를 크게 줄일 수 있을 것 같은데 방법을 궁구하지 못했다.

1. **참고 문헌**

없음