4주차 결과보고서

전공: 경제학과 학년: 4학년 학번: 20180501 이름: 김연수

1.

LinkedList라는 기본 클래스를 사용해서, 파라미터적 다형성을 이용해 Stack 클래스를 구현할 것이다.

Stack 클래스는 LinkedList 클래스의 파생클래스로 선언한다.

* class Stack : public LinkedList<T>

위 코드로 Stack을 LinkedList의 파생클래스로 정의할 수 있다.

파라미터적 다형성을 만족시키기 위해서 다음과 같이 노드 클래스에 templete <typename T>를 적어준다. 노드 뿐만이 아니라 파라미터적 다형성을 사용하는 모든 클래스는 템플릿클래스로 구현해주도록 한다.

Node<T> \*newnode = new Node<T>(element);

노드 객체를 정의할 때는 Node만 적는 것이 아니라, 위와 같이 Node<T>라고 명시해줘야 한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

위는 Stack.h의 코드다. 다른 함수는 LinkedList와 똑같이 동작하지만, Delete함수는 다시 정의해줘야 한다. Delete를 하면 LIFO형식으로 작동하도록, 즉 제일 마지막에 들어온 element가 첫 번째로 삭제되도록 만들어줘야 한다.

* this->first가 0인 경우에 false를 리턴한다. 노드가 아무것도 없는 상태를 뜻한다. 여기서 this 포인터를 사용하는 이유는 first와 current\_size가 LinkedList의 멤버변수이기 때문이다.
* current에 가장 최근에 들어온 노드를 저장한다. 가장 최근에 들어온 element는 first에 있으므로 ‘current = this->first’로 가장 최근에 들어온 노드를 받아올 수 있다. 그리고나서는 현재 first를 다음 노드의 위치로 옮겨줘야한다. ‘this->first = this->first->link’로 이를 수행해줬다.
* 가장 마지막으로 들어온 노드의 data를 변수 element에 저장해준다.
* current를 할당해제한다.
* 노드를 한개 뺐으므로 current\_size에 -1을 해주고, 참을 리턴한다.