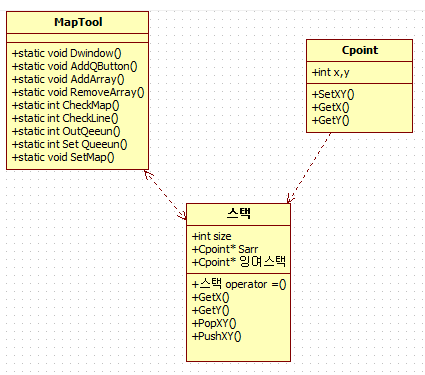
**N –퀸 문제 설계서**

20120302 김우진

**시스템 구성**



작업환경

C++사용과 windows기반의 프로그래밍.

**Class MapTool**

개요: 퀸의 배치와 공격범위를 갖는 배열인 Map을 관리하는 객체.

<설명>

라인단위로 퀸을 삽입, 삭제한다. 퀸을 삽입 할 시 입력 받은 라인에서 퀸이 배치 가능한 지점을 잉여스택에 역순 저장( line,size->line,size-1….->line,0 순으로 push )하고 가장 최근의 값을 꺼내서 Sarr(퀸의 위치정보를 갖는 스택)에 저장한다.

**Class Cpoint**

개요: 좌표를 저장, 스택클래스에서 배열 형태로 스택의 역할을 하게됨.

**Class 스택**

개요: 클래스내부에 2개의 스택과 pop ,push 함수가 내장되어 있다.

<설명>

현재 맵의 퀸의 위치를 저장는 Sarr와 현재 퀸을 pop 할 경우 퀸이 배치될 수 있는 예비 퀸의 좌표를 저장하는 잉여스택이 있다.

Sarr의 경우는 push나 pop과 동시에 맵을 update하여 새로운 퀸의 위치와 추가된 퀸에 의한 새로운 공격범위를 지정한다.

Push : push를 하기 전 행을 조사하여 퀸이 배치될 수 있는 곳을 잉여스택에 역순으로 push 한 후 가장 마지막 값을 꺼내어 push한다. (가령0번 라인에 퀸을 배치한다면, 잉여스택에 0,size=>0,size-1=>…=>0,0을 배치하고 0,0을 pop하여 Sarr에 push )

Pop: 잉여스택에서 잉여스택의 저장된 라인과 라인이 같이 않다면 같을 때 까지 전부 빼내고 같다면 빼고 잉여스택의 값을 집어넣는다.

**모듈설계**

**스택 -** 구성요소: 스택, Cpoint, MapTool 클래스

**<스택 내부 멤버 소개>**

GetX, GetY – 스택에 pop호출 시 빠져나오는 좌표의 x,y값을 반환. 호출 시 실제 스택에는 영향을 주지 않는다.

Push,pop : 두 스택의 push와 pop을 담당한다. 스택은 기본적으로 -1,-1로 초기화 되어 있으므로 -1,-1을 기준으로 뽑을 좌표와 집어넣을 배열의 위치를 구분 한다.

**<MapTool 내부 멤버 소개>**

**-버튼 관련함수**

static void DWindow(HWND\*\* p,int size): 버튼 배열 삭제 함수

static void AddQButton(HWND\*\*\* p,int size): 버튼 추가함수

static void RemoveQButton(HWND\*\*\* p,int size): 버튼 삭제 함수

static void RemoveArray(int\*\*\* p,int size):맵 추가함수

static void AddArray(int\*\*\* p,int size):맵 삭제함수

**-맵 관련함수**

static int CheckMap(int\*\* p,int size):맵에 퀸이 모두 있나 체크

static int CheckLine(int \*\*p,int line,int size):라인을 검사하는 함수

static void OutQeeun(int x,int size,int\*\* a,스택 A):퀸을 빼면서 pop호출

static int SetQeeun(int size,int\*\* a, 스택 A,int x, int y):퀸을 배치하며 push 호출

static void SetMap(스택 A,int\*\* a,int size):스택의 상태를 맵에 반영

**주요 알고리즘과 자료구조**

개요: 스택과 백트래킹 기법으로 퀸을 자동배치 한다.

요약:

두개의 스택을 이용하여 스택 1에는 현재 저장된 좌표를 스택 2에는 스택1의 pop호출 시 pop된 위치에 새로 push 가능한 좌표를 저장하여 push와 pop을 통하여 재귀구조 없이 백트레킹을 한다. 스택1에서 동작을 하면서 스택2의 상태와 값을 통하여 동작을 진행하는 방식이며 스택2가 완전히 pop 때까지 반복한다

**<알고리즘(자연어)>**

1. 라인에 퀸 배치가능 좌표를 역순으로 스택2에 저장
2. 스택2에 값을 pop 후 스택1에 push
3. 만약 라인에 배치 할 곳이 없다면 스택1의 x(라인)과 스택2의 x를 비교하여 같지 않다면 같게 될 동안 스택1을 pop한 뒤 같게 된 순간 스택1을 한번더 pop하고 스택2의 값을 push후 스택2를 pop
4. 1~3을 반복하다 스택1이 풀이고 퀸이 배치되었다면 다른 곳에 저장한 후 3번을 실행
5. 스택2가 하나라도 차있다면 1~4과정 반복. **(아래쪽 예시 참조.)**

