

BÁO CÁO VỀ BOARD.JAVA

1. **Board** đại diện cho bảng tetris ở một trạng thái, tổ chức dạng mảng 2 chiều kiểu Boolean, với true là vị trí lấp đầy và false chưa lấp (**grid**)
2. Các dữ liệu khác như widths và heights
 - Mảng **widths** chỉ số ô đã lấp đầy tại mỗi hàng => **clearRows()** biết khi hàng lấp kín
 - Mảng **heights** lưu chiều cao mà mỗi cột đã được lấp => **dropHeight()** tính tọa độ 1 mảnh sẽ rơi xuống khi thả tại 1 cột
3. Song với nó là 1 cấu trúc tương tự gồm **gridBackup**, **widthsBackup**, heightsBackup để lưu trạng thái của **Board** trước đó
4. **maxHeight** lưu chiều cao nhất của mảng

Constructor

- Khởi tạo bảng với tham số **width** và **height**
- Bảng ở trạng thái committed

public void **sanityCheck()**

- Kiểm tra xem bảng có được đồng nhất không, nếu không đồng nhất thì ném ngoại lệ
- DEBUG để kiểm tra xem đã kiểm tra chưa nếu DEBUG == false thì không kiểm tra

public int **dropHeight**(Piece piece, int x)

- Tính giá trị y nơi gốc tọa độ (0,0) của một mảnh sẽ tới nếu mảnh được thả từ trên cao thẳng xuống khi gốc tọa độ của nó nằm tại cột x

public void **setGrid**(int x, int y, boolean bool)

- Cài đặt lại grid dung trong hàm swapH()

Public int place(piece, x,y)

- Đặt mảnh Piece vào lưới và kiểm tra xem nếu
 - + đặt thành công trả về PLACE_OK
 - + đặt thành công và lấp đầy 1 hàng trả về PLACE_ROW_FILLED
 - + Một phần của mảnh nằm ngoài bảng PLACE_OUT_BOUNDS
 - + Đè lên khối đã lấp từ trước PLACE_BAD
- Bảng phải ở trạng thái committed trước khi đặt và có thể gọi hàm undo() để đưa bảng về trạng thái hợp lệ nếu gặp trường hợp không hợp lệ

int clearRows()

- Xóa hàng khi đã lấp đầy trả về số hàng đã xóa

undo()

- Đổi chỗ giữa các dữ liệu đã backup và bảng hiện tại bị lỗi
- Chỉ thực hiện khi bảng ở trạng thái không committed

public void **swapH()**

- Dịch chuyển board