BÀI TẬP – JAVA

* Trò chơi cờ caro: 0 - X
  1. Input: ma trận 10x10
  2. Giá trị của ma trận:
     1. -1 : chưa điền
     2. 0 : giá trị ô của người 1
     3. 1: giá trị ô của người 2

Yêu cầu:

* Cho ma trận với các giá trị -1, 0, 1. Hãy xây dựng hàm tại vị trí (i, j) thực hiện kiểm tra điều kiện như sau:
  1. Kiểm tra ô (i, j) có là ô trống hay không? Chú ý: một ô trống là ô mà giá trị tại đó = -1.
  2. 5 vị trí liên tiếp có chứa ô (i, j) theo hàng có thỏa mãn bằng nhau hay không?
  3. 5 vị trí liên tiếp có chứa ô (i, j) theo cột có thỏa mãn bằng nhau hay không?
  4. 5 vị trí liên tiếp có chứa ô (i, j) theo đường chéo chính có thỏa mãn bằng nhau hay không?
  5. 5 vị trí liên tiếp có chứa ô (i, j) theo đường chéo phụ có thỏa mãn bằng nhau hay không?
  6. Kiểm tra tính hòa: kiểm tra xem còn ô trống hay không?
* Xây dựng trò chơi

1. Khởi tạo game: gán các giá trị -1.
2. Hiển thị game (show)
3. int isPerson1 = 1;
4. while(true){

B1: Hiển thị game

B2: 1.f

true:

“Draw”

return;

B3: Nhập giá trị - row/col

B4: 1.a

true:

game[r][c] = isPerson1;

1.b/c/d/e

true:

“Chúc mừng”;

return;

false:

isPerson1 = 0;

continue;

false: continue

}

BÀI TẬP – JAVA

* Trò chơi Hangman – người treo cổ:
  1. Input: đầu vào là một từ, hoặc một cụm từ, hoặc một câu bằng tiếng Anh. Ví dụ: Hello world.
  2. Người chơi được lựa chọn các ký tự để kiểm tra sự xuất hiện trong từ được cho (a, b, c …, z).
  3. Nếu người chơi chọn ký tự đã có trong từ 🡺 hiển thị ký tự đó.
  4. Nếu người chơi chọn ký tự không có trong từ 🡺 count ++;
  5. Nếu count = 5 🡺 người chơi thua cuộc.
  6. Nếu toàn bộ từ đầu vào được xác định 🡺 người chơi thắng cuộc.
* Solution
  1. Xây dựng hàm hiển thị (\*\*\*\* \*\*\*a\* b\*\*c\*\*\*).
  2. Xây dựng hàm check (input: ký tự người chơi nhập, chuỗi ban đầu – hoặc chuỗi đã xử lý)
  3. Xây dựng hàm kiểm tra kết thúc.
  4. Mở rộng:
     1. Input: mảng – mỗi phần tử là một từ.
     2. Xây dựng hàm random để chọn một từ trong mảng => bắt đầu trò chơi.
  5. while {

…

}