

Algorithm jogoMulheresNaCiencia

Objetivo: Apresentar a lógica por trás do funcionamento do jogo Mulheres na Ciência

|> Declaração de funções

|> Objetivo: retornar um valor aleatório entre 1 e 6 para o programa principal

|> Parâmetros formais: não há

|> Valor de retorno: um valor gerado aleatoriamente entre 1 e 6

Dado()

begin

|> Declaração de constantes

tamanhoTabuleiro <-- 32

|> Abrir o arquivo

|> Abre os arquivos da jogada anterior ou cria um novo arquivo com os valores 0, na hora de iniciar uma nova partida

|> Geração do HTML

|> Abertura do arquivo e atribuição dos valores gravados

|> O valor da posição do jogador que foi lido será utilizado para colocá-lo na posição correta do tabuleiro, a partir da tag apropriada em HTML

|> Chamada da função Dado

valorDado <-- Dado()

|> Atualização da posição do jogador

posicaoJogador <- posicaoJogador + valorDado

|> Condições que imprimem a carta correspondente à casa em que o jogador caiu, caso seja necessário

|> Efetua o cálculo da nova posição, de acordo com a condição da carta da casa na qual o jogador caiu

if posicaoJogador = 2 then

posicaoJogador <-- posicaoJogador + 3 |> Atualização da posição do jogador

else if posicaoJogador = 4 then

posicaoJogador <-- posicaoJogador - 4 |> Atualização da posição do jogador

else if posicaoJogador = 5 then

posicaoJogador <-- posicaoJogador + 3 |> Atualização da posição do jogador

else if posicaoJogador = 6 then

posicaoJogador <-- posicaoJogador - 4 |> Atualização da posição do jogador

else if posicaoJogador = 7 then

posicaoJogador <-- posicaoJogador - 4 |> Atualização da posição do jogador

else if posicaoJogador = 10 then

posicaoJogador <-- posicaoJogador + 2 |> Atualização da posição do jogador

```

else if posicaoJogador = 11 then
  posicaoJogador <-- posicaoJogador - 2 |> Atualização da posição do jogador
else if posicaoJogador = 13 then
  posicaoJogador <-- posicaoJogador - 1 |> Atualização da posição do jogador
else if posicaoJogador = 14 then
  posicaoJogador <-- posicaoJogador + 1 |> Atualização da posição do jogador
else if posicaoJogador = 16 then
  posicaoJogador <-- posicaoJogador + 1 |> Atualização da posição do jogador
else if posicaoJogador = 19 then
  posicaoJogador <-- posicaoJogador - 3 |> Atualização da posição do jogador
else if posicaoJogador = 20 then
  posicaoJogador <-- posicaoJogador - 3 |> Atualização da posição do jogador
else if posicaoJogador = 22 then
  posicaoJogador <-- posicaoJogador + 3 |> Atualização da posição do jogador
else if posicaoJogador = 23 then
  posicaoJogador <-- posicaoJogador + 3 |> Atualização da posição do jogador
else if posicaoJogador = 24 then
  posicaoJogador <-- posicaoJogador - 4 |> Atualização da posição do jogador
else if posicaoJogador = 26 then
  posicaoJogador <-- posicaoJogador - 4 |> Atualização da posição do jogador
else if posicaoJogador = 27 then
  posicaoJogador <-- posicaoJogador - 2 |> Atualização da posição do jogador
else if posicaoJogador = 28 then
  posicaoJogador <-- posicaoJogador - 3 |> Atualização da posição do jogador
else if posicaoJogador = 29 then
  posicaoJogador <-- posicaoJogador - 4 |> Atualização da posição do jogador
else if posicaoJogador = 30 then
  posicaoJogador <-- posicaoJogador - 5 |> Atualização da posição do jogador
else if posicaoJogador = 32 then
  posicaoJogador <-- posicaoJogador - 7 |> Atualização da posição do jogador
end if
|> Cada operação de adição ou subtração representa a quantidade de casas que o
jogador anda (para frente ou para trás)

|> Condição que verifica se o jogador venceu
if posicaoJogador > tamanhoTabuleiro then
  |> Abre a página de vitória

|> Caso o jogo continue, imprime a página com os dados atualizados e o jogador na
posição correta

|> Grava os valores atualizados no arquivo

end Algorithm

```