

Processing - Vetores e Matrizes

1. O código tem um vetor de tamanho 5 que recebe valores inteiros aleatórios de 0 a 3 e ao ser executado desenha 5 quadrados de cor cinza. Modifique o código de forma que cada um dos quadrados tenham a cor correspondente aos valores do vetor, sendo eles: 0 = cinza, 1 = vermelho, 2 = verde, 3 = azul.

Exemplo: Se os valores do vetor forem [0, 1, 3, 2, 1] então as cores dos quadrados devem ser, da esquerda para a direita, cinza, vermelho, azul, verde, vermelho.



Obs1: a função fill() define a cor do próximo quadrado a ser desenhado e recebe como parâmetro três valores no padrão RGB (Red, Green, Blue). Sendo vermelho = (255, 0, 0), verde = (0, 255, 0) e azul = (0, 0, 255).

Obs2: verifique no console os valores do vetor a cada execução

2. O código mostra as coordenadas X e Y do mouse no canto superior esquerdo da janela e no canto inferior direito é mostrado os valores de todas as posições da matriz [5][2]. Insira código dentro da função mouseClicked() para que a matriz receba os valores atuais de X e Y, de forma que ao clicar uma vez, X é salvo em matriz[0][0] e Y em matriz [0][1]. Ao clicar novamente, o novo X deve ser salvo em matriz[1][0] e o novo Y em [1][1] e assim por diante, voltando à posição 0 ao chegar no final da matriz.

Exemplo: se o mouse for clicado 3 vezes nas respectivas coordenadas: X: 36, Y: 120; X: 200, Y: 40 e X: 400, Y: 350, então a matriz deve ter os valores:

```
matriz[5][2] = [36, 120,  
                200, 40,  
                400, 350,  
                0, 0,  
                0, 0]
```

Obs1: deve ser inserido código apenas dentro do mouseClicked() para atribuir os valores à matriz, não é necessário mexer nas outras partes do código, apenas entendê-lo.

Obs2: o código dentro de mouseClicked() já é executado automaticamente ao clicar no botão do mouse.