

Exercícios da Aula 05 - Parte 01

Recomendações:

- TODO
 - Estamos usando primariamente o Processing como ferramenta para desenvolvimento das respostas dos exercícios mas se você tem utilizado outra ferramenta pode também apresentar para os colegas seja via fórum ou via Discord!
1. Abra o projeto exercicio_01. Nele há um projeto Processing com duas abas. Na primeira, já defini um comportamento e preciso que você implemente o funcionamento da classe RodaDeCores
 - a. A classe RodaDeCores deve seguir a estrutura deste diagrama de classe


RodaDeCores
- cores : color[] - indiceCorAtual : int
+ mudaProximaCor() + corAtual() : color

Portanto, escreva na aba RodaDeCores os atributos e os métodos que farão parte da classe. Já defini a assinatura do construtor que será utilizado. Mesmo sem saber a utilidade deles ainda, você já pode descrevê-los na classe.

- b. Escreva o corpo do construtor RodaDeCores. Nele, você recebe 3 cores como parâmetro e deve inicializar o vetor cores atribuindo c1 à posição cores[0] e assim por diante. Além disso, deve inicializar o inteiro indiceCorAtual.
- c. Implemente o método corAtual(). Ele deve retornar um dos elementos de cores, mais especificamente aquele indexado pelo valor de corAtual.
- d. Implemente o método mudaProximaCor(). Como indiceCorAtual indexa os elementos em cores, este atributo deve variar entre 0, 1, 2, 0, 1, 2 .. a cada chamada de mudaProximaCor(). Ou seja, se indiceCorAtual é 0, seu valor deve virar 1, se for 1 deve virar 2 e se for 2 deve virar 0. Você já construiu esse mecanismo em MAMI.

Feito isso, rode o projeto. Se você fez tudo correto, deve aparecer dois retângulos na tela e a cada vez que você clicar em um deles, a cor do mesmo vai mudar. Percebe então que na aba exercicio_01 eu não precisei saber como você implementou sua

RodaDeCores, só precisei entender como funcionam os métodos públicos! Veja também o código que mandei para que você revise um pouco do *evento* de clique do mouse.

2. Abra o projeto exercicio_02. Nele há um projeto Processing com três abas. Na primeira, já defini um comportamento e preciso que você implemente a o funcionamento da classe RodaDeCores e da Quadrado
 - a. Copie a classe RodaDeCores que você implementou anteriormente, não vamos reinventar a roda 
 - b. A classe Quadrado deve seguir a estrutura deste diagrama de classe

Quadrado
???
+ getXSuperiorEsquerdo() : int + getYSuperiorEsquerdo() : int + getLado() : int + toString(): String

Perceba que não me interessei em como você vai representar internamente a classe Quadrado, para eu que vou fazer uso dela só importa como ela se comunica com o mundo exterior.

- c. A classe Quadrado define um quadrado em um sistemas de eixo X/Y como apresentado pelo Processing. Logo, defina um construtor que recebe como parâmetro 3 inteiros, o ponto X do canto inferior direito, o ponto Y do canto inferior direito e o tamanho do lado do quadrado. Lembre-se que no Processing, quanto mais para baixo maior o Y e quanto mais para a direita, maior o X.
- d. Implemente os 3 primeiros métodos pedidos. getXSuperiorEsquerdo() retorna um inteiro referente à coordenada X do canto superior esquerdo do quadrado. getYSuperiorEsquerdo() faz o mesmo mas para a coordenada Y. getLado() é bem mais simples e retorna o tamanho do lado do quadrado.
- e. Implemente o método toString() que deve retornar como resultado uma string representando textualmente o objeto da classe Quadrado que fez a chamada. Você que define como representar textualmente esses objetos, então você deve retornar uma String mais ou menos assim “o quadrao tem coordenadas ...”. Dica: para converter de inteiro para string, você pode usar a função str(valor). + é um operador que concatena strings.

Feito isto, rode o projeto. Se você fez tudo correto, deve aparecer novo quadrado de cor diferente a cada vez que você segurar e depois soltar o mouse. Além disso, vai aparecer uma mensagem no console dizendo os dados do quadrado que acabou de ser criado! Perceba como é interessante: na linha 50 da aba exercicio_02 eu faço a chamada de println(quad). Essa chamada só funciona pois você criou o método toString() em Quadrado e ela reproduz exatamente a string que você definiu nesse método.

