

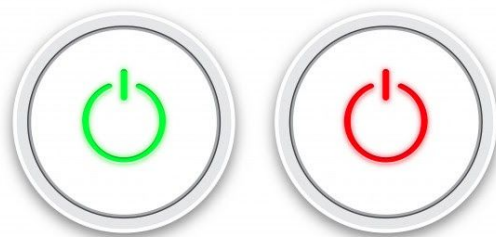
### Exercícios da Aula 05 - Parte 02 - Herança

Recomendações:

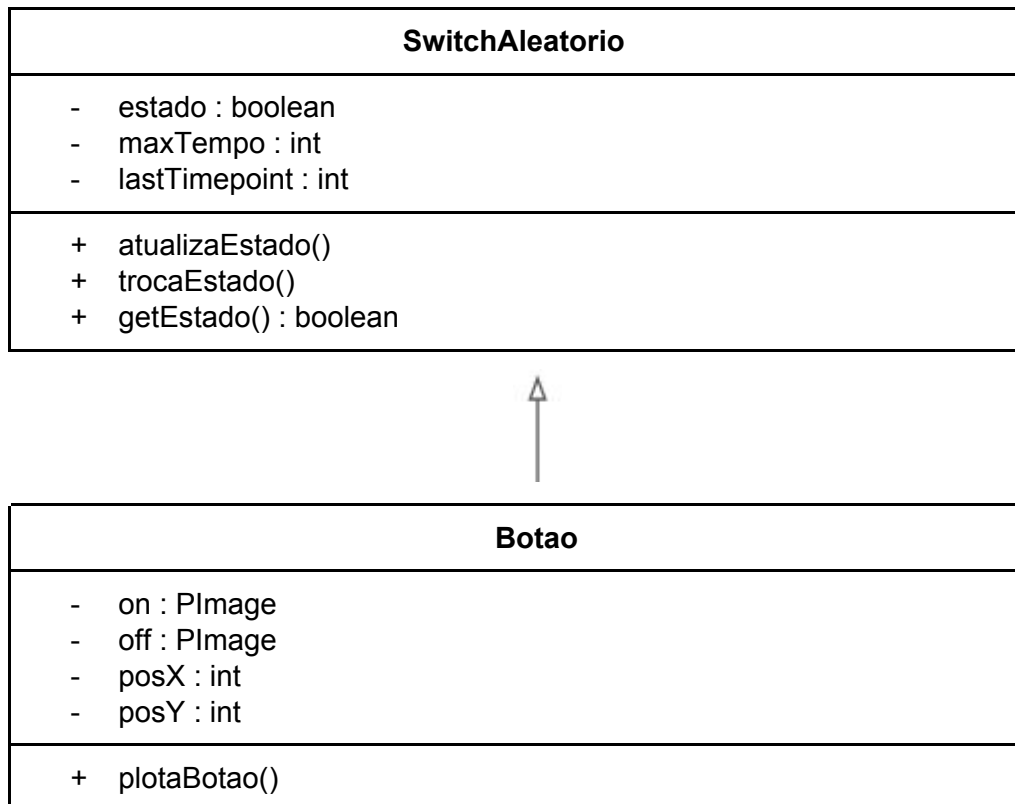
- Estamos usando primariamente o Processing como ferramenta para desenvolvimento das respostas dos exercícios mas se você tem utilizado outra ferramenta pode também apresentar para os colegas seja via fórum ou via Discord!
- Discord => Comunicação com colegas e com o monitor da disciplina, o Daniel Mesquita
- Fórum do Solar => Comunicação com colegas e comigo

# ATENÇÃO

O exercício 01, quando completa, possuirá 15 dos botões da imagem abaixo trocando de verde para vermelho em ciclos de 0,5 segundo a 1,5 segundo. Logo, se você tiver epilepsia ou qualquer condição que possa ser afetada pela troca de cores, favor não desenvolver o exercício e me avisar para que eu não gere mais elementos com essa característica.



1. Abra o projeto exercicio\_01\_parte02. Nele há um projeto Processing com tres abas. Na primeira, já defini uma classe chamada SwitchAleatorio, que é basicamente uma classe que define o comportamento um objeto com dois estados possíveis (true e false) que ficam sendo trocados de maxTempo em maxTempo. Abra esta aba e tente entender o código lá presente.
  - a. Na aba Botao, você deve implementar a classe Botão definida a seguir



Abaixo eu apresento parte do construtor dessa nova classe que você vai criar

```
Botao(int x, y, String fileOn, String fileOff, int max){
    // chamada do construtor da classe SwitchAleatorio
    super(int(random(500, max)));

    this.on = loadImage(fileOn);
    // complete o construtor aqui
}
```

Aqui há uma novidade que ainda vamos estudar. A primeira linha do construtor de Botao faz a chamada de `super()` que é o meio de se fazer a chamada do construtor da superclasse SwitchAleatorio, para que todos os atributos herdados desta classe possam ser inicializados. O atributo `on` armazena a imagem que deve ser plotada quando o estado do botao é `true` e a `PImage` `off` armazena a imagem quando o estado é `false`. Já `x` e `y` armazenam a posição superior esquerda onde deve ser plotada a imagem.

- b. Implemente o método `plotaBotao()`. Como o nome diz, ele deve plotar o botão dependendo do estado. Dica: para implementar este método você vai precisar de um condicional `if`, uma chamada para o método que retorna o estado `boolean` de um botao e usar a função `image()` do processing, que desenha uma imagem na tela.
- c. Com suas palavras, descreva em um comentário multilinha (aquele definido entre `/* COMENTARIO AQUI */` acerca do que está acontecendo nos procedimentos `draw()` e `mouseClicked()`. Para esta explicação, você deve

também explicar o funcionamento dos métodos que são executados dentro desses procedimentos.

Feito isso, rode o projeto. Se você fez tudo correto, devem 15 botões organizados em um grid. Se você clicar num botão On você ganha um ponto, se clicou em um botão Off, perde um ponto. Ao clicar em um botão On, ele se torna off automaticamente.

- d. Eu não fiz SMD :/, logo meu game design é péssimo. Como desafio, modifique o jogo, ajustando os tempos e informando a atualização de pontos diferente de escrever uma linha no console. Compartilhe sua adaptação no fórum e/ou no Discord