**POINTCUTS**

DisplayUpdating.aj

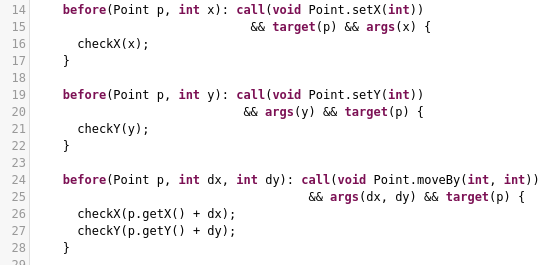


**ADVICE**

DisplayUpdating.aj



PointBoundsChecking.aj



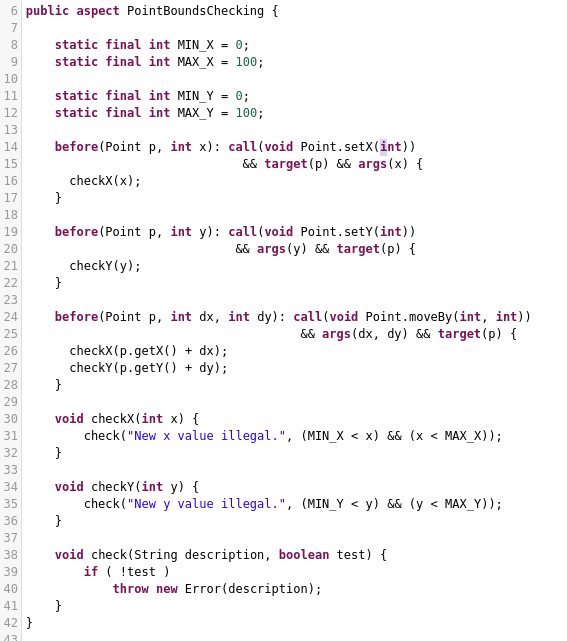
**WILDCARDS**

Sim,

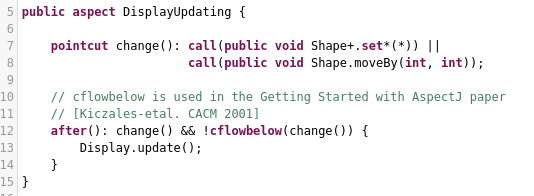
DisplayUpdating.aj



**ASPECT**

PointBoundsChecking.aj

DisplayUpdating.aj



**Q1. O que os dois aspectos do exemplo estão fazendo, qual propósito? Que se ganha com eles?**

Eles estão permitindo, nesse caso o acréscimo de métodos de pointcuts e advices, fora as propriedades de classe normal como em *PointBoundsChecking.aj*, que permite validar se o valor de x e y está dentro do esperado para o caso do *PointBoundsChecking.aj*, e permite atualizar o display depois de cada mudança no *DisplayUpdating.aj*.  
 O bom disso é que com a centralização desses métodos em uma classe modular você lida melhor com os “crosscutting concerns” que irão surgir no código.