

O método `isAlertPeriod()` verifica se o horário atual está dentro do intervalo configurado para o modo de alerta. Se o horário inicial for depois do horário final, considera-se que o período de alerta cruza a meia-noite. Nesse caso, o método retorna verdadeiro se o horário atual for depois do horário inicial ou antes do horário final. O método retorna verdadeiro apenas se o horário atual estiver entre o início e o fim.

O método `doAlert()` executa um loop enquanto o período de alerta estiver ativo. Durante esse período, ele configura todos os semáforos controlados para o modo de alerta piscante (geralmente o amarelo intermitente) e aguarda por um segundo. Esse comportamento simula o funcionamento reduzido do semáforo durante horários de menor movimento, como de madrugada.

O método `doYellowRedGreen()` representa o ciclo de operação normal dos semáforos. Primeiro, define todos os semáforos como amarelos e espera pelo tempo correspondente ao amarelo. Em seguida, define os semáforos como vermelhos e aguarda o tempo do vermelho. Por fim, muda todos para verde e aguarda o tempo do verde.

O método `run()` é responsável por iniciar o funcionamento contínuo do controle de semáforo. Ele cria uma nova thread que, enquanto o estado do controle estiver ligado (ON), executa o método `doAlert()` se for período de alerta e, caso contrário, executa o método `doYellowRedGreen()`. Caso ocorra uma interrupção na thread, todos os semáforos são configurados para o modo de alerta.

O aplicativo `AppOneWaySimpleSemaphoreControlTest` cria e exibe um único semáforo controlado por um sistema que alterna entre os estados verde, amarelo, vermelho e um modo de alerta com luz piscante. O controle é iniciado com um período de alerta de 30 segundos, e o semáforo é mostrado em uma janela gráfica.

Já o `AppOneWayTwoSemaphoreControlTest` faz o mesmo, mas com dois semáforos exibidos na mesma janela. Ambos funcionam de forma sincronizada, compartilhando o mesmo controle e seguindo o mesmo ciclo de estados e período de alerta.