

**Data:** 07 de julho de 2015.      **Prof.:** Fernando Castor

A classe `Rooms` gerencia uma coleção de quartos, indexada de 0 a  $m$  (onde  $m$  é um argumento para o construtor). Threads podem entrar ou sair de qualquer quarto nessa faixa. Cada quarto pode conter uma quantidade arbitrária de threads simultâneas, mas a qualquer momento, apenas um quarto pode estar ocupado. Por exemplo, se há dois quartos com índices 0 e 1, então qualquer número de threads pode entrar no quarto 0 mas nenhuma pode entrar no quarto 1 enquanto o quarto 0 ainda estiver ocupado.

Cada quarto pode ter atribuído a ele um **tratador de saída**: chamar `setHandler(i, h)` estabelece que `h` é o tratador de saída para o quarto `i`. O tratador de saída é chamado pela última thread a sair de um quarto, mas antes que qualquer thread entre em qualquer quarto. Este método é chamado exatamente uma vez e, enquanto ele está rodando, nenhuma thread entra em nenhum dos quartos. Implemente a classe `Rooms`. Certifique-se de que:

- Se alguma thread esta no quarto  $i$ , nenhuma thread está em um quarto  $j$  tal que  $i \neq j$ ;
- A última thread a sair de um quarto chama seu tratador de saída e nenhuma thread está em qualquer um dos quartos enquanto esse tratador está em execução;
- Threads não devem sofrer *starvation* ao tentar entrar nos quartos. Suponha que o escalonador é justo e que a possibilidade de *starvation* depende apenas da aplicação.