Trabalho 1 (PLP - 2017)

Autores:

Alexandre Regali Seleghim RA: 551473 Thiago Bretas de Souza RA: 551899

O programa possui 3 classes: AulasTeoricas.java Sala.java ReservaThread.java

AulasTeoricas.java

É onde o programa principal irá rodar. Aqui foi declarado o array de objetos (10 elementos da classe Sala) e inicializado o array com algumas instâncias de Sala.

Depois foram criadas as Threads (t1, t2 e t3) chamando pelo seu construtor. Após criadas e inicializadas as instâncias, as Threads foram iniciadas com start().

Sala.java

A classe sala possui 5 atributos:

- -ID (identificador da sala, tipo int)
- -tamanho (número de alunos que comporta, tipo int)
- -projetor (tem ou não projetor, tipo boolean)
- -ponto_acesso (nome de quem está utilizando a sala, tipo String)
- -esta_livre (caso a sala esteja livre/ocupada, tipo boolean)

Um objeto da classe Sala pode ser chamado pelo seu construtor para inicializar alguns de seus atributos, mas uma nova instância sempre será inicializada com ponto_acesso = null e esta_livre = true.

A classe possui todos os getters e setters de seus atributos.

ReservaThread.java

Essa classe estende a classe Thread.

Possui 5 atributos:

- -nome (nome de guem irá fazer a reserva/liberação das salas, tipo String)
- -array (array compartilhado com outras threads contendo todas as salas)
- -tamanho (tamanho de sala exigido por quem faz a reserva, tipo int)

- -projetor (se exige projetor ou não, tipo boolean)
- -vezes (quantas reservas serão feitas durante a execução, tipo int)

Há construtor para essa classe, sem nada de especial.

Possui 2 métodos:

-Reservar(array de salas, ponto_acesso, tamanho, projetor)
Irá procurar uma sala dentro do array que tenha as especificações necessárias (de tamanho, projetor e que esteja livre).

Caso não encontre, a thread espera com wait(). Quando acontecer alguma liberação, o método tentará uma nova busca.

Caso encontre, faz a reserva alterando os atributos do objeto sala (ponto_acesso = nome de quem faz a reserva, esta_livre = falso) e usa notify().

Também imprime no console anunciando que a reserva foi feita.

Após feito a reserva, é chamado a função liberar().

-Liberar(array de salas[index], ponto_acesso)

Sorteia um número (de 2000 a 8000). Esse número é passado na chamada sleep(), para determinar quanto tempo a thread esperará para liberar a sala (em milisegundos).

Após passado esse tempo, realiza a liberação alterando os atributos do objeto sala (ponto_acesso = null, esta_livre = true) e usa notify(). Também imprime no console anunciando que a liberação foi feita.

Tanto Liberar quanto Reservar são synchronized.

Em **run**(), apenas será chamado a função Reservar() com os parâmetros da classe. Essa chamada ocorrerá N vezes, sendo N a variável "*vezes*" da classe ReservaThread.

A função Liberar() já é chamada implicitamente dentro da função de reservar, então não é necessário chamá-la novamente.

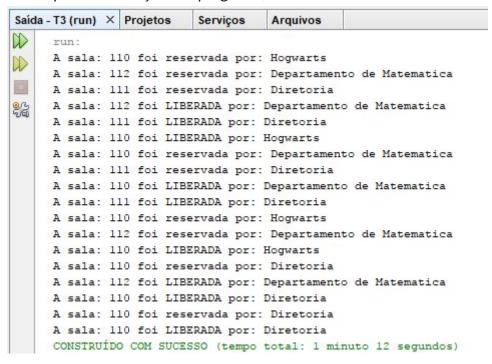
Dentro de run() também serão sorteados outros dois números para determinar um tempo de espera entre cada chamada de Reservar().

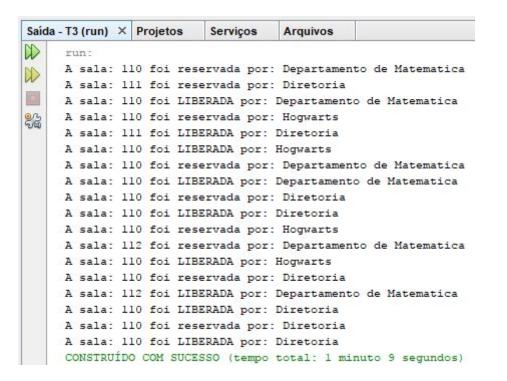
Exemplos

Instanciações de salas e de threads:

```
12
      public class AulasTeoricas {
13
14 🖃
           public static void main(String[] args) {
15
16
               //Sala asd = new Sala(110, 30, true);
17
              Sala[] lista_salas = new Sala[10];
18
19
              lista salas[0] = new Sala(110, 50, true);
              lista_salas[1] = new Sala(111, 30, true);
lista_salas[2] = new Sala(112, 35, true);
21
22
               lista_salas[3] = new Sala(113, 40, true);
24
               lista_salas[4] = new Sala(114, 50, false);
               lista_salas[5] = new Sala(115, 30, false);
25
               lista_salas[6] = new Sala(116, 10, false);
               lista_salas[7] = new Sala(117, 20, false);
27
               lista_salas[8] = new Sala(118, 50, false);
28
              lista_salas[9] = new Sala(119, 30, true);
29
30
31
32
               ReservaThread tl = new ReservaThread("Diretoria", lista_salas, 10, false, 4);
33
               ReservaThread t2 = new ReservaThread("Departamento de Matematica", lista_salas, 35, true, 3);
               ReservaThread t3 = new ReservaThread("Hogwarts", lista_salas, 50, true, 2);
34
35
               tl.start();
36
              t3.start();
37
38
39
40
41
42
```

3 exemplos de execuções do programa, obtendo diferentes resultados:





Saida -	T3	(run)	×	Projetos	Serviços	Arquivos	
	rı	in:		Co.	330	(i) - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 11	\$100 m
	A	sala:	11	0 foi res	ervada por	: Departame	nto de Matematica
	A	sala:	11	0 foi LI	BERADA por	Departamen	to de Matematica
	A	sala:	11	0 foi res	ervada por	: Hogwarts	
<mark>ම</mark> දු	A	sala:	11	l foi res	ervada poi	r: Diretoria	
	A	sala:	11	l foi LI	BERADA por	Diretoria	
	A	sala:	11	.0 foi LI	BERADA por	Hogwarts	
	A	sala:	11	.0 foi res	ervada poi	r: Diretoria	
	A	sala:	11	2 foi res	ervada poi	r: Departame	nto de Matematica
	A	sala:	11	0 foi LI	BERADA por	Diretoria	
	A	sala:	11	.0 foi res	ervada por	r: Hogwarts	
	A	sala:	11	2 foi LI	BERADA por	Departamen	to de Matematica
	A	sala:	11	.0 foi LI	BERADA por	Hogwarts	
	A	sala:	11	.0 foi res	ervada poi	r: Departame	nto de Matematica
	A	sala:	11	l foi res	ervada poi	r: Diretoria	
	A	sala:	11	l foi LI	BERADA por	Diretoria	
	A	sala:	11	.0 foi LI	BERADA por	Departamen	to de Matematica
	A	sala:	11	.0 foi res	ervada por	r: Diretoria	
	A	sala:	11	.0 foi LIE	BERADA por:	Diretoria	