

Trabalho 3 de PLP: Programação Concorrente

Felipe Sampaio de Souza, RA: 619523

Marcio Lima Inácio, RA: 587265

1 Classes

1.1 Sala

```
package com.company;

public class Sala
{
    private int identificador;
    private int tamanho;
    private boolean projetor;
    private String reserva;
    private boolean ocupada;

    // Construtor
    Sala(int id, int tam, boolean proj)
    {
        identificador = id;
        tamanho = tam;
        projetor = proj;
        reserva = null;
        ocupada = false;
    }

    boolean isOcupada()
    {
        return ocupada;
    }

    boolean getProjetor()
    {
        return projetor;
    }

    int getTamanho()
    {
        return tamanho;
    }

    int getIdentificador()
    {
        return identificador;
    }
}
```

```
void reserva(String pontoDeAcesso)
{
    if (!ocupada)
    {
        reserva = pontoDeAcesso;
        ocupada = true;
    }
}

void libera()
{
    if (ocupada)
    {
        reserva = null;
        ocupada = false;
    }
}

// Metodo toString() para facilitar impressoes na tela
@Override
public String toString()
{
    return "Sala: " + identificador + "; " +
           "Tamanho: " + tamanho;
}
}
```

1.2 Aulas Teoricas

```
package com.company;

import java.util.ArrayList;
import java.util.List;
import java.util.Random;

class AulasTeoricas
{
    private List<Sala> salas;

    // Construtor
    // Gera 10 salas com capacidade de ate 100 lugares e projetores aleatoriamente
    AulasTeoricas()
    {
        salas = new ArrayList<>();

        Random randomGenerator = new Random();

        for (int i = 0; i < 10; i++)
        {
            salas.add(new Sala(i, randomGenerator.nextInt(100), randomGenerator.nextBoolean()));
        }

        // Verifica se existe pelo menos uma sala que atenda aos requisitos
        // Nao importa se a sala esta reservada ou nao
        boolean existeSalaQueComporta(boolean requerProjetor, int requerTamanho)
        {
            for (Sala sala : salas)
            {
                if ((sala.getProjetor() == requerProjetor) && (sala.getTamanho() >= requerTamanho))
                {
                    return true;
                }
            }

            return false;
        }

        // Verifica se existe uma sala livre que atenda aos requisitos para ser reservada
        // Retorna a sala a ser reservada
        private Sala existeSalaAdequada(boolean requerProjetor, int requerTamanho)
        {
            for (Sala sala : salas)
            {
                if (!sala.isOcupada() && (sala.getProjetor() == requerProjetor) && (sala.getTamanho()
                    >= requerTamanho))
                {
                    return sala;
                }
            }
        }
    }
}
```

```

        return null;
    }

    // Reserva uma sala para pontoDeAcesso de acordo com os requisitos
    // Utiliza o metodo existeSalaAdequada para procurar a sala a ser reservada
    synchronized int reservar(String pontoDeAcesso, boolean requerProjeto, int requerTamanho)
        throws InterruptedException
    {
        Sala sala = existeSalaAdequada(requerProjeto, requerTamanho);

        // Enquanto nao houver sala para ser reservada, a Thread espera
        while (sala == null)
        {
            System.out.println("Nao foi possivel reservar uma sala para " + pontoDeAcesso);
            wait();
            sala = existeSalaAdequada(requerProjeto, requerTamanho);
        }
        // Existe uma sala que pode ser reservada

        // Realiza a reserva
        sala.reserva(pontoDeAcesso);
        System.out.println("Sala " + sala.getIdentificador() + " reservada para " +
            pontoDeAcesso);
        notify();
        return sala.getIdentificador();
    }

    // Libera a sala de identificador id
    // Percorre a lista de salas e busca pelo identificador
    // Nao verifica se a sala esta de fato reservada, pois se nao estiver o estado continua o
    mesmo
    synchronized void liberar(int id)
    {
        for (Sala sala : salas)
        {
            if (sala.getIdentificador() == id)
            {
                sala.libera();
                System.out.println("Sala " + sala.getIdentificador() + " liberada");
                notify();
                return;
            }
        }

        System.out.println("Nao foi possivel encontrar a sala " + id);
    }

    // Metodo toString() para facilitar impressoes na tela
    @Override
    public String toString()
    {
        StringBuilder retorno = new StringBuilder();

        for (Sala sala : salas)

```

```
{
    retorno.append(sala).append("\n");
}

return retorno.toString();
}
```

1.3 Thread para Reservas

```
package com.company;

import java.util.Random;

public class ReservaThread implements Runnable
{
    private String pontoDeAcesso;
    private AulasTeoricas salas;
    private Random randomGenerator;
    private int numeroDeOperacoes;

    // Construtor
    ReservaThread(String nome, AulasTeoricas ats)
    {
        pontoDeAcesso = nome;

        // Cada Thread possui um numero aleatorio de ate 10 operacoes
        randomGenerator = new Random();
        numeroDeOperacoes = randomGenerator.nextInt(9) + 1;

        // Dado compartilhado
        salas = ats;
    }

    @Override
    public void run()
    {
        for (int i = 0; i < numeroDeOperacoes; i++)
        {
            try
            {
                // Numero de lugares requeridos e presenca de Projetor sao gerados aleatoriamente
                int numeroDeLugares = randomGenerator.nextInt(80);
                boolean precisaDeProjetor = randomGenerator.nextBoolean();

                // Deve existir pelo menos uma sala que comporte os requisitos da reserva
                // Caso contrario a Thread ficara presa na chamada wait(), pois nao havera em
                // momento algum uma sala para ela
                while (!salas.existeSalaQueComporta(precisaDeProjetor, numeroDeLugares))
                {
                    numeroDeLugares = randomGenerator.nextInt(80);
                    precisaDeProjetor = randomGenerator.nextBoolean();
                }

                // Realiza a reserva da sala
                int salaId = salas.reservar(pontoDeAcesso, precisaDeProjetor, numeroDeLugares);
                // Espera ate 1 segundo para liberar a sala
                Thread.sleep((int) (Math.random() * 1000));
                // Libera a sala
                salas.liberar(salaId);
            }
            catch (InterruptedException e)
            {
            }
        }
    }
}
```

```
        {
            e.printStackTrace();
        }
    }
}

// Metodo toString() para facilitar impressoes na tela
@Override
public String toString()
{
    return "Thread de " + pontoDeAcesso + "; " +
        numeroDeOperacoes + " operacoes" + "\n";
}
}
```

1.4 Sistema de Reservas

```
package com.company;

public class SistemaDeReservas
{

    public static void main(String[] args) throws InterruptedException
    {
        // Cria e imprime as salas criadas
        AulasTeoricas ats = new AulasTeoricas();
        System.out.println("Salas criadas");
        System.out.println(ats);

        // Cria e imprime as Threads
        ReservaThread thread1 = new ReservaThread("Nicanor", ats);
        System.out.println(thread1);

        ReservaThread thread2 = new ReservaThread("Vania", ats);
        System.out.println(thread2);

        ReservaThread thread3 = new ReservaThread("Heloisa", ats);
        System.out.println(thread3);

        ReservaThread thread4 = new ReservaThread("Valter", ats);
        System.out.println(thread1);

        ReservaThread thread5 = new ReservaThread("Helena", ats);
        System.out.println(thread2);

        ReservaThread thread6 = new ReservaThread("Helio", ats);
        System.out.println(thread3);

        // Executa as Threads
        new Thread(thread1).start();
        new Thread(thread2).start();
        new Thread(thread3).start();
        new Thread(thread4).start();
        new Thread(thread5).start();
        new Thread(thread6).start();
    }
}
```

2 Execução

Primeira execução

Salas criadas

Sala: 0; Tamanho: 40

Sala: 1; Tamanho: 83

Sala: 2; Tamanho: 9

Sala: 3; Tamanho: 82

Sala: 4; Tamanho: 16

Sala: 5; Tamanho: 91

Sala: 6; Tamanho: 8

Sala: 7; Tamanho: 73

Sala: 8; Tamanho: 43

Sala: 9; Tamanho: 10

Thread de Nicanor; 1 operacoes

Thread de Vania; 3 operacoes

Thread de Heloisa; 1 operacoes

Thread de Nicanor; 1 operacoes

Thread de Vania; 3 operacoes

Thread de Heloisa; 1 operacoes

Sala 1 reservada para Nicanor

Sala 3 reservada para Vania

Sala 5 reservada para Heloisa

Sala 0 reservada para Valter

Sala 8 reservada para Helena

Nao foi possivel reservar uma sala para Helio

Sala 0 liberada

Sala 7 reservada para Valter

Nao foi possivel reservar uma sala para Helio

Sala 3 liberada

Sala 3 reservada para Helio

Nao foi possivel reservar uma sala para Vania

Sala 8 liberada

Sala 8 reservada para Helena

Nao foi possivel reservar uma sala para Vania

Sala 3 liberada

Sala 3 reservada para Helio

Nao foi possivel reservar uma sala para Vania

Sala 5 liberada

Sala 5 reservada para Vania

Sala 7 liberada

Sala 7 reservada para Valter
Sala 1 liberada
Sala 8 liberada
Nao foi possivel reservar uma sala para Helena
Sala 7 liberada
Sala 1 reservada para Valter
Sala 7 reservada para Helena
Sala 5 liberada
Sala 0 reservada para Vania
Sala 3 liberada
Sala 1 liberada
Sala 1 reservada para Valter
Sala 7 liberada
Sala 2 reservada para Helena
Sala 0 liberada
Sala 1 liberada
Sala 5 reservada para Valter
Sala 2 liberada
Sala 1 reservada para Helena
Sala 1 liberada
Sala 0 reservada para Helena
Sala 0 liberada
Sala 0 reservada para Helena
Sala 5 liberada
Sala 0 liberada

Segunda execucao

Salas criadas

Sala: 0; Tamanho: 43
Sala: 1; Tamanho: 30
Sala: 2; Tamanho: 93
Sala: 3; Tamanho: 34
Sala: 4; Tamanho: 3
Sala: 5; Tamanho: 90
Sala: 6; Tamanho: 74
Sala: 7; Tamanho: 37
Sala: 8; Tamanho: 69
Sala: 9; Tamanho: 73

Thread de Nicanor; 2 operacoes

Thread de Vania; 3 operacoes

Thread de Heloisa; 6 operacoes

Thread de Nicanor; 2 operacoes

Thread de Vania; 3 operacoes

Thread de Heloisa; 6 operacoes

Sala 5 reservada para Nicanor
Sala 6 reservada para Vania
Sala 0 reservada para Heloisa
Sala 9 reservada para Valter
Nao foi possivel reservar uma sala para Helena
Sala 2 reservada para Helio
Nao foi possivel reservar uma sala para Helena
Sala 5 liberada
Sala 1 reservada para Nicanor
Sala 5 reservada para Helena
Sala 9 liberada
Sala 9 reservada para Valter
Sala 1 liberada
Sala 6 liberada
Sala 6 reservada para Vania
Sala 5 liberada
Sala 8 reservada para Helena
Sala 2 liberada
Sala 5 reservada para Helio
Sala 0 liberada
Sala 0 reservada para Heloisa
Sala 9 liberada
Sala 9 reservada para Valter
Sala 8 liberada
Nao foi possivel reservar uma sala para Helena
Sala 5 liberada
Sala 5 reservada para Helena
Sala 1 reservada para Helio
Sala 0 liberada
Nao foi possivel reservar uma sala para Heloisa
Sala 6 liberada
Sala 6 reservada para Vania
Nao foi possivel reservar uma sala para Heloisa
Sala 1 liberada
Nao foi possivel reservar uma sala para Heloisa
Sala 0 reservada para Helio
Nao foi possivel reservar uma sala para Heloisa
Sala 9 liberada
Sala 9 reservada para Heloisa
Sala 0 liberada
Sala 2 reservada para Helio
Sala 6 liberada
Sala 5 liberada
Sala 8 reservada para Helena
Sala 8 liberada
Sala 5 reservada para Helena
Sala 9 liberada
Sala 0 reservada para Heloisa
Sala 0 liberada

Sala 6 reservada para Heloisa
Sala 5 liberada
Sala 6 liberada
Sala 0 reservada para Heloisa
Sala 2 liberada
Sala 0 liberada

Terceira execução

Salas criadas
Sala: 0; Tamanho: 28
Sala: 1; Tamanho: 35
Sala: 2; Tamanho: 0
Sala: 3; Tamanho: 4
Sala: 4; Tamanho: 64
Sala: 5; Tamanho: 1
Sala: 6; Tamanho: 77
Sala: 7; Tamanho: 68
Sala: 8; Tamanho: 26
Sala: 9; Tamanho: 66

Thread de Nicanor; 6 operacoes

Thread de Vania; 2 operacoes

Thread de Heloisa; 3 operacoes

Thread de Nicanor; 6 operacoes

Thread de Vania; 2 operacoes

Thread de Heloisa; 3 operacoes

Sala 1 reservada para Nicanor
Sala 4 reservada para Vania
Sala 6 reservada para Heloisa
Sala 0 reservada para Valter
Sala 7 reservada para Helena
Sala 9 reservada para Helio
Sala 6 liberada
Sala 6 reservada para Heloisa
Sala 0 liberada
Sala 0 reservada para Valter
Sala 6 liberada
Nao foi possivel reservar uma sala para Heloisa
Sala 7 liberada
Nao foi possivel reservar uma sala para Heloisa
Sala 9 liberada
Nao foi possivel reservar uma sala para Heloisa
Sala 6 reservada para Helio

Nao foi possivel reservar uma sala para Heloisa
Sala 0 liberada
Nao foi possivel reservar uma sala para Heloisa
Sala 1 liberada
Sala 1 reservada para Nicanor
Nao foi possivel reservar uma sala para Heloisa
Sala 4 liberada
Nao foi possivel reservar uma sala para Heloisa
Sala 2 reservada para Vania
Nao foi possivel reservar uma sala para Heloisa
Sala 6 liberada
Nao foi possivel reservar uma sala para Heloisa
Sala 4 reservada para Helio
Nao foi possivel reservar uma sala para Heloisa
Sala 1 liberada
Sala 6 reservada para Nicanor
Sala 1 reservada para Heloisa
Sala 6 liberada
Sala 0 reservada para Nicanor
Sala 1 liberada
Sala 0 liberada
Sala 6 reservada para Nicanor
Sala 2 liberada
Sala 4 liberada
Sala 7 reservada para Helio
Sala 6 liberada
Sala 0 reservada para Nicanor
Sala 7 liberada
Sala 1 reservada para Helio
Sala 0 liberada
Sala 1 liberada
Sala 4 reservada para Helio
Sala 4 liberada
Sala 4 reservada para Helio
Sala 4 liberada
Sala 4 reservada para Helio
Sala 4 liberada