

ESCOLA DE ARTES, CIÊNCIAS E HUMANIDADES
COMPUTAÇÃO ORIENTADA A OBJETOS – ACH2003-104-2021
PROF. MARCOS LORDELLO CHAIM
RELATÓRIO DO EXERCÍCIO PROGRAMA 2

HEITOR LOPES BIANCHI – 10258730
MYCAELLI CERQUEIRA DE LIMA – 10723562

Introdução

O relatório a seguir contém detalhes da implementação proposta para o exercício, instruções para a compilação através de um arquivo makefile, exemplos de utilização do sistema construído assim como suas respectivas saídas e um diagrama de classes UML.

Implementação

Além das classes propostas no enunciado do exercício, `MarcadorDeReuniao` e `GerenciadorDeSalas`, foram desenvolvidas algumas classes auxiliares para facilitar implementação dos métodos propostos nas classes primárias:

- **Reunião:** mantém o rastreo dos participantes de uma reunião, a intersecção de disponibilidades dos participantes – instâncias da classe `Participante` –, o início e o fim da mesma.
- **Sala:** instanciada por objetos que delimitam a capacidade uma sala de reunião, assim como seu nome, local e observações necessárias.
- **Disponibilidade:** classe de intervalos de tempo com início e fim utilizado em `Participante`, `Reunião` e `Reserva`.
- **Participante:** define objetos participantes de uma `Reunião`, cada um tendo seu e-mail e um início e fim da disponibilidade (atributos isolados e também em um `ArrayList`).
- **Reserva:** define a reserva feita em uma sala para uma reunião com seus participantes e intervalo de duração definido através da intersecção de disponibilidade dos participantes.

Além dos construtores, getters e setters das classes auxiliares, os principais métodos implementados são os das classes principais propostos no próprio enunciado desenvolvidos através de instâncias das classes auxiliares.

Compilação

Para facilitar a compilação dos arquivos foi criado um Makefile incluído junto no diretório das classes e possui a seguinte forma:

```
all: EP2

EP2:
    javac *.java -Xlint:unchecked

clean:
    rm -f *.class *.exe
```

Para utiliza-lo basta navegar no prompt de comando (cmd) até o diretório das classes e aplicar o comando *make*, gerando assim o .class de cada .java.

Exemplo

Para testar e exemplificar o funcionamento do sistema foi desenvolvida uma classe externa ao mesmo que executa alguns comandos:

```
public static void main(String[] args) {
    //Código teste para entender o fluxo de dados do programa

    Collection<String> participantes = new ArrayList<String>();

    participantes.add("Paulo");
    participantes.add("Pedro");
    participantes.add("Ale");

    MarcadorDeReuniao marcador = new MarcadorDeReuniao();
    marcador.marcarReuniaoEntre(LocalDate.parse("2019-01-01"), LocalDate.parse("2019-01-10"), participantes);

    marcador.indicaDisponibilidadeDe("Paulo", LocalDateTime.parse("2019-01-01T08:30"), LocalDateTime.parse("2019-01-01T10:00"));
    marcador.indicaDisponibilidadeDe("Paulo", LocalDateTime.parse("2019-01-01T16:00"), LocalDateTime.parse("2019-01-01T18:00"));
    marcador.indicaDisponibilidadeDe("Paulo", LocalDateTime.parse("2019-01-03T12:00"), LocalDateTime.parse("2019-01-03T16:00"));
    marcador.indicaDisponibilidadeDe("Pedro", LocalDateTime.parse("2019-01-01T09:00"), LocalDateTime.parse("2019-01-01T17:00"));
    marcador.indicaDisponibilidadeDe("Pedro", LocalDateTime.parse("2019-01-03T11:00"), LocalDateTime.parse("2019-01-03T13:00"));
    marcador.indicaDisponibilidadeDe("Pedro", LocalDateTime.parse("2019-01-03T15:00"), LocalDateTime.parse("2019-01-03T17:00"));
    marcador.indicaDisponibilidadeDe("Ale", LocalDateTime.parse("2019-01-02T01:00"), LocalDateTime.parse("2019-01-02T23:00"));
    marcador.indicaDisponibilidadeDe("Ale", LocalDateTime.parse("2019-01-01T09:30"), LocalDateTime.parse("2019-01-01T16:30"));
    marcador.indicaDisponibilidadeDe("Ale", LocalDateTime.parse("2019-01-03T12:30"), LocalDateTime.parse("2019-01-03T15:30"));

    marcador.novaReuniao.mostraSobreposicao();
}
```

A partir desse bloco de testes a saída obtida foi compatível com a esperada pela proposta do exercício:

```
Participante: Paulo
De 01/01/2019 08:30 ate 01/01/2019 10:00
De 01/01/2019 16:00 ate 01/01/2019 18:00
De 03/01/2019 12:00 ate 03/01/2019 16:00

Participante: Pedro
De 01/01/2019 09:00 ate 01/01/2019 17:00
De 03/01/2019 11:00 ate 03/01/2019 13:00
De 03/01/2019 15:00 ate 03/01/2019 17:00

Participante: Ale
De 02/01/2019 01:00 ate 02/01/2019 23:00
De 01/01/2019 09:30 ate 01/01/2019 16:30
De 03/01/2019 12:30 ate 03/01/2019 15:30

Lista de horarios disponiveis para todos:
De 01/01/2019 09:30 ate 01/01/2019 10:00
De 01/01/2019 16:00 ate 01/01/2019 16:30
De 03/01/2019 12:30 ate 03/01/2019 13:00
De 03/01/2019 15:00 ate 03/01/2019 15:30
```

Diagrama de Classes

