

Exercício Programa 1

Marcador de Reuniões

Prof. Dr. Daniel Cordeiro
Escola de Artes, Ciências e Humanidades
Universidade de São Paulo

Entrega: 30 de abril de 2019

Introdução

Este é um programa¹ que será utilizado para auxiliar na marcação de uma reunião com vários participantes. Um dos participantes, responsável pela organização da reunião, indica em qual período ele gostaria de marcar a reunião (por exemplo, entre 10 e 20 de abril); ele determina também a lista de participantes identificados pelos seus endereços eletrônicos. A seguir, cada um dos participantes indica os horários de disponibilidade dentro do período determinado pelo organizador. O organizador então visualiza a sobreposição dos horários de todos os participantes e escolhe um horário para a reunião.

O trabalho deve ser feito preferencialmente em grupos de 2. Grupos individuais são aceitos (a contragosto ☹). Recomendo fortemente que os pares desenvolvam os programas usando programação pareada (os dois programadores sentados lado a lado compartilhando o mesmo computador).

Fase 1: disponibilidade de horários

Para a Fase 1, que deve ser entregue até o final de abril, a implementação deve se basear em uma interface de texto somente, ou seja, a visualização dos horários dos participantes será feita em modo texto utilizando-se, por exemplo, o objeto Console.

A definição dos participantes da reunião será feita utilizando-se do seguinte método que deverá ser implementado na classe `MarcadorDeReuniao`:

```
public void marcarReuniaoEntre(LocalDate dataInicial,  
                               LocalDate dataFinal,  
                               Collection<String> listaDeParticipantes)
```

¹Adaptado de um EP de MAC0441 gentilmente cedido pelo prof. Fábio Kon (IME).

As datas devem ser do tipo `java.time.LocalDate` e a `listaDeParticipantes` deve ser uma `Collection` de `Strings` que identificam os participantes.

Cada participante define os seus horários disponíveis através do seguinte método:

```
public void indicaDisponibilidadeDe(String participante,
                                   LocalDateTime inicio,
                                   LocalDateTime fim)
```

onde cada participante é identificado com uma `String` e o início e fim da disponibilidade é indicado com dias e horários dados por instâncias de `java.time.LocalDateTime`.

Finalmente, você deve implementar o método

```
public void mostraSobreposicao()
```

que, provavelmente, vai dar o maior trabalho e vai exigir mais criatividade de vocês para informar os dados de uma forma clara e elegante. Ele deve exibir um relatório com as escolhas realizadas e indicar em quais horários todos os participantes poderiam participar da reunião.

Entrega

Um arquivo compactado no formato `.tar.xz` ou `.zip` deve ser entregue no eDisciplinas contendo o código-fonte produzido, uma versão compilada (arquivos `.class`) do programa e um pequeno relatório (nos formatos abertos ODT², PDF ou TXT) com o nome e número USP de todos os integrantes do grupo, exemplos de utilização do sistema e suas respectivas saídas.

O seu EP será recompilado e testado em um computador com o seguinte software instalado:

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description: Debian GNU/Linux buster/sid
Release: testing
Codename: buster

$ java -version
openjdk version "12" 2019-03-19
OpenJDK Runtime Environment (build 12+0-Debian-3)
OpenJDK 64-Bit Server VM (build 12+0-Debian-3, mixed mode, sharing)
```

Dúvidas, problemas, etc.

Quais dúvidas ou problemas relacionados ao EP devem ser discutidos no fórum de Discussões no eDisciplinas: <https://edisciplinas.usp.br/mod/forum/view.php?id=2534318>.

²OpenDocument Format: <https://www.libreoffice.org/discover/what-is-opendocument/>.