



## MC322 - Laboratório 2

### Prof. Esther Colombini

esther@ic.unicamp.br

<http://www.ic.unicamp.br/~esther/teaching/2020s2/mc322>

### PEDs:

Pedro Santos de Rezende Alves (pedrorezendesantos@gmail.com)

Renata Falguera Gonçalves (renatafalguera@gmail.com)

### PADs:

Bernardo do Amaral Teodosio (b167494@dac.unicamp.br)

Fabrício de Souza Maruta (f138313@dac.unicamp.br)

Gabriel de Freitas Garcia (g216179@dac.unicamp.br)

---

## 1 Descrição Geral

Com o avanço da pandemia de SARS-COV-2 e a necessidade mundial de isolamento, atividades acadêmicas em todo o mundo migraram para um modelo temporário integralmente virtual, o que culminou em uma mudança de paradigma e necessidade de reorganização do modelo de estudo por parte dos estudantes. Neste cenários, ferramentas capazes de permitir a organização das atividades (Trello), a comunicação entre os alunos divididos por grupos de interesse (Discord, Slack) e a troca de material de apoio, se tornaram importantes aliados no processo de aprendizagem. Neste contexto, os laboratórios desenvolvidos ao longo do semestre terão por objetivo construir uma ferramenta colaborativa de organização das atividades acadêmicas dos alunos com vistas à comunicação, troca de conteúdos e alinhamento de atividades do semestre letivo.

## 2 Objetivo

O objetivo desta atividade consiste na familiarização do aluno com a construção de classes que possuam atributos e métodos estáticos, bem como a utilização de sobrecarga de métodos e de construtores.

## 3 Atividade

Nesta atividade, adicionaremos ao projeto do Lab1 mais duas classes: Sala e Cartao. Essas classes contém atributos e métodos estáticos capazes de auxiliar, futuramente, no cômputo de estatísticas do sistema. Na figura 1 apresentamos o diagrama UML destas classes.

Implemente as novas classes conforme apresentado no diagrama UML e seguindo as instruções descritas a seguir, no item 3.1.

### 3.1 Observações

- Para as duas classes, todos os métodos de acesso para as propriedades de classe e de instância devem ser criados.
- Cada uma das classes deve ter ao menos dois construtores sobrecarregados implementados. Um construtor que não recebe parâmetros e o outro a sua escolha.

Sala	Cartao
<ul style="list-style-type: none"> <li>- numeroSalas: int</li> <li>- id: int</li> <li>- capacidadeMaxima: int</li> <li>- numeroUsuarios: int</li> <li>- descricao: String</li> <li>- dono: Usuario</li> <li>- ingressoAutomatico: boolean</li> <li>- dataCriacao: Calendar</li> </ul>	<ul style="list-style-type: none"> <li>- numeroCartoes: int</li> <li>- id: int</li> <li>- visibilidade: int</li> <li>- nome: String</li> <li>- dono: Usuario</li> <li>- invitationOnly: boolean</li> <li>- dataCriacao: Calendar</li> </ul>
<ul style="list-style-type: none"> <li>+ toString(): String</li> <li>+ adicionaUsuario(): boolean</li> <li>+ removeUsuario(): boolean</li> <li>+ getNumeroSalas(): int</li> </ul>	<ul style="list-style-type: none"> <li>+ toString(): String</li> <li>+ getNumeroCartoes(): int</li> </ul>

Figura 1: Diagrama UML das classes

- O *id* da sala e do cartão deve ser inicializado automaticamente pelo sistema com o número da última sala ou cartão criado.
- O método **adicionaUsuario()** deve adicionar um usuário à sala, incrementando o valor de **numeroUsuarios**. A **capacidadeMaxima** da sala não deve ser ultrapassada e o método deve devolver se foi possível ou não adicionar um usuário.
- O método **removeUsuario()** deve remover um usuário a sala, decrementando o valor de **numeroUsuarios**. O método deve devolver se foi possível ou não remover um usuário. Você pode criar métodos auxiliares adicionais **salaCheia()** e **salaVazia()** caso queira.

Após a criação de cada classe do diagrama UML, crie um `main` onde você instanciará objetos do tipo Sala e Cartao, utilizando cada um dos construtores criados anteriormente, ou seja, serão ao menos 2 objetos de cada classe. Você deve utilizar `System.in` para fazer com que a instanciação das classes seja interativa, ou seja, o usuário deverá definir os valores dos parâmetros via entrada padrão. Posteriormente, você deve imprimir os dados de cada instância criada utilizando o método `toString`, como realizado no laboratório anterior.

## 4 Tarefas

- Criação do projeto e classes.
- Programação dos métodos construtores das classes Cartao e Sala.
- Programação dos métodos get e set das classes Cartao e Sala.
- Programação dos métodos `toString` das classes Cartao e Sala.
- Programação dos demais métodos da classe Sala.
- Programação do método `main` e impressão de dados de 2 objetos Cartao e 2 Sala.

## 5 Questões

Sobre a atividade realizada, responda como comentário no início do código da classe que contém o `main`.

- É possível escrever uma classe sem escrever nenhum construtor? Por quê?
- Um método estático pode acessar uma variável (atributo) não estático da classe? Por quê?
- Um método não estático pode acessar uma variável (atributo) estático da classe? Por quê?

## **6 Submissão**

Para submeter a atividade utilize o Google Classroom. Salve os arquivos dessa atividade em um arquivo comprimido no formato .tar.gz ou .zip e nomeie-o **Lab2-000000.tar.gz ou Lab2-000000.zip** trocando '000000' pelo seu número de RA. Submeta o arquivo na seção correspondente para esse laboratório no Classroom da disciplina MC322.

### **Datas de entrega**

- Dia **12/10/2020** até às 21:00h