



## MC322 - Laboratório 1

**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 com o ambiente de desenvolvimento integrado (IDE, *Integrated Development Environment*) chamado Eclipse<sup>1</sup> e a linguagem de programação Java<sup>2</sup>.

## 3 Atividade

Nesta atividade o principal foco será a familiarização com o Eclipse e a programação de duas classes chamadas **Usuario** e **Perfil**. A primeira tarefa será configurar o ambiente com a criação de um novo projeto e de uma nova classe para então programar.

Os seguintes passos podem ser tomados para a criação do projeto:

1. Abra o Eclipse.
2. Crie um novo projeto (File -> New -> Java Project).
3. Digite o nome do projeto (ex: Lab1).
4. Na aba JRE escolha a última versão do JavaSE instalado na máquina (ex: JavaSE-14 ou outra versão).

---

<sup>1</sup><https://eclipse.org>

<sup>2</sup><https://www.java.com>

## 5. Clique em Finish.

Para criar uma nova classe faça:

1. Utilize a aba 'Package Explorer' que aparece do lado esquerdo da IDE.
2. Crie uma nova classe no projeto (Botão direito no projeto -> New -> Class).
3. Digite o nome da classe.
4. Programe a classe.

## 4 Classe Usuário

A classe Usuario será responsável por armazenar os dados únicos de acesso do usuário ao sistema de apoio ao aluno.

A classe Usuário deve possuir a seguinte estrutura:

- id (número inteiro)
- login (cadeia de caracteres - *String*)
- email (cadeia de caracteres - *String*)
- senha (cadeia de caracteres - *String*)
- dataAtivacao (data - *Calendar*)
- status (variável da lógica booleana - *boolean*)

O exemplo a seguir apresenta a declaração da classe Usuario com alguns de seus atributos. Note que todas as variáveis são declaradas como privadas (**private**). Note também a implementação dos métodos de acesso get() e set(). Esses métodos são comumente utilizados na linguagem Java para acessar os atributos dos objetos.

```
1 public class Usuario {  
2  
3     private int id;  
4     private String nome;  
5  
6     // Metodo construtor aqui  
7  
8     // Demais metodos aqui  
9     public int getId() {  
10         return id;  
11     }  
12  
13     public void setId(int id) {  
14         this.id = id;  
15     }  
16  
17 }
```

Usuario.java

Além disso, a classe Usuario deve conter um método construtor e este deve receber como argumentos os atributos para inicializar o objeto. Para ilustrar esse conceito melhor, veja o exemplo abaixo.

```
1 // — Dentro da classe Usuario  
2     public Usuario(int id, String nome ...) {  
3         this.id = id;  
4         this.nome = nome;  
5         // demais inicializacoes  
6     }  
7     public String getNome() {
```

```

8         return nome;
9     }
10    public void setNome(String nome) {
11        this.nome = nome;
12    }
13    // outros metodos ...

```

MetodoConstrutor.java

Também é necessário que a classe Usuario possua uma função **toString()** que devolve uma String contendo uma descrição geral dos atributos do usuário. Veja o exemplo abaixo:

```

1 @Override
2     public String toString() {
3         String out = getNome()+" (id: "+getID()+"\n";
4         out = out + "email: "+getEmail()+"\n";
5         out = out + "senha: "+getSenha()+"\n";
6         // Talvez haja uma melhor forma de apresentar esta informacao
7         out = out + "status: "+getStatus()+"\n";
8         return out;
9     }
10

```

toString.java

Observe que os métodos get e set devem ser utilizados para acessar o valor dos atributos. Deve existir um método get e set correspondente a cada atributo declarado. O formato do método **toString()** a ser implementado é livre, mas todos os atributos devem ser impressos.

Faça a implementação do método **construtor**, métodos **get()** e **set()** de todos os atributos e do método **toString()** para a classe **Usuario**.

## 5 Classe Perfil

Utilizando a classe Usuario como base, crie a classe Perfil e faça a implementação do método **construtor**, dos métodos **get()** e **set()** de todos os atributos e do método **toString()** para a classe **Perfil**.

A classe Perfil deve ter os seguintes atributos:

- sexo (caracter)
- dataNascimento (data - *Calendar*)
- cidade (cadeia de caracteres - *String*)
- estado (cadeia de caracteres - *String*)
- telefone (cadeia de caracteres - *String*)
- descricao (cadeia de caracteres - *String*)
- foto (cadeia de caracteres - *Strings*)

## 6 Classe Main

Para um programa Java funcionar é necessário que um método main, que serve de ponto de partida para o programa ser iniciado, seja criado. Crie uma nova classe através do Eclipse chamada Main e escolha a opção para gerar automaticamente o método main.

Na função main realize a instanciação de objetos do tipo Usuario e Perfil (pelo menos dois de cada) com valores de atributos quaisquer. Após instanciar os objetos, imprima seus dados utilizando o método System.out.println(). Veja o exemplo a seguir:

```

1 // — dentro da classe Main
2 public static void main(String[] args) {
3     // instanciando objetos
4     Usuario user1 = new Usuario(1, "Darth Vader", ...);
5     Usuario user2 = new Usuario(2, "Yoda", ...);
6     Perfil pf1 = new Perfil('f', new GregorianCalendar(2013,10,28), ...);
7
8     // impressao dos dados dos objetos
9     System.out.println("Usuario 1:\n"+user1);
10    // ...
11 }
```

print.java

Observe que as chamadas dos construtores está incompleta e você precisará adequá-las a sua declaração do construtor. Note que ao imprimir os dados dos objetos da classe Usuario ou Perfil, o método `toString()` que você implementou foi chamado implicitamente.

Após implementar as três classes, para executar o programa e ver o resultado clique no botão “Run” do Eclipse.

## 7 Tarefas

- Criação do projeto e classes.
- Programação dos métodos construtores das classes Usuario e Perfil.
- Programação dos métodos get e set das classes Usuario e Perfil.
- Programação dos métodos `toString` das classes Usuario e Perfil.
- Programação do método `main` e impressão de dados de 2 objetos Usuario e 2 perfil.

## 8 Questões

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

- Qual o impacto de se tentar mudar diretamente o valor do atributo `login` de um dos objetos `Usuario` no método `main`? Por quê?
- Como você trataria o problema levantado na questão anterior? Qual seriam os prós e os contras desta nova abordagem? Que conceito de Orientação a Objetos estaria sendo “quebrado” neste caso?

## 9 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 **Lab1-000000.tar.gz** ou **Lab1-000000.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 **28/09/2020** até às 21:00h