



**BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**Programação Orientada a Objetos II – 2s2025**

**Atividade 07/08/25 – Entrega pelo Classroom, hoje até 15h55**

**Classes e Objetos**

**Instruções**

- Além dos atributos e métodos especificados, implemente, em Java, também todos os *getters* e *setters*. Além disso, crie alguns objetos para testar sua implementação;
- Em um arquivo texto – README.txt -, informe o IDE ou o Editor que utilizou para a implementação e a forma de compilar e executar o programa;
- Faça um print da tela com o resultado do teste da execução;
- Faça o *upload* de um arquivo compactado (.zip ou .rar), no formulário da atividade no Classroom.
  - Nome do arquivo compactado: <seuprimeironome\_seusobrenome>-atividade0708.zip (.rar):
    - Exemplo: Fulado\_Tal-atividade0708.zip
  - Conteúdo do arquivo compactado:
    - os arquivos ou o projeto da implementação;
    - o arquivo README.txt;
    - o print do teste.

**Implementação – Médico e Hospital**

Implemente uma Classe Médico e uma Classe Hospital. Para isso, considere as informações a seguir e a representação UML para classes.

**Médico-<seuprimeironome> [exemplo: Medico-Fulano]**

Atributos: CRM (Ex: 156785), nome, especialidade (ex: pediatria), dataContratacao, salário. [para uso de data em Java, veja um código exemplo ao final deste texto]

Métodos:

- ✓ Construtores
- ✓ *Getters* e *Setters*
- ✓ Imprime : Exibe todos os dados de um médico, ou seja os valores de seus atributos.

## Hospital-<seuprimeironome> [exemplo: Hospital-Fulano]

Atributos: Nome, Endereço, Lista dos Médicos contratados (Objetos da classe Médico),  
sugestão: Utilize por exemplo um ArrayList

### Métodos:

- ✓ Construtores
- ✓ *Getters* e *Setters*
- ✓ *adicionarMedico* : Recebe como parâmetro um Objeto Médico e o adiciona na lista de contatos da Agenda.
- ✓ *excluirMedico* : Recebe como parâmetro o CRM de um médico e o exclui da lista de contratados.
- ✓ *exibirMedicosAnoContrato* : Dado um ano, exibe todos os médicos presentes na lista de contratados pelo Hospital após o ano, inclusive, especificado.
- ✓ *Imprime* : Exibe o CRM e nome de todos os médicos presentes na lista de contratados pelo Hospital.

Medico-<seuprimeironome>
private CRM: int private Nome: String private Especialidade: String private dataContratacao: LocalDate private Salario: float Medico (int CRM, String Nome, String Especialidade, LocalDate dataContratacao, float Salario) public void imprime ()

Hospital-<seuprimeironome>
private Nome: String private Endereco: String private MedicosContratados: ArrayList<Medico-<seuprimeironome>) Hospital() Hospital(String Nome, String Endereco) public void adicionarMedico(Medico-<seuprimeironome> médico) public void excluirMedico(int CRM) public void exibirMedicosAnoContrato(int ano) public void imprime()

## Exemplo de uso de data em Java

```
import java.time.LocalDate;
public class Date {
    public static void main(String[] args) {

        //Inserção da data manualmente
        LocalDate dataEmprestimo = LocalDate.parse("2023-08-21");
        System.out.println(dataEmprestimo);

        //Inserção da data atual do sistema
        dataEmprestimo = LocalDate.now();
        System.out.println(dataEmprestimo);

        //Adicionando dias, meses e anos
        LocalDate dataDevolucao = dataEmprestimo.plusDays(10);
        System.out.println(dataDevolucao);

        dataDevolucao = dataEmprestimo.plusMonths(2);
        System.out.println(dataDevolucao);

        dataDevolucao = dataEmprestimo.plusYears(4);
        System.out.println(dataDevolucao);

    }
}
```