

## Programação Orientada a Objetos - Lista 1

---

**Professor:** Daniel Bezerra

**Disciplina:** POO

**Atividade:** Métodos e Atributos

### Instruções

---

Zipar todos os seus .java em um único arquivo e submeter até o limite definido da atividade.

O nome dos arquivos .java devem ser q1.java, q2.java, ..., qN.java

### Descrição da Atividade

---

#### Questão 01

Uma locadora de filmes precisa de um sistema simples para gerenciar os filmes disponíveis.

Crie uma classe chamada `Filme`, que contenha:

- Atributos privados: `titulo` (String), `genero` (String) e `anoLancamento` (int).
- Um construtor que receba todos os atributos como parâmetro.
- Um segundo construtor (sobrecarga) que receba apenas o título e o gênero, atribuindo o ano de lançamento como 0 (indefinido).
- Métodos públicos getters e setters para acessar e modificar os atributos.
- Um método `toString()` que retorne uma string formatada com as informações do filme.

Crie uma classe chamada `Locadora`, que contenha:

- Um atributo privado `nome` (String).
- Um atributo privado que armazene um `ArrayList` `filmes`.
- Um construtor que inicialize o nome da locadora e a lista de filmes como vazia.
- Um método público `adicionarFilme(Filme filme)` que insere um novo filme na lista.

- Um método público `listarFilmes()` que imprime no console todos os filmes cadastrados (usando o `toString` de cada filme).

Em uma nova classe `LocadoraEstoque`, crie pelo menos três filmes usando os dois tipos de construtores e cadastre-os em uma instância da `Locadora`.

Por fim, chame o método `listarFilmes()` para exibir os dados.

## Questão 02

Siga os seguintes passos para criar a classe `Cliente`:

- Crie o programa `Cliente.java` e crie a classe `Cliente`
- Declare quatro atributos de `Cliente`: uma `String` `nome`, uma `String` `cpf`, um inteiro `saldo` e um inteiro `numConta`
- Crie dois construtores para essa classe. Um `Cliente(String nome, String cpf, int saldo, int numConta)` para cadastrar novos clientes com um saldo inicial e um `Cliente(String nome, String cpf, int numConta)` para criar clientes com conta zerada.
- Crie todos os getters e setters dos atributos
- Crie um método `public int extrato()` para retornar o saldo da conta de um `Cliente`.
- Crie um método `public void saque(int val)` para simular a retirada de um valor `val` do saldo.
- Crie um método `public void deposito(int val)` para simular o depósito de um valor `val` na conta do usuário.
- Crie um método `public String toString()` que retorne a seguinte `String`:

`Cliente: NOME_DO_CLIENTE\nCPF: CPF_DO_CLIENTE\nConta:`

`NUMERO_DA_CONTA\nSaldo: SALDO_DA_CONTA`

Siga os seguintes passos para criar a classe `Banco`

- Crie um objeto que seja um vetor de clientes `Cliente clientes[]` e que armazene dois objetos do tipo `Cliente`
- Crie o método `get` para o vetor `clientes[]`
- Crie o método `public Cliente[] cadastraClientes()` em `Banco` que irá criar e inicializar os Clientes `c1` e `c2`.
  - `c1` é um cliente que vai ser criado com as seguintes informações `Cliente("Alan Turing", "1289", 1500, 25896)`
  - `c2` é um cliente que vai ser criado com as seguintes informações `Cliente("Ada Love", "98765431", 12345)`

- Ainda no método `cadastraClientes()` insira `c1` na posição 0 do vetor de clientes e insira `c2` na posição 1;
- Crie o método `public String operacoes()` e implemente as seguintes operações com os clientes:
  - Saque R\$100 de cada cliente
  - Deposite R\$50 na conta de cada cliente
  - Verifique o extrato de todos os clientes, se algum deles estiver com saldo negativo, modifique o valor do saldo para R\$0
  - Altere o nome do `Cliente2` para "Ada Lovelace"
  - Altere o cadastro do CPF do `Cliente1` para "12345689"
- Por fim, ainda no método `operacoes()`, usando o método `toString()` de cada cliente, retorne uma string com as informações de todos os clientes do vetor `clientes[]` separados por uma quebra de linha. Exemplo:

```
Cliente: NOME_DO_CLIENTE1
CPF: CPF_DO_CLIENTE1
Conta: NUMERO_DA_CONTA1
Saldo: SALDO_DA_CONTA1
Cliente: NOME_DO_CLIENTE2
CPF: CPF_DO_CLIENTE2
Conta: NUMERO_DA_CONTA2
Saldo: SALDO_DA_CONTA2
```