

# Aula anterior

## Métodos de Ordenação

- Método Bolha (Bubble-Sort)
- Método Seleção (Selection-Sort)
- Método de Inserção (insertion-Sort)
- Método de Contagem (Counting-Sort)
- Método de Chave (Radix-Sort)
- Método Shell-Sort
- Método Quick-Sort
- Método Merge-Sort
- Método Heap-Sort

# Tipo Abstrato de Dados

---

Prof. Diego Silva Caldeira Rocha

---

## Exercício

- Mostre a execução do programa abaixo

```
...  
Cliente c1 = new Cliente(1, "aa");  
Cliente c2 = null;  
c2 = c1;  
c2 = null;  
c2 = c1.clone();  
...
```

Representação gráfica

c1 → ?

c2 → ?

Memória


Nome das variáveis

Endereços de memória

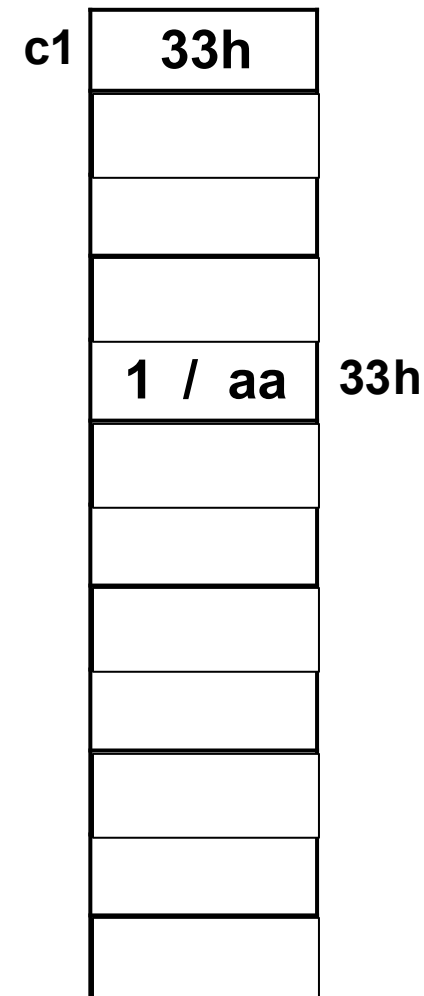
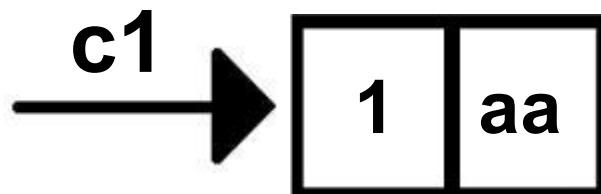
## Exercício

- Mostre a execução do programa abaixo

...

`Cliente c1 = new Cliente(1, "aa");``Cliente c2 = null;``c2 = c1;``c2 = null;``c2 = c1.clone();`

...



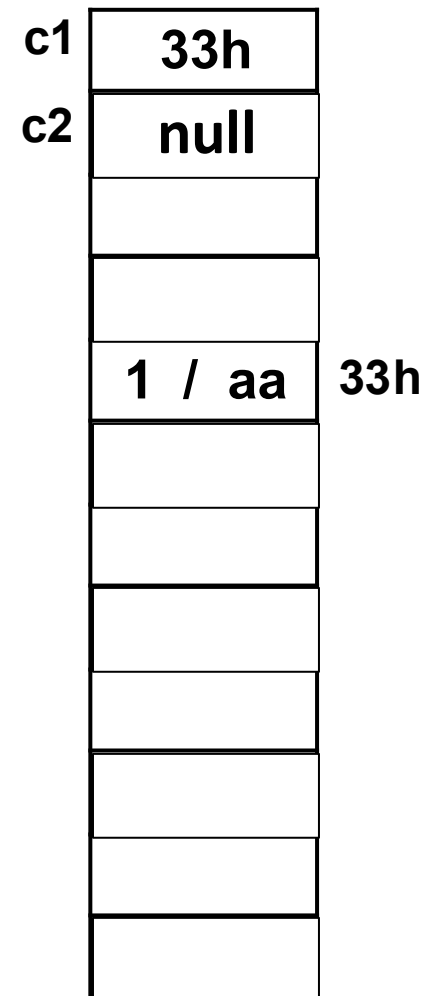
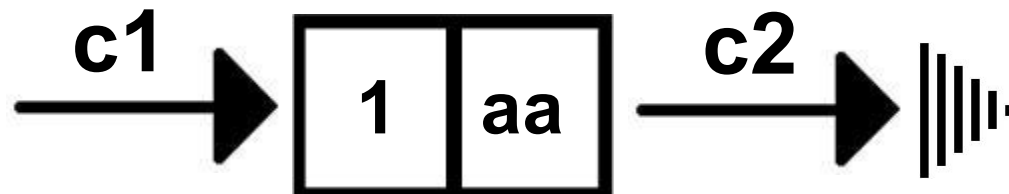
## Exercício

- Mostre a execução do programa abaixo

```

...
Cliente c1 = new Cliente(1, "aa");
Cliente c2 = null;
c2 = c1;
c2 = null;
c2 = c1.clone();
...

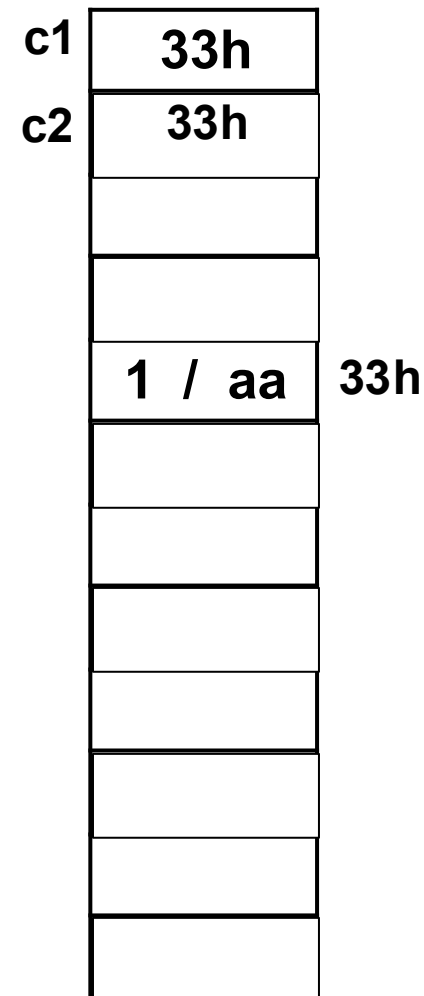
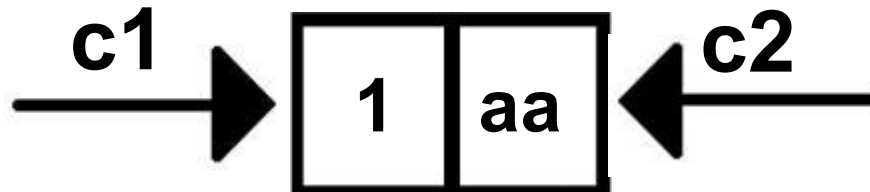
```



## Exercício

- Mostre a execução do programa abaixo

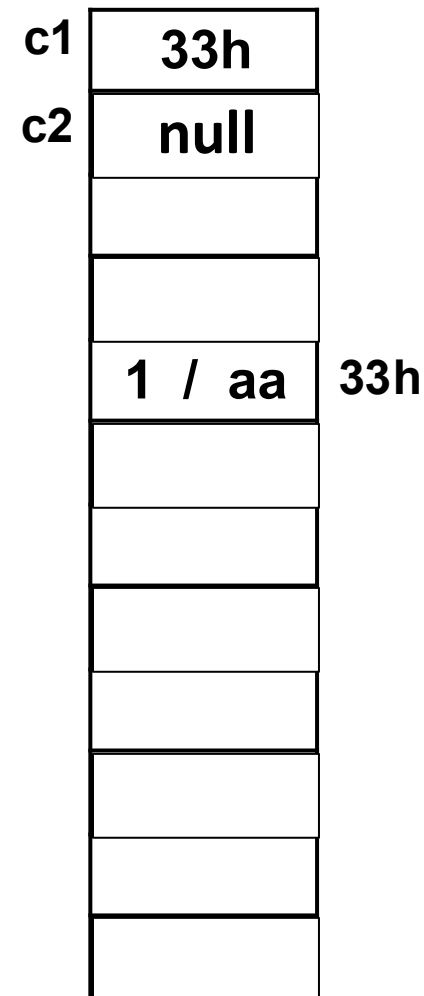
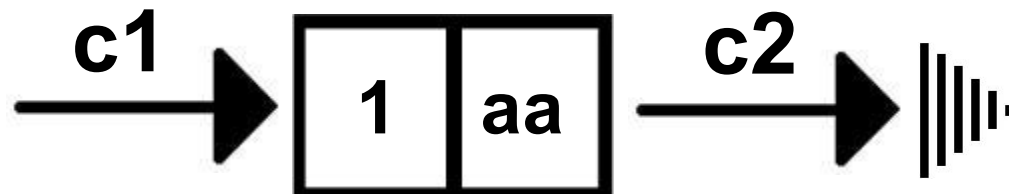
```
...  
Cliente c1 = new Cliente(1, "aa");  
Cliente c2 = null;  
c2 = c1;  
c2 = null;  
c2 = c1.clone();  
...
```



## Exercício

- Mostre a execução do programa abaixo

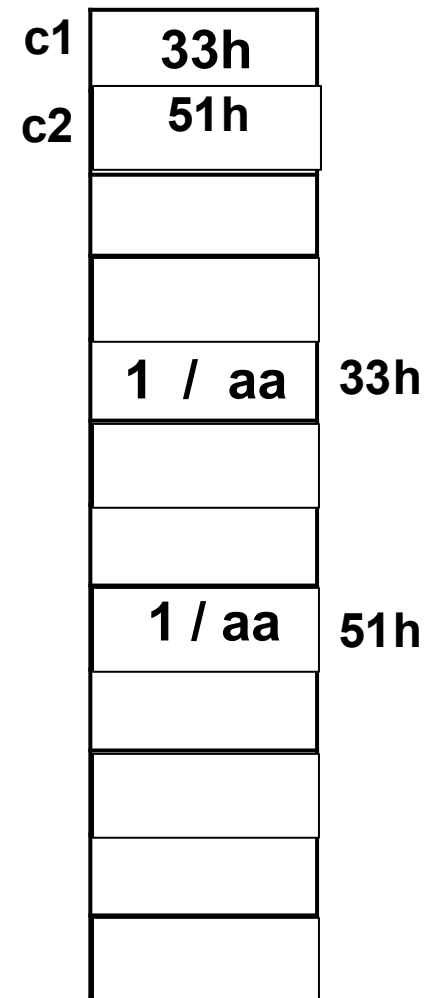
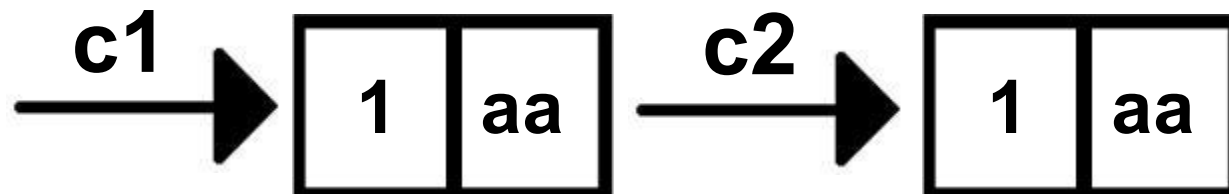
```
...  
Cliente c1 = new Cliente(1, "aa");  
Cliente c2 = null;  
c2 = c1;  
c2 = null;  
c2 = c1.clone();  
...
```



## Exercício

- Mostre a execução do programa abaixo

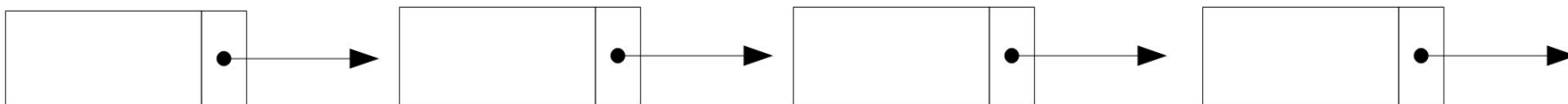
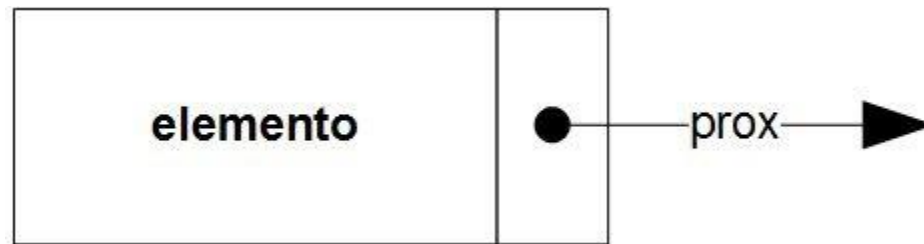
```
...  
Cliente c1 = new Cliente(1, "aa");  
Cliente c2 = null;  
c2 = c1;  
c2 = null;  
c2 = c1.clone();  
...
```





## Criação de uma célula

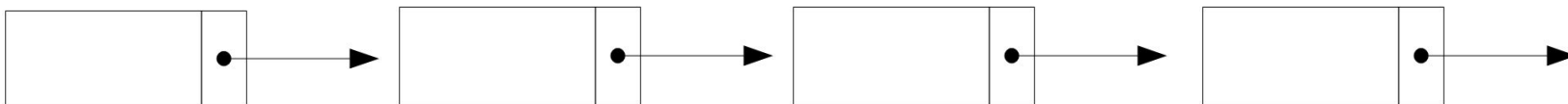
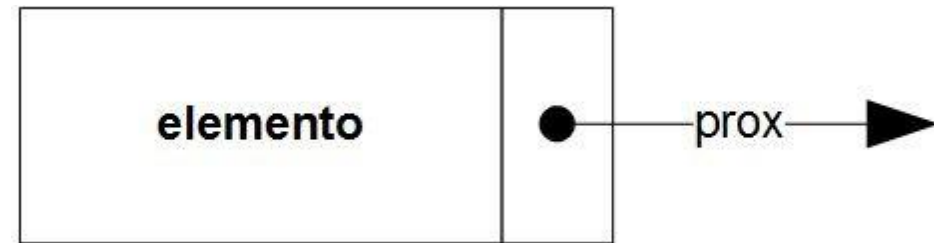
- Criação de uma classe célula contendo os atributos elemento (inteiro) e prox (apontador para outra célula)



## Exercício

- Crie uma classe célula contendo os atributos elemento (inteiro) e prox (apontador para outra célula)

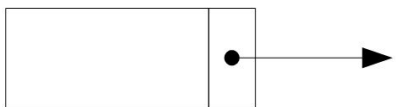
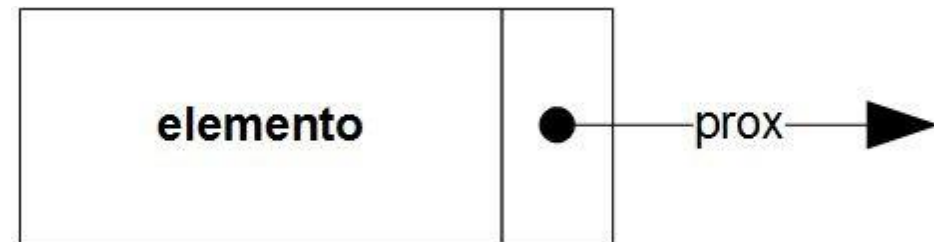
```
class Celula {  
    public int elemento;  
    public Celula prox;  
    public Celula() {  
        this(0);  
    }  
    public Celula (int x) {  
        this.elemento = x;  
        this.prox = null;  
    }  
}
```



## Exercício

- Crie uma classe célula contendo os atributos elemento (inteiro) e prox (apontador para outra célula)

```
class Celula {  
    public int elemento;  
    public Celula prox;  
    public Celula() {  
        this(0);  
    }  
    public Celula (int x) {  
        this.elemento = x;  
        this.prox = null;  
    }  
}
```

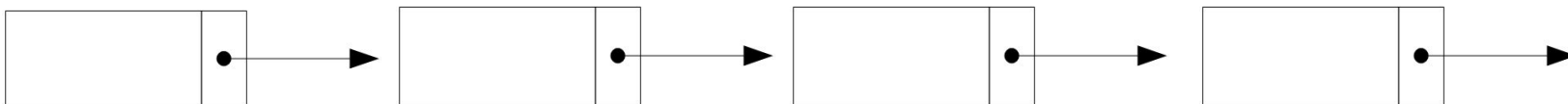
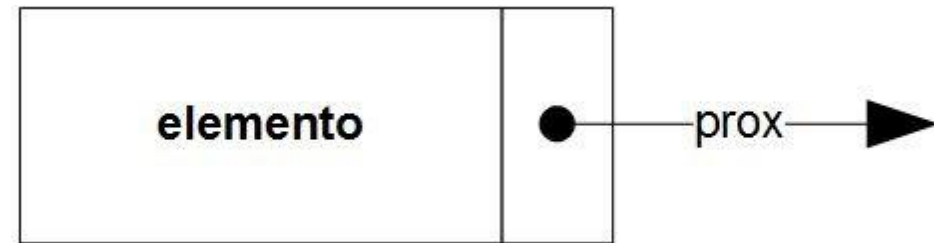


**A boa prática da orientação por objetos recomenda que todos os atributos de uma classe sejam privados e tenham os métodos públicos de get e set!!!**

## Exercício

- Crie uma classe célula contendo os atributos elemento (inteiro) e prox (apontador para outra célula)

```
class Celula {  
    public int elemento;  
    public Celula prox;  
    public Celula() {  
        this(0);  
    }  
    public Celula (int x) {  
        this.elemento = x;  
        this.prox = null;  
    }  
}
```



## Exercício

- Mostre o que acontece se outra classe tiver o comando

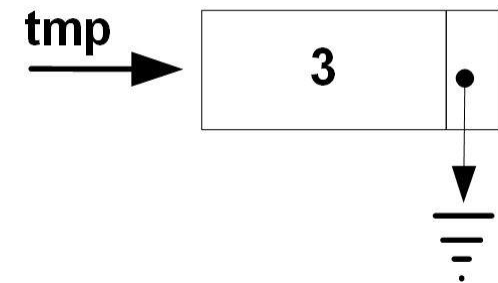
*Celula tmp = new Celula(3).*

```
class Celula {  
    public int elemento;  
    public Celula prox;  
    public Celula() {  
        this(0);  
    }  
    public Celula (int x) {  
        this.elemento = x;  
        this.prox = null;  
    }  
}
```

## Exercício

- Mostre o que acontece se outra classe tiver o comando *Celula tmp = new Celula(3).*

```
class Celula {  
    public int elemento;  
    public Celula prox;  
    public Celula() {  
        this(0);  
    }  
    public Celula (int x) {  
        this.elemento = x;  
        this.prox = null;  
    }  
}
```



## Exercício

- Mostre o que acontece se outra classe tiver o comando

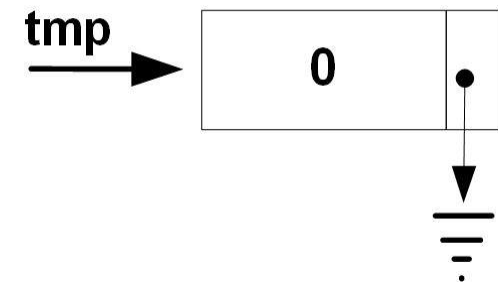
*Celula tmp = new Celula();*

```
class Celula {  
    public int elemento;  
    public Celula prox;  
    public Celula() {  
        this(0);  
    }  
    public Celula (int x) {  
        this.elemento = x;  
        this.prox = null;  
    }  
}
```

## Exercício

- Mostre o que acontece se outra classe tiver o comando *Celula tmp = new Celula()*.

```
class Celula {  
    public int elemento;  
    public Celula prox;  
    public Celula() {  
        this(0);  
    }  
    public Celula (int x) {  
        this.elemento = x;  
        this.prox = null;  
    }  
}
```





## Célula Genérico

*Celula tmp = new Celula<int>().*

```
class Celula <E> {  
    public E elemento;  
    public Celula <E> prox;  
    public Celula() {  
  
    }  
    public Celula (E x) {  
        this.elemento = x;  
        this.prox = null;  
    }  
}
```

# Próxima aula

Tipo abstrato de dado FILA