

# Unidade IV: Tipos Abstratos de Dados Flexíveis - Introdução



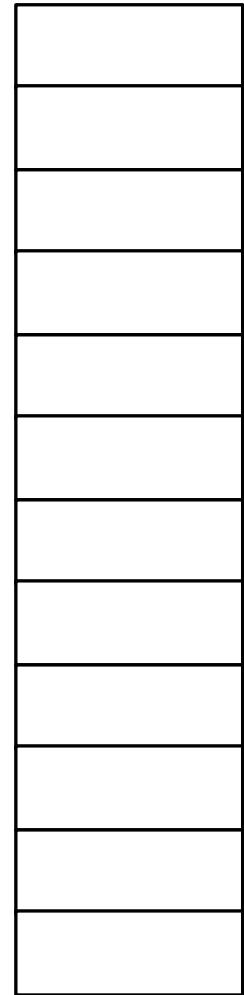
Instituto de Ciências Exatas e Informática  
Departamento de Ciência da Computação

# Exercício

- Mostre a execução do programa abaixo

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

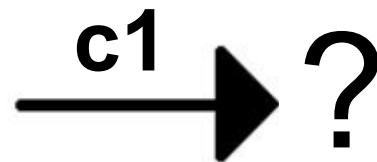
Memória



Nome das variáveis

Endereços de memória

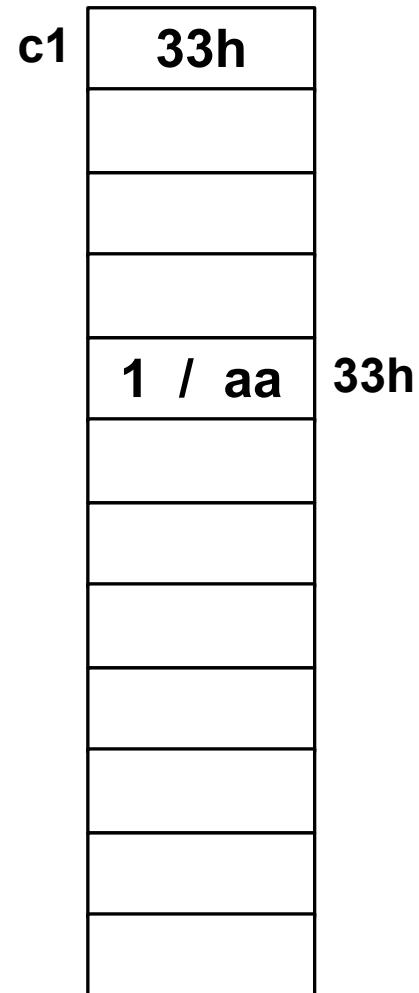
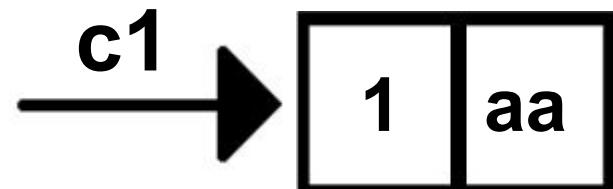
Representação gráfica



## 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;
```

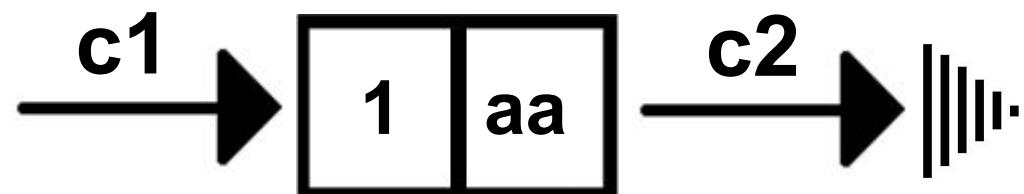
```
c2 = c1;
```

```
c2 = null;
```

```
c2 = c1.clone();
```

```
...
```

c1	33h
c2	null
1 / aa	33h

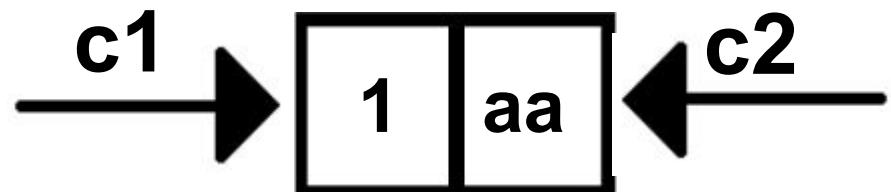


## Exercício

- Mostre a execução do programa abaixo

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

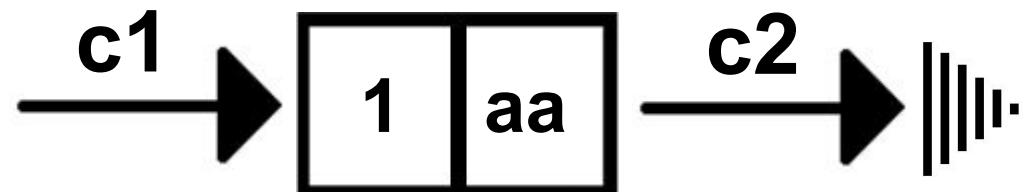
c1	33h
c2	33h
1 / aa	33h



## Exercício

- Mostre a execução do programa abaixo

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

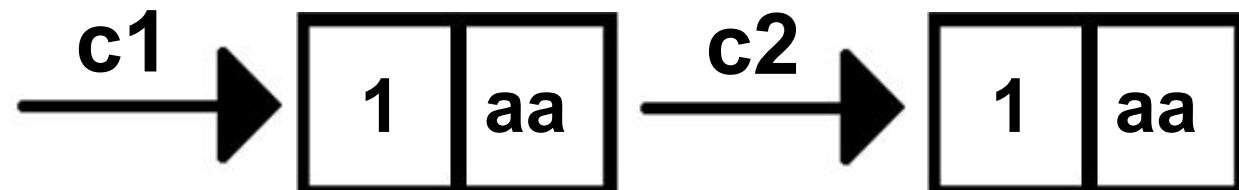


<code>c1</code>	33h
<code>c2</code>	null
	1 / aa
	33h

## Exercício

- Mostre a execução do programa abaixo

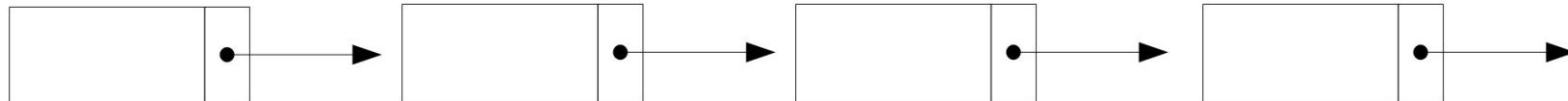
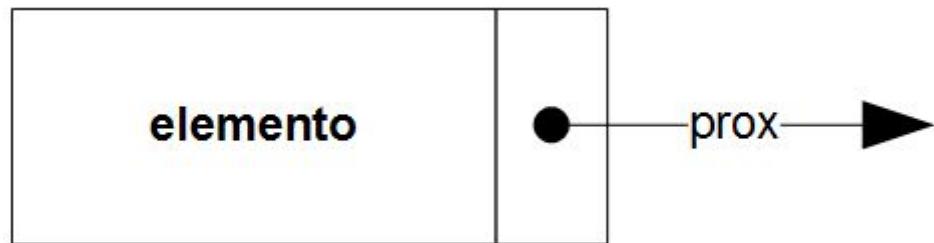
```
...
Cliente c1 = new Cliente(1, "aa");
Cliente c2 = null;
c2 = c1;
c2 = null;
c2 = c1.clone();
...
```



<code>c1</code>	<code>33h</code>	
<code>c2</code>	<code>51h</code>	
	<code>1 / aa</code>	<code>33h</code>
	<code>1 / aa</code>	<code>51h</code>

# Exercício

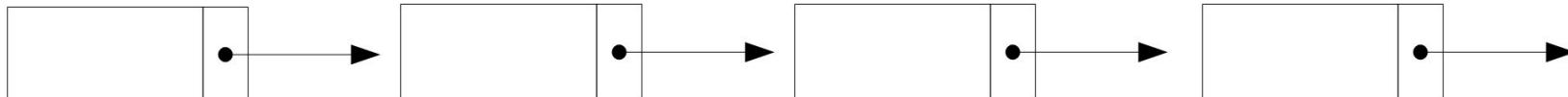
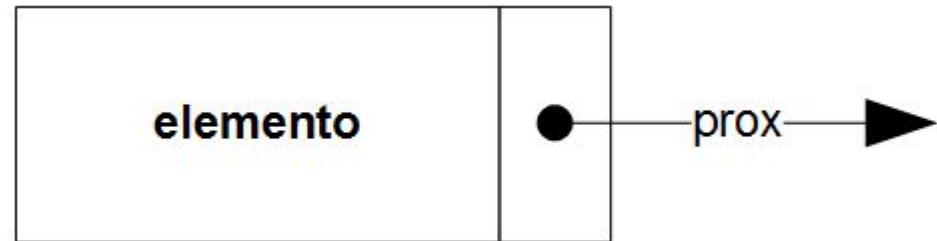
- Crie uma classe célula contendo os atributos **elemento** (inteiro) e **prox** (apontador para outra célula)



## Exercício

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

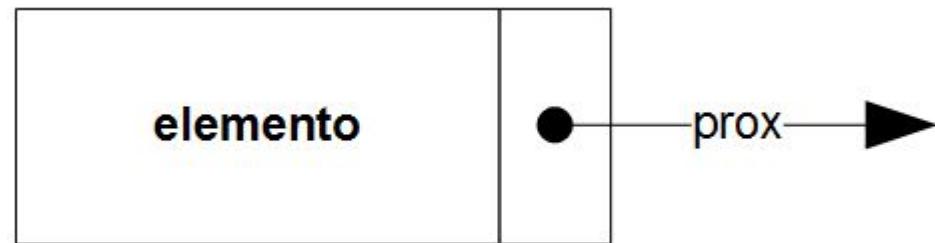
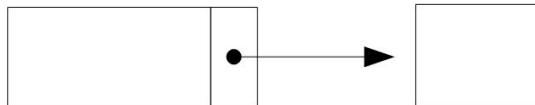
```
class Celula {  
    public int elemento;  
    public Celula prox;  
    public Celula() {  
        this(0);  
    }  
    public Celula (int x) {  
        this.elemento = x;  
        this.prox = null;  
    }  
}
```



## 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;  
    }  
}
```

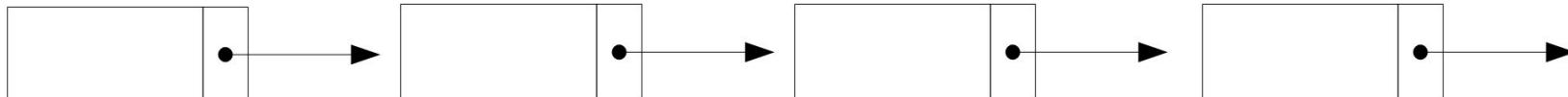
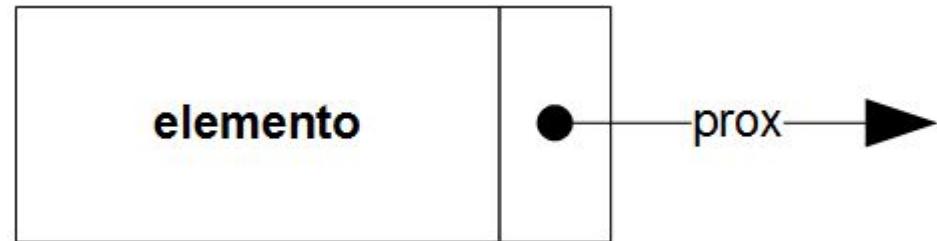


A boa prática da orientação por objetos recomenda que todos os atributos de uma classe sejam privados e tenham os métodos públicos de get e set!!!

## 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

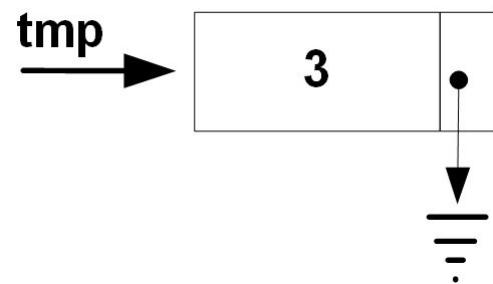
*Celula tmp = new Celula(3).*

```
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;  
    }  
}
```

