

Unidade V:

Estruturas de Dados Básicas

com Alocação Flexível - Introdução

Prof. Max do Val Machado



PUC Minas

Instituto de Ciências Exatas e Informática
Curso 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();  
...
```

Representação gráfica



Memória	
Nome das variáveis	Endereços de memória

Exercício

- Mostre a execução do programa abaixo

...

Cliente c1 = **new** Cliente(1, "aa");

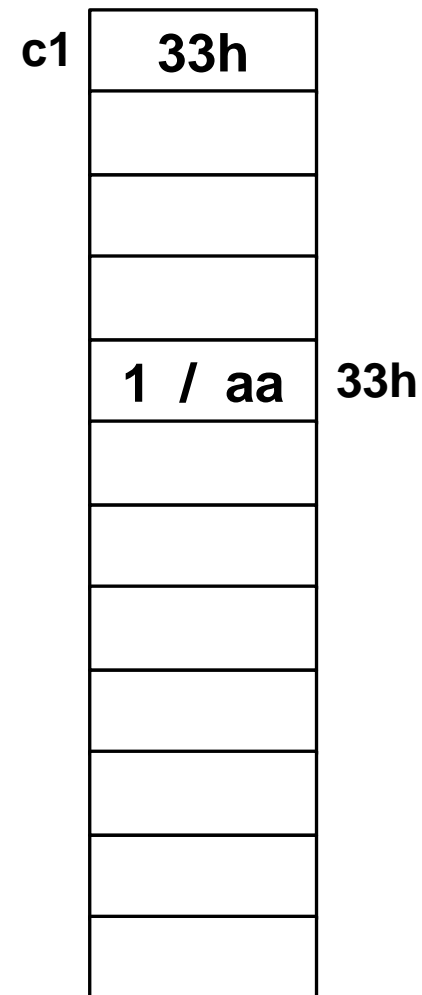
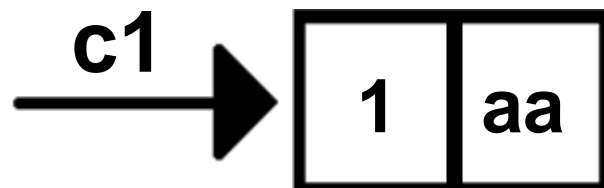
Cliente c2 = **null**;

c2 = c1;

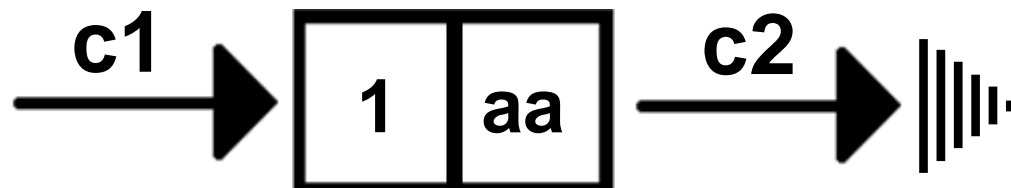
c2 = **null**;

c2 = c1.clone();

...



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

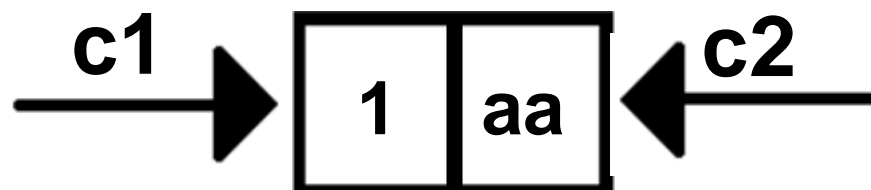
Algoritmos e Estruturas de Dados II (4)

# 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 |
|    |        |     |
|    |        |     |
|    |        |     |
|    |        |     |
|    |        |     |
|    |        |     |
|    |        |     |

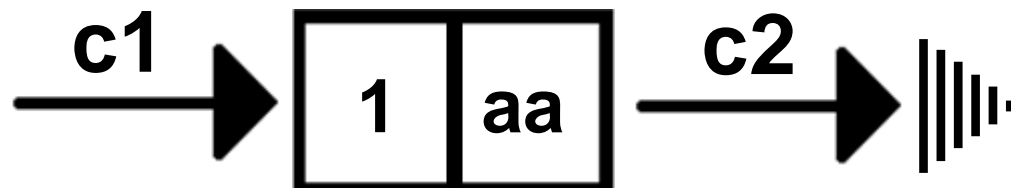


# Exercício

- Mostre a execução do programa abaixo

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

|    |        |     |
|----|--------|-----|
| c1 | 33h    |     |
| c2 | null   |     |
|    |        |     |
|    |        |     |
|    | 1 / aa | 33h |
|    |        |     |
|    |        |     |
|    |        |     |
|    |        |     |
|    |        |     |
|    |        |     |
|    |        |     |

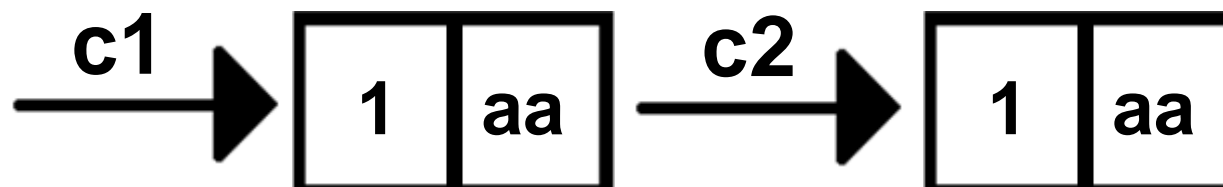


# Exercício

- Mostre a execução do programa abaixo

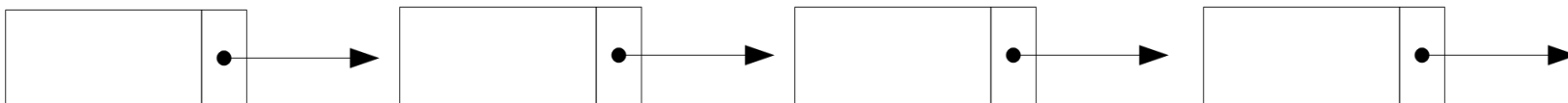
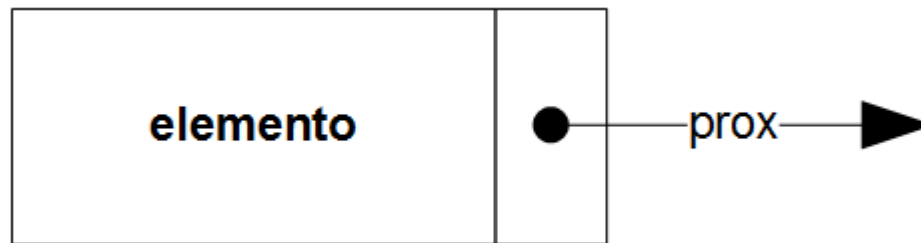
```
...
Cliente c1 = new Cliente(1, "aa");
Cliente c2 = null;
c2 = c1;
c2 = null;
c2 = c1.clone();
...
```

|    |        |     |
|----|--------|-----|
| c1 | 33h    |     |
| c2 | 51h    |     |
|    |        |     |
|    |        |     |
|    | 1 / aa | 33h |
|    |        |     |
|    |        |     |
|    | 1 / aa | 51h |
|    |        |     |
|    |        |     |
|    |        |     |
|    |        |     |



# Exercício

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

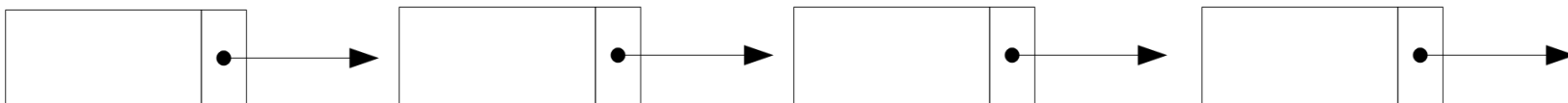
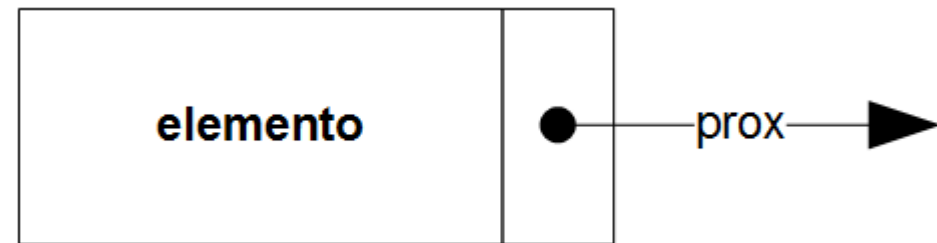




# Exercício

- 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

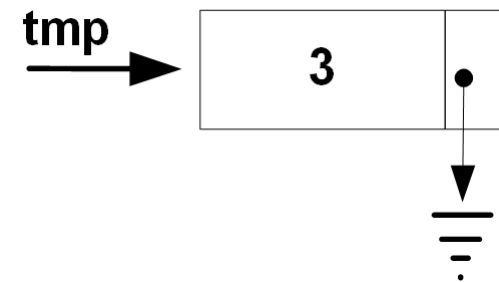
- Mostre o que acontece se outra classe tiver o comando *Celula tmp = new Celula(3).*

```
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;
 }
}
```

