

INE5408 – Estruturas de Dados

Semestre 2008/1
Prof. Leandro J. Komosinski

ALOCAÇÃO DINÂMICA DE MEMÓRIA

Memória de Computador

1 byte = 8 bits (ex.: 00100110)

1 KB = 1024 bytes

1 MB = 1024 KB

1 GB = 1024 MB

“Meu computador tem 4 GB de memória”.

$4 \text{ GB} = 4 * 1 \text{ GB}$

$4 * 1 \text{ GB} = 4 * 1024 \text{ MB} = 4096 * 1 \text{ MB}$

$4096 * 1 \text{ MB} = 4096 * 1024 \text{ KB} = 4.194.304 * 1 \text{ KB}$

$4.194.304 * 1 \text{ KB} = 4.194.304 * 1024 \text{ bytes}$

$4.194.304 * 1024 \text{ bytes} = 4.294.967.296 \text{ bytes}$

Memória Lógica de Computador

A memória do computador pode ser entendida, do ponto de vista lógico, como sendo um array.

“Meu computador tem 4 GB de memória”.



Endereço de Memória

Índice do array.

Variáveis em um Programa

```
graph TD; A[Variáveis em um Programa] --> B[Para o Programador]; A --> C[Para o Computador];
```

Para o Programador

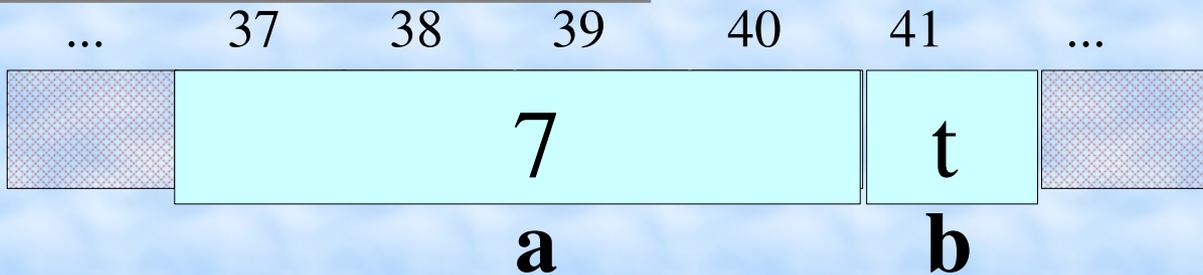
Um símbolo que representa uma informação.

Para o Computador

N bytes da memória reservados para representar algum dado.

```
...  
public void m() {  
    int a;  
    char b;  
  
    a = 7;  
    b = 't';  
}  
...
```

Em Java, uma variável do tipo int ocupa 4 bytes da memória. Uma variável do tipo char ocupa 1 byte da memória.



Toda variável ocupa alguma quantidade de bytes na memória.

Tipos de Variáveis

Quase todas as linguagens de programação permitem usar dois tipos de variáveis: **estáticas** e **dinâmicas**.

Variáveis Estáticas

O dado é armazenado **diretamente** na área de memória reservada para a variável.

Variáveis Dinâmicas

O programador deve alocar (reservar) uma outra área de memória para armazenar o dado. **A variável armazena o endereço da área que foi alocada.**

```

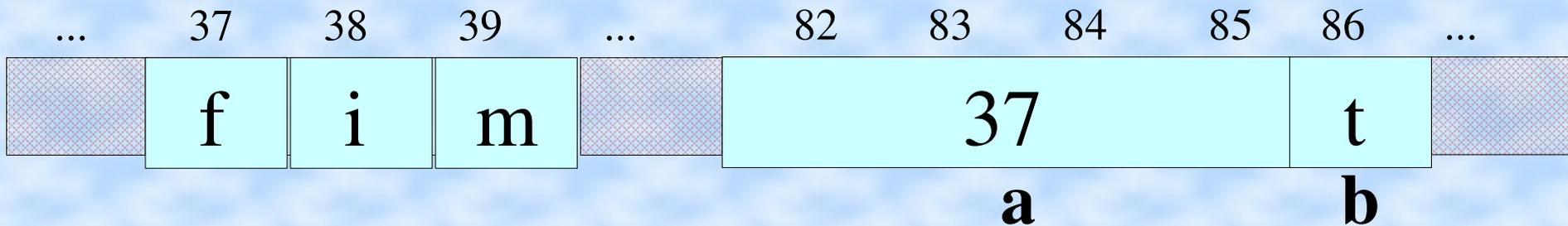
...
public void m() {
    char[] a;
    char b;

    a = new char [3];
    a[0] = 'f';
    a[1] = 'i';
    a[2] = 'm';
    b = 't';
}
...

```

a é variável dinâmica
b é variável estática

Variáveis dinâmicas também são chamadas de **ponteiros** (*pointers*) ou **apontadores**.



Tipos de Variáveis em Java

```
graph TD; A[Tipos de Variáveis em Java] --> B[Tipo Primitivo]; A --> C[Tipo Referência];
```

Em Java existem **dois tipos** de variáveis: **primitivo** e **referência**.

Tipo Primitivo

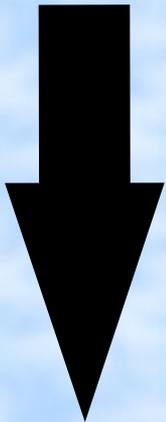
Variáveis do tipo primitivo são **estáticas** e representam números, caracter e booleano.

Tipo Referência

Variáveis do tipo referência são **dinâmicas** e representam objetos, classes, interfaces e arrays.

Alocação de Memória em Java

A **Máquina Virtual Java (MVJ)** é quem se encarrega de encontrar espaço livre na memória e reservar o número de bytes necessários para o tipo de dado que se quer armazenar.



Em outras linguagens, como C e C++, o programador pode saber, escolher e manipular o endereço onde os dados serão armazenados.

Para o programador Java, não interessa em qual endereço um dado está armazenado.

Alocação Dinâmica Memória

É quando cabe ao programador definir em que momento, durante a execução do programa, será alocado pelo sistema operacional (em Java, pela MVJ) uma área de memória grande o suficiente para representar um dado.

Uma variável dinâmica (ponteiro) armazena endereço de memória. Logo, esta variável só terá utilidade para o programador quando ela apontar para uma outra área da memória que aí sim conterá um dado útil.

Alocação Dinâmica em Java

`new <tipo>`

```
...  
int[] b;  
int tam;  
X p;  
  
tam = 5;  
b = new int [tam];  
p = new X();  
...
```

```
...  
X p;  
  
p = m();  
...  
public X m() {  
    return new X();  
}
```

```
...  
String s;  
int[] v;  
s = "teste";  
  
v = new int[] {3, 70};  
...
```

Alocação Dinâmica em Java - Exemplo 1

```
public class X {  
  
    public int v1;  
    public char v2;  
}
```

```
int a;  
char b;  
X p;
```

MEMÓRIA



a



b



p

Alocação Dinâmica em Java - Exemplo 1

```
public class X {  
  
    public int v1;  
    public char v2;  
}
```

```
int a;  
char b;  
X p;  
  
a = 50;
```

MEMÓRIA

50

a

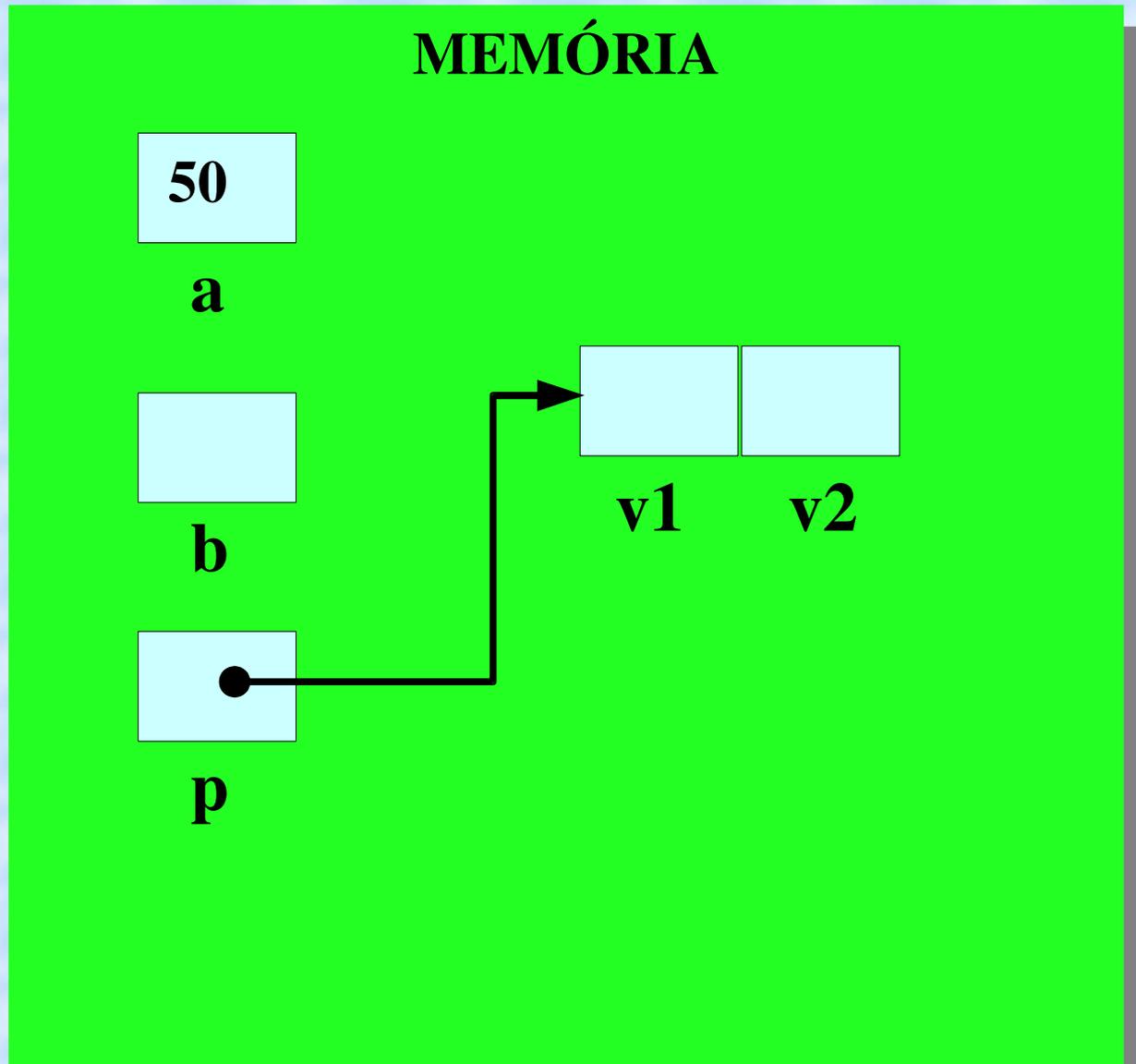
b

p

Alocação Dinâmica em Java - Exemplo 1

```
public class X {  
  
    public int v1;  
    public char v2;  
  
}
```

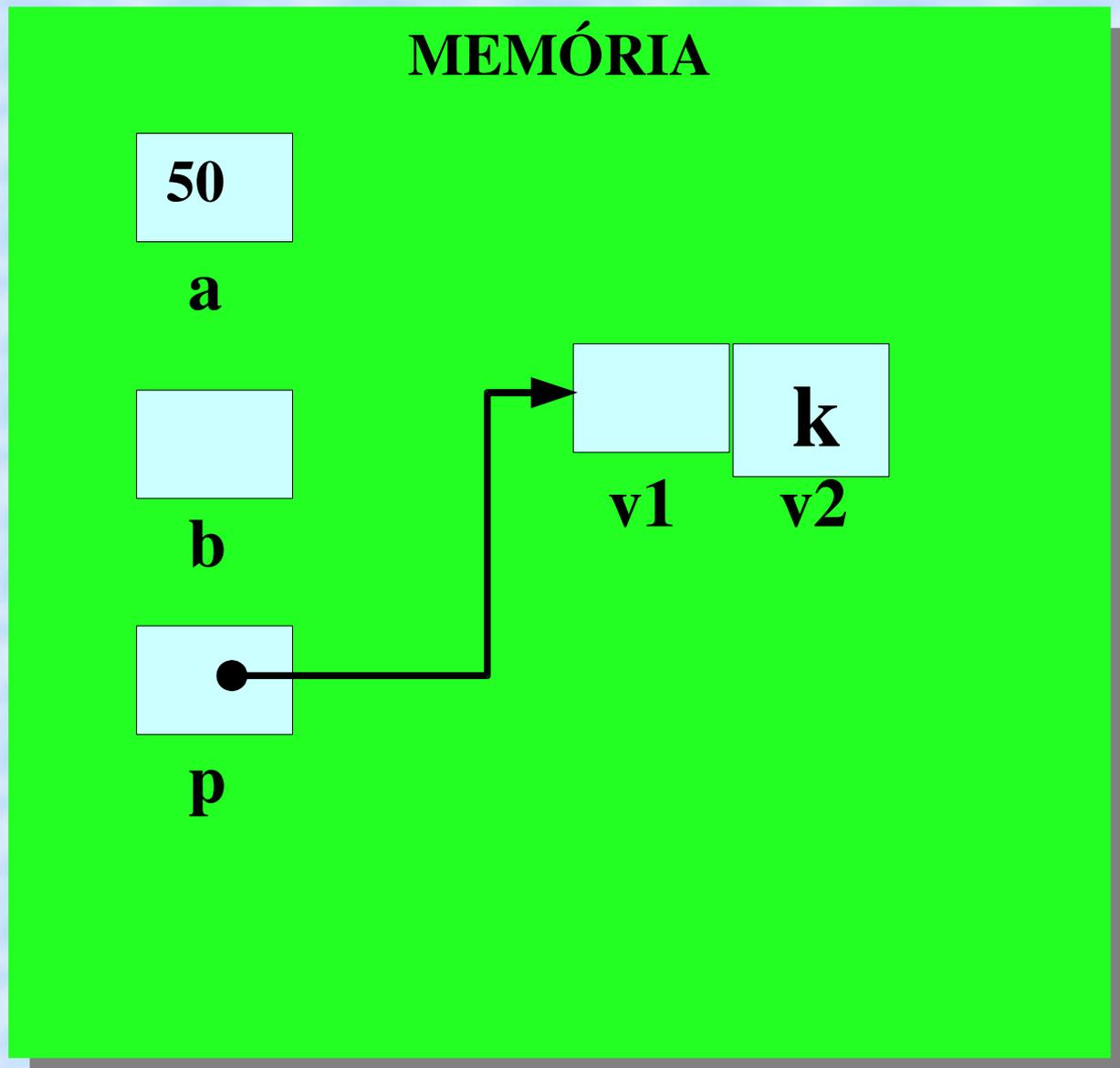
```
int a;  
char b;  
X p;  
  
a = 50;  
p = new X();
```



Alocação Dinâmica em Java - Exemplo 1

```
public class X {  
  
    public int v1;  
    public char v2;  
  
}
```

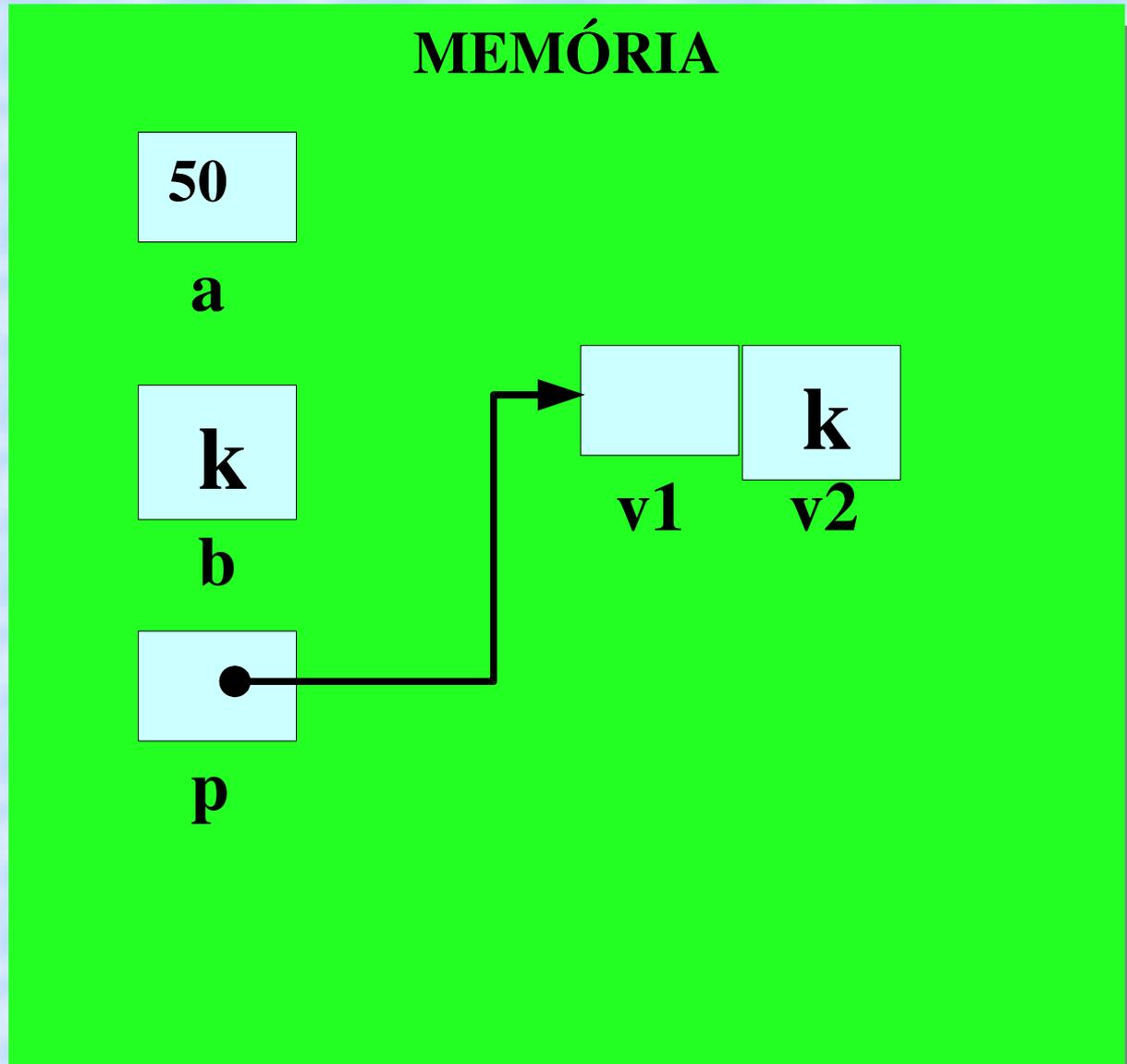
```
int a;  
char b;  
X p;  
  
a = 50;  
p = new X();  
p.v2 = 'k';
```



Alocação Dinâmica em Java - Exemplo 1

```
public class X {  
  
    public int v1;  
    public char v2;  
}
```

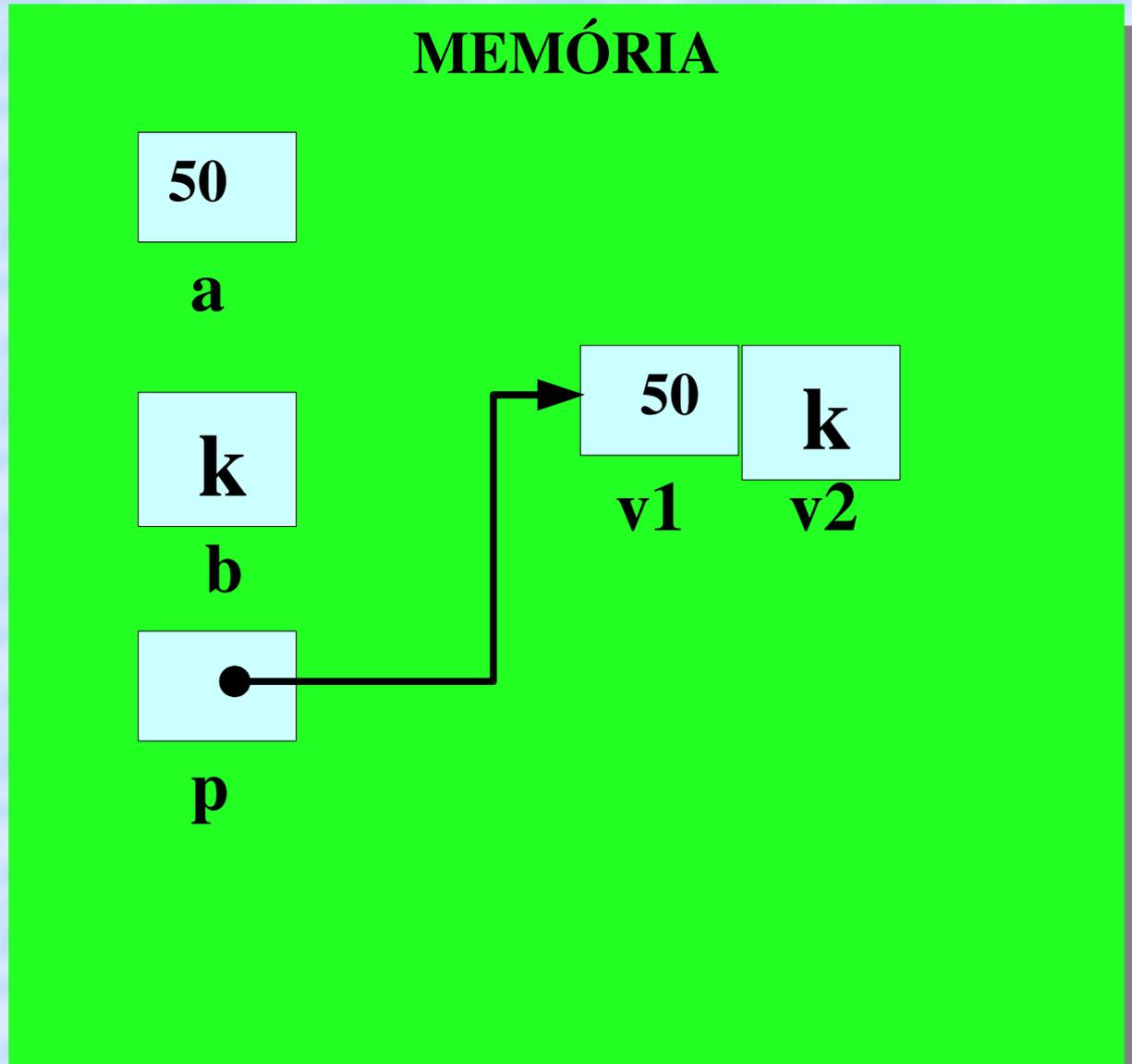
```
int a;  
char b;  
X p;  
  
a = 50;  
p = new X();  
p.v2 = 'k';  
b = p.v2;
```



Alocação Dinâmica em Java - Exemplo 1

```
public class X {  
  
    public int v1;  
    public char v2;  
  
}
```

```
int a;  
char b;  
X p;  
a = 50;  
p = new X();  
p.v2 = 'k';  
b = p.v2;  
p.v1 = a;
```



Alocação Dinâmica em Java - Exemplo 2

```
public class X {  
    public int v1;  
    public char v2;  
}
```

```
X o1, o2, o3;
```

MEMÓRIA



o1



o2



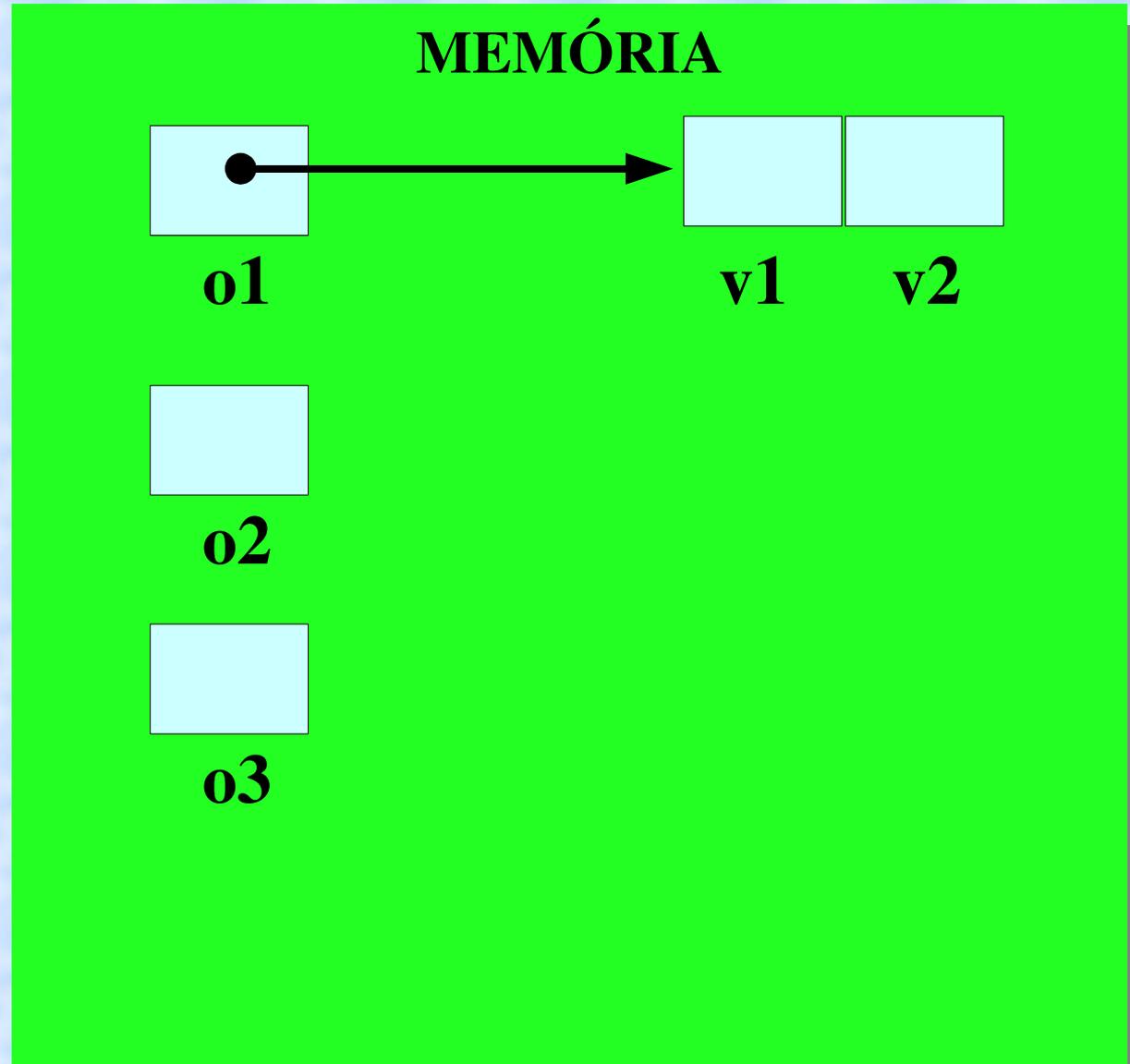
o3

Alocação Dinâmica em Java - Exemplo 2

```
public class X {  
    public int v1;  
    public char v2;  
}
```

```
X o1, o2, o3;
```

```
o1 = new X();
```

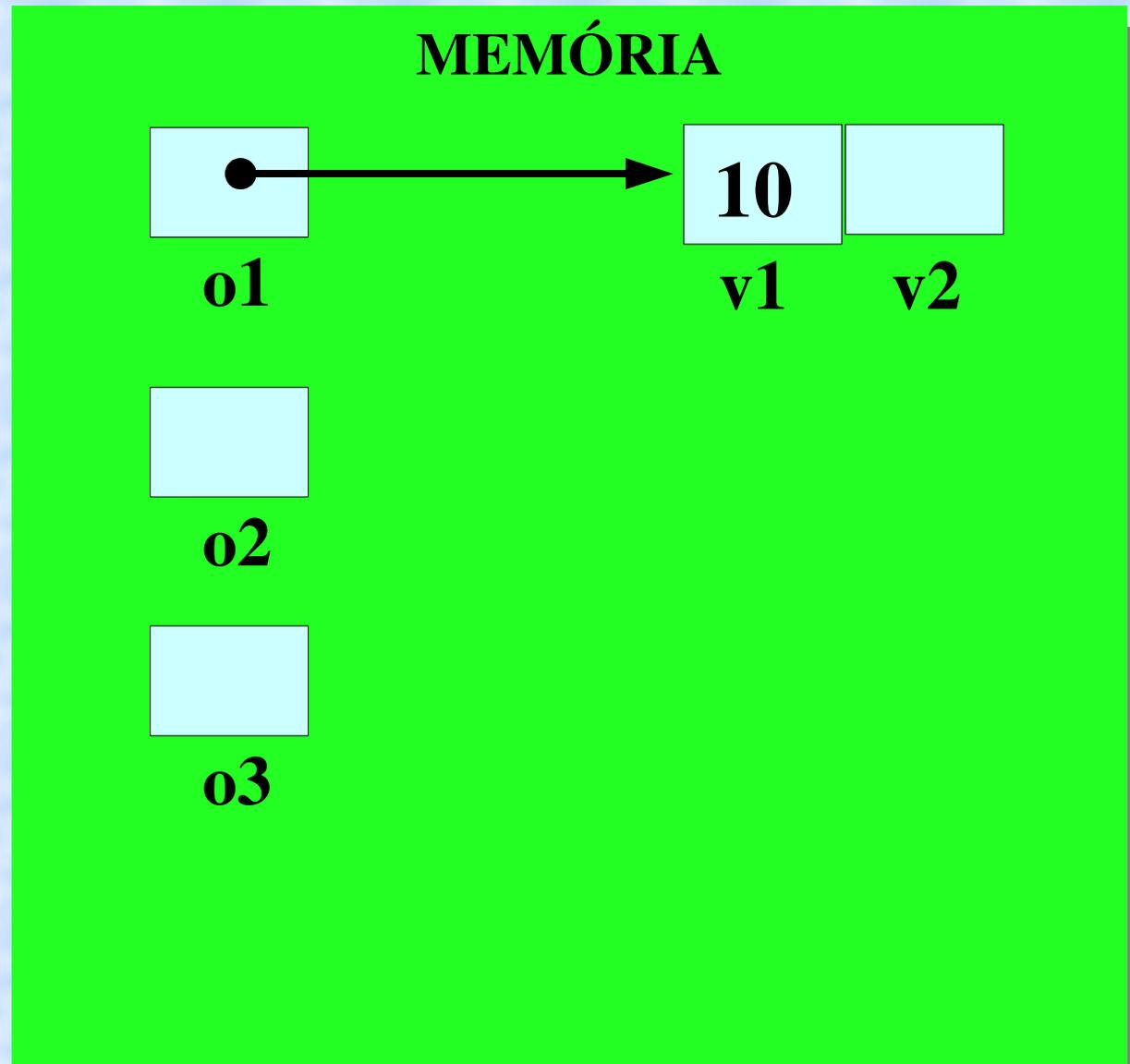


Alocação Dinâmica em Java - Exemplo 2

```
public class X {  
    public int v1;  
    public char v2;  
}
```

```
X o1, o2, o3;
```

```
o1 = new X();  
o1.v1 = 10;
```



Alocação Dinâmica em Java - Exemplo 2

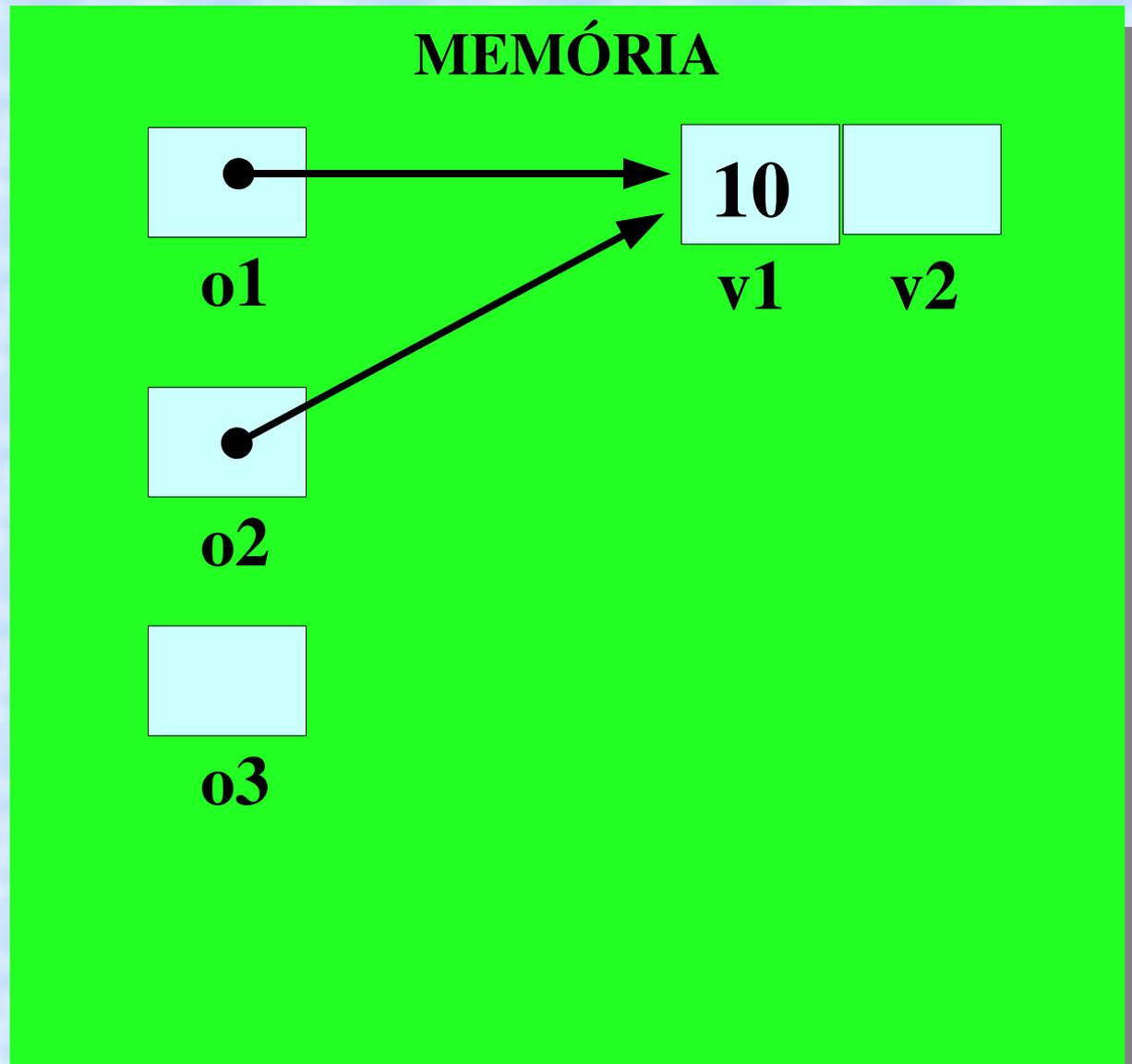
```
public class X {  
    public int v1;  
    public char v2;  
}
```

```
X o1, o2, o3;
```

```
o1 = new X();
```

```
o1.v1 = 10;
```

```
o2 = o1;
```



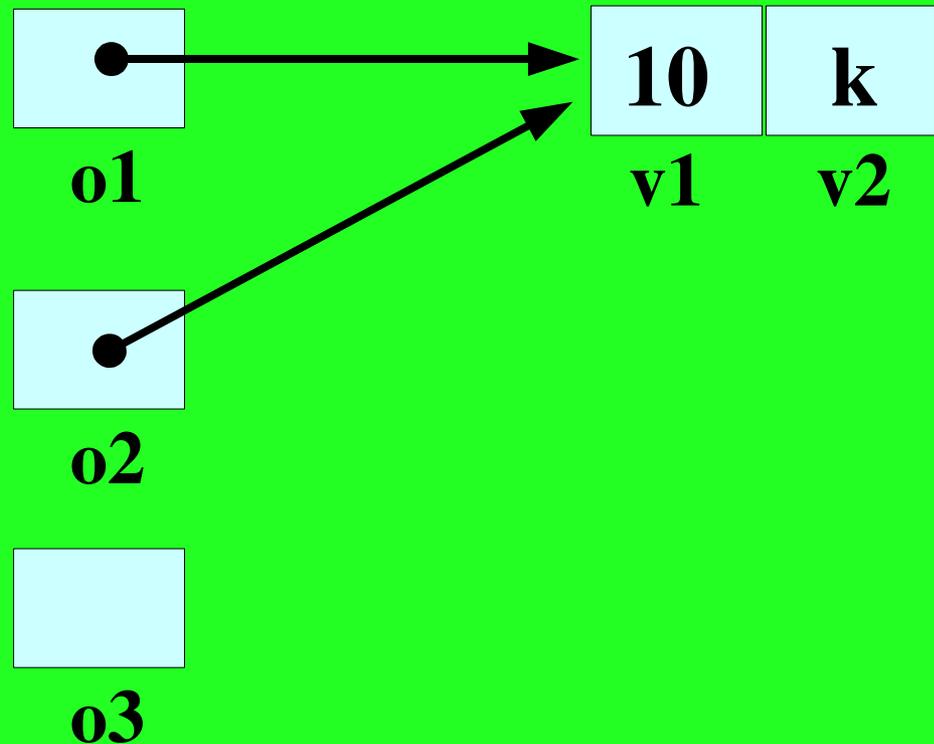
Alocação Dinâmica em Java - Exemplo 2

```
public class X {  
    public int v1;  
    public char v2;  
}
```

```
X o1, o2, o3;
```

```
o1 = new X();  
o1.v1 = 10;  
o2 = o1;  
o2.v2 = 'k';
```

MEMÓRIA



Alocação Dinâmica em Java - Exemplo 2

```
public class X {  
    public int v1;  
    public char v2;  
}
```

```
X o1, o2, o3;
```

```
o1 = new X();
```

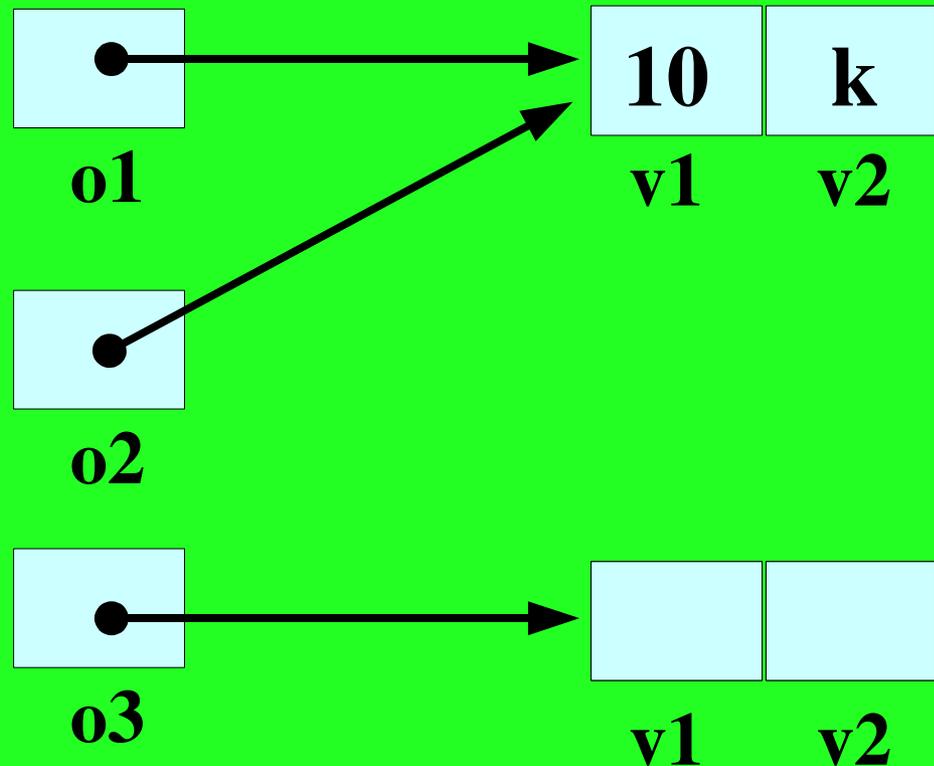
```
o1.v1 = 10;
```

```
o2 = o1;
```

```
o2.v2 = 'k';
```

```
o3 = new X();
```

MEMÓRIA



Alocação Dinâmica em Java - Exemplo 2

```
public class X {  
    public int v1;  
    public char v2;  
}
```

```
X o1, o2, o3;
```

```
o1 = new X();
```

```
o1.v1 = 10;
```

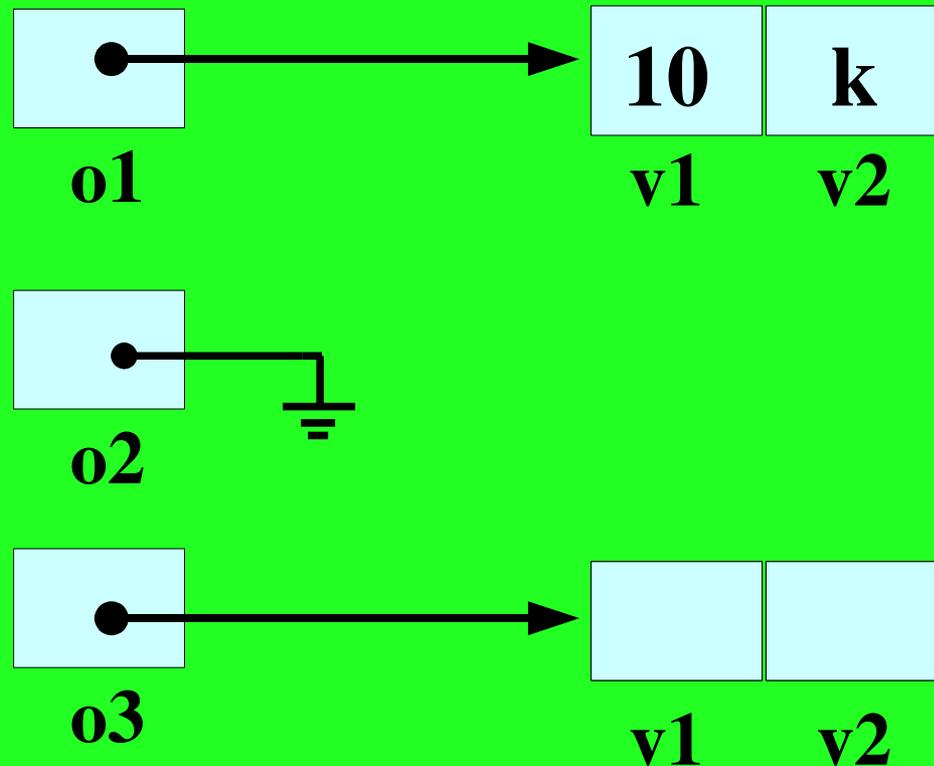
```
o2 = o1;
```

```
o2.v2 = 'k';
```

```
o3 = new X();
```

```
o2 = null;
```

MEMÓRIA



Alocação Dinâmica em Java - Exemplo 2

```
public class X {  
    public int v1;  
    public char v2;  
}
```

```
X o1, o2, o3;
```

```
o1 = new X();
```

```
o1.v1 = 10;
```

```
o2 = o1;
```

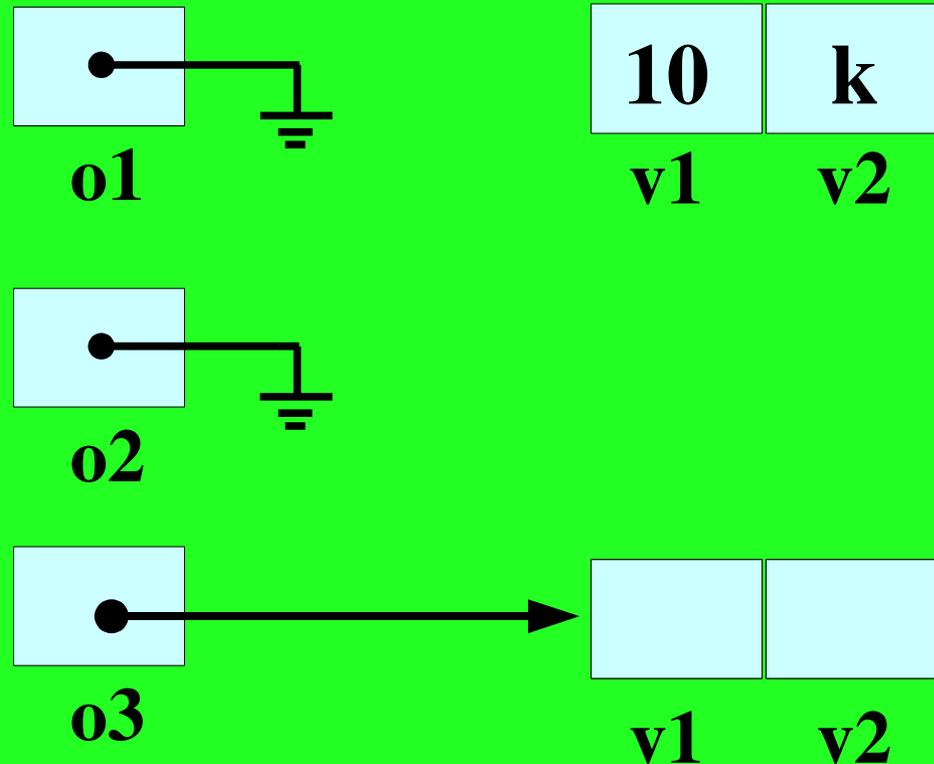
```
o2.v2 = 'k';
```

```
o3 = new X();
```

```
o2 = null;
```

```
o1 = null;
```

MEMORIA



Alocação Dinâmica em Java - Exemplo 2

```
public class X {  
    public int v1;  
    public char v2;  
}
```

```
X o1, o2, o3;
```

```
o1 = new X();
```

```
o1.v1 = 10;
```

```
o2 = o1;
```

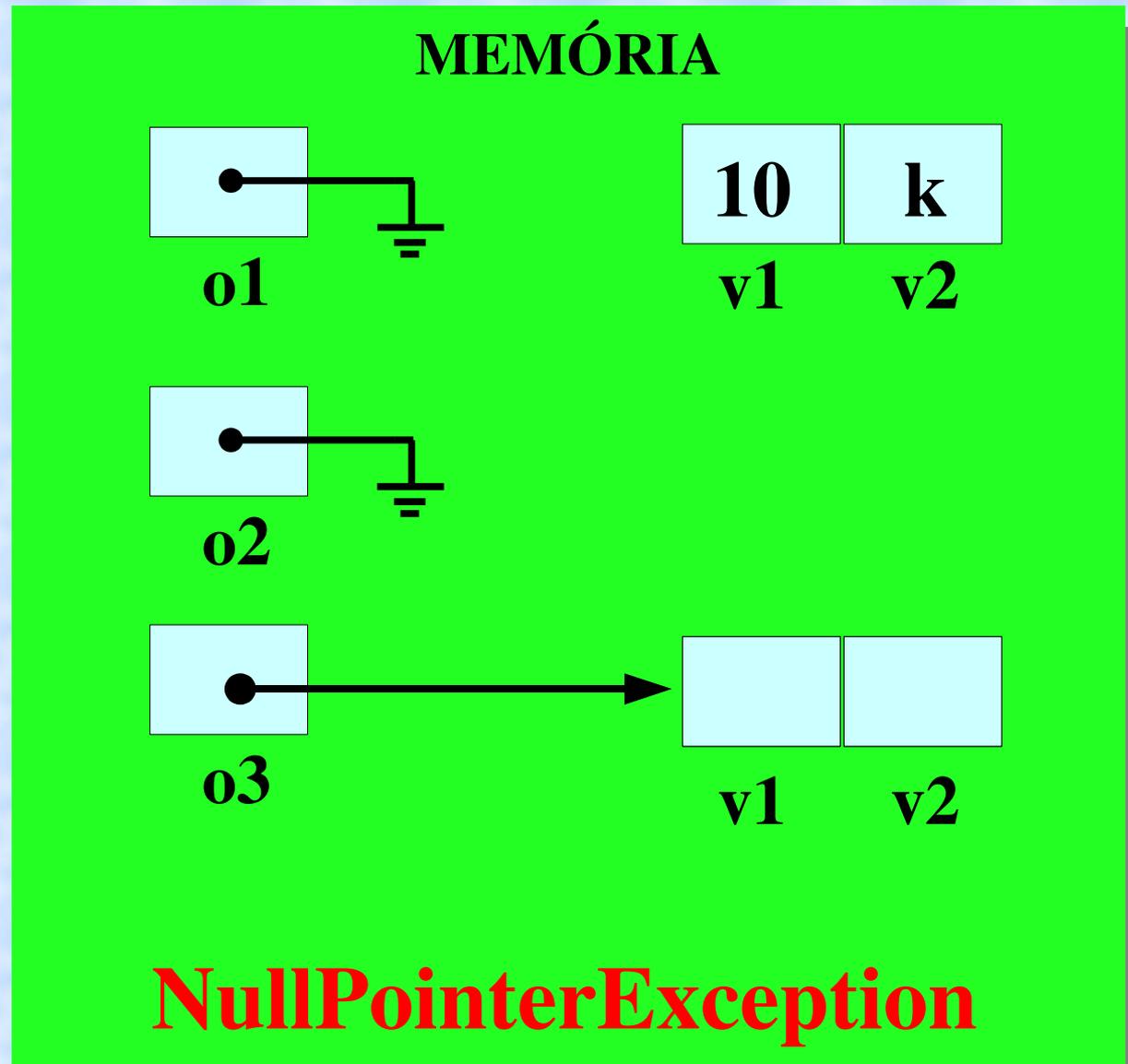
```
o2.v2 = 'k';
```

```
o3 = new X();
```

```
o2 = null;
```

```
o1 = null;
```

```
o3.v1 = o1.v1;
```



Fim