

Apontadores – Era uma vez...

Ana



Rua do Beco, nº10

João



Rua da Tavessa, nº23

NOME	TELEFONE	MORADA
Ana	789 45 61	Rua do Beco, nº10
Joao	456 78 12	Rua da Travessa, nº23

Apontadores

```
int x = 5, y = 10;
```

quando falamos de **x**, falamos do seu valor interno, ou seja, **5**

```
int Ana = 789 45 61
```

```
int Joao = 456 78 12
```

quando falamos de **Joao**, falamos do seu valor interno, ou seja, **456 78 12**

Apontadores

NOME	TELEFONE	MORADA
Ana	789 45 61	Rua do Beco, nº10
Joao	456 78 12	Rua da Travessa, nº23

Os endereços correspondem à morada da casa de cada um.

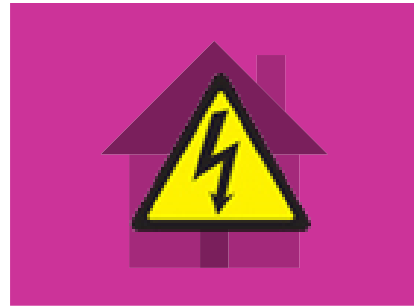
Em C++ existe o operador **&** que permite saber qual o endereço de uma variável.

&Ana = Rua do Beco, nº10

&Joao = Rua da Travessa, nº23

Apontadores

Baltazar



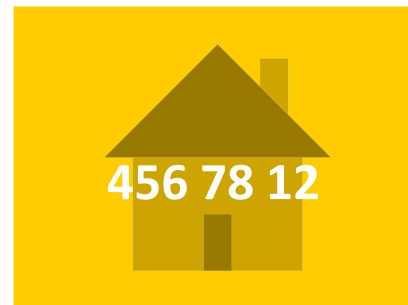
Rua do Perigo, nº7

Ana



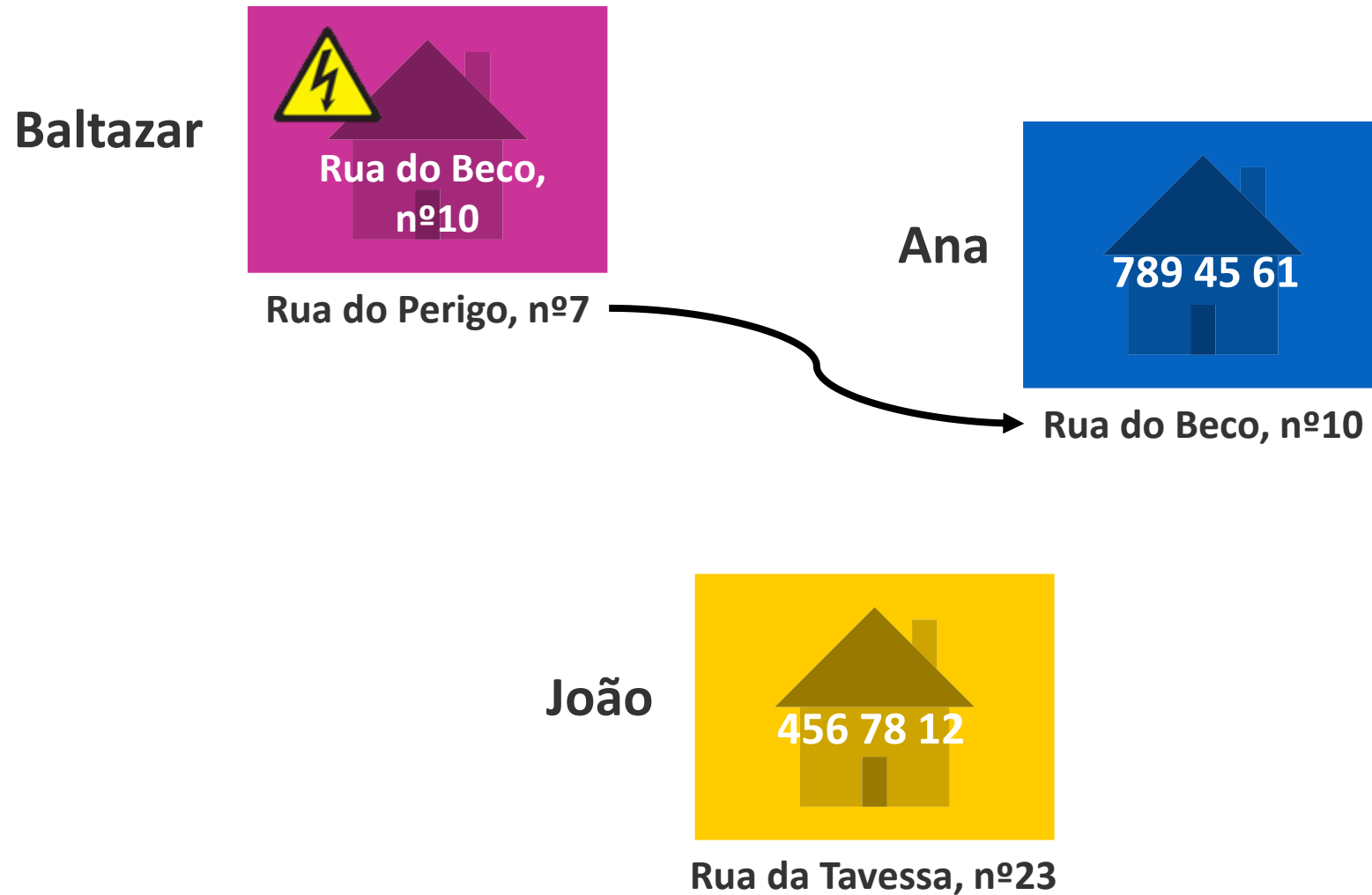
Rua do Beco, nº10

João



Rua da Tavessa, nº23

Apontadores



Apontadores

A casa de Baltazar é declarada como uma casa que contém o endereço de outra casa.

Isto é, **Baltazar aponta para uma casa com telefone.**

Um **apontador** é uma variável que aponta sempre para outra variável de um determinado tipo.

Se uma variável **x** contém o endereço de **y**, é possível aceder ao valor de **y** a partir de **x**.

Coloca-se um asterisco antes da variável **x** → ***x**



OPERADOR
DE INDIRECÇÃO

int * Baltazar

Apontadores

EXPRESSÃO	VALOR
Ana	789 45 61
&Ana	Rua do Beco, nº10
Joao	456 78 12
&Joao	Rua da Travessa, nº23
Baltazar	Rua do Beco, nº10
&Baltazar	Rua do Perigo, nº7
*Baltazar	789 45 61

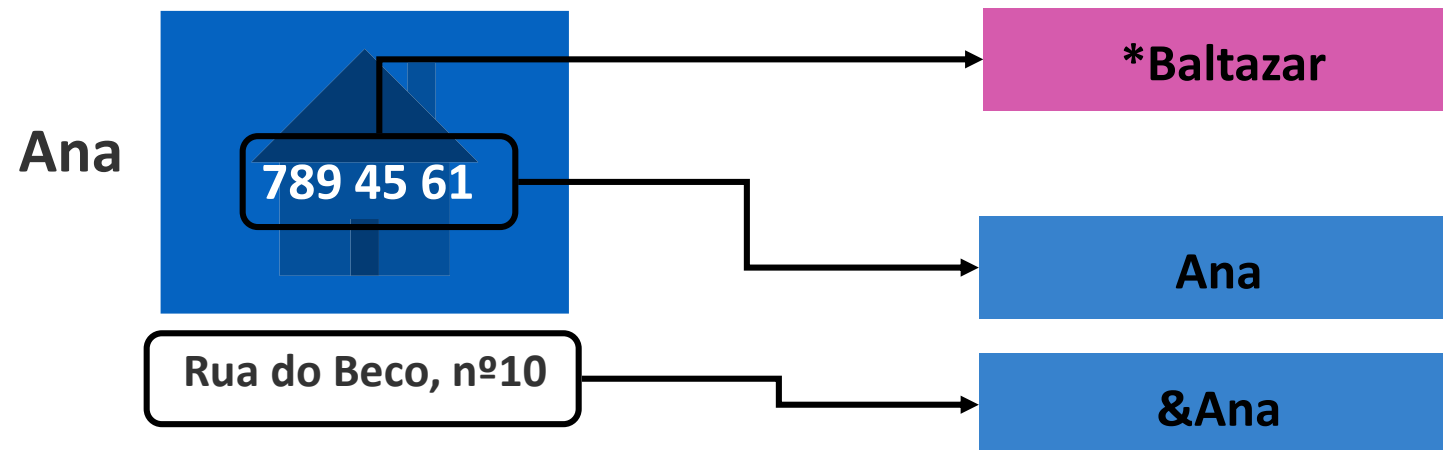
Apontadores

O nome de cada pessoa é uma forma bem mais simples de representar a posição de cada uma.

X: Representa o valor que está dentro da casa chamada X.

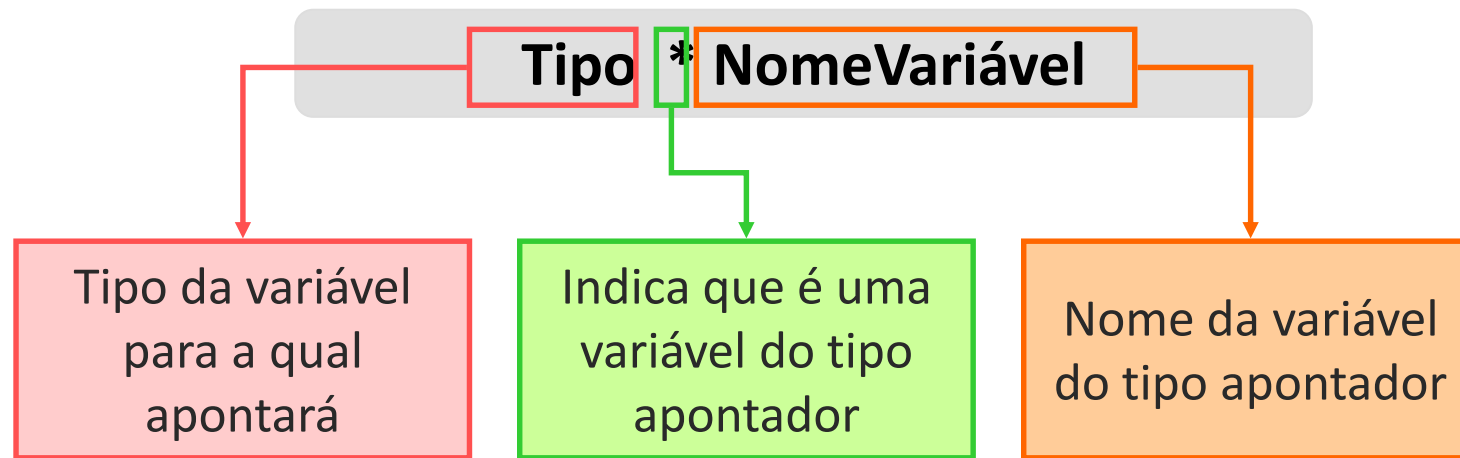
&X: Representa o endereço onde a casa X se situa.

***Y**: Representa o valor que está colocado no endereço armazenado em Y.



Declaração de Apontadores

A sintaxe genérica da declaração de um apontador é:



Apontadores - Exemplos

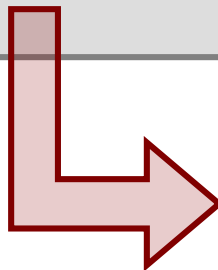
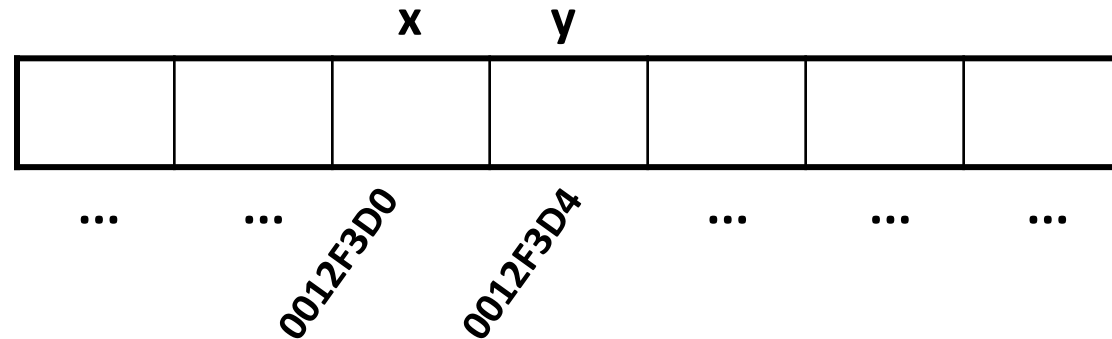
```
#include <iostream>
using namespace std;

void main() {
    int x, y;

    cout<<sizeof(x)<<endl;
    cout<<sizeof(y)<<endl;

    cout<<&x<<endl;
    cout<<&y<<endl;
}
```

MEMÓRIA RAM

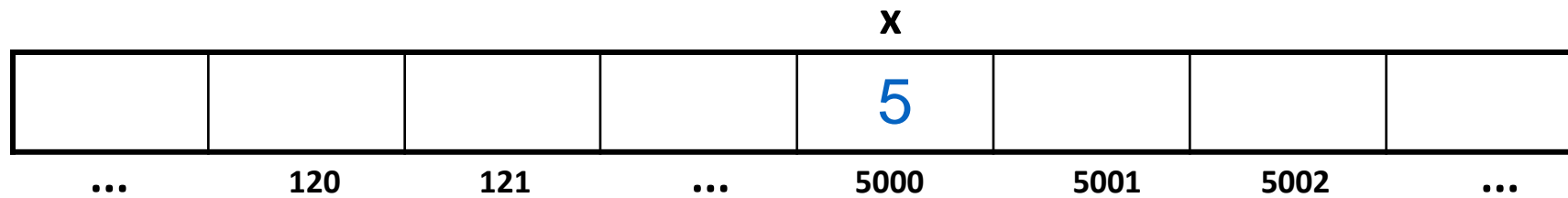


```
C:\WINDOWS\system32\cmd.exe
4
4
0012F3D0
0012F3D4
Prima qualquer tecla para continuar . . .
```

Inicialização de Apontadores

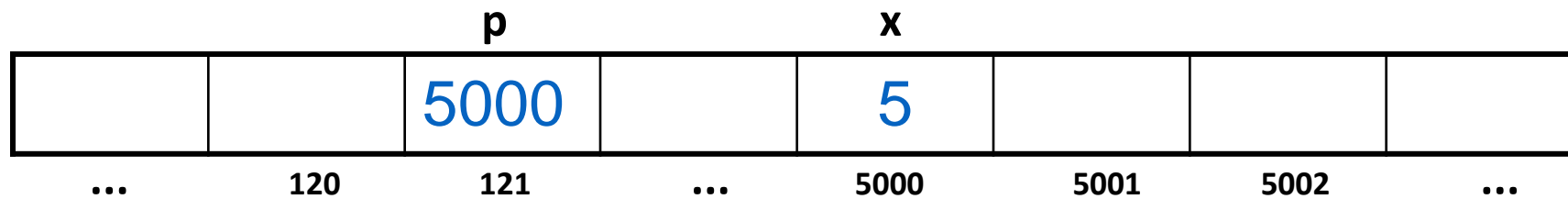
Exemplo 1:

```
int x;  
x = 5;
```



Exemplo 2:

```
int *p;  
p = &x;
```



Apontadores - Exemplos

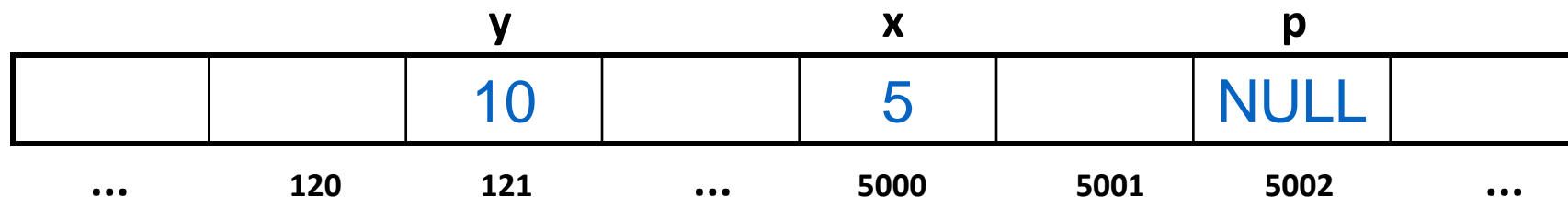
Podem existir situações em que posso declarar um apontador e não querer que este aponte para nenhuma variável.

Posso então colocá-lo a apontar para **NULL** (constante definida na biblioteca `<stdlib.h>`).

Exemplo 3:

```
int x=5, y=10;  
int *p=NULL;
```

Como fazer para **p**
apontar para **y**?

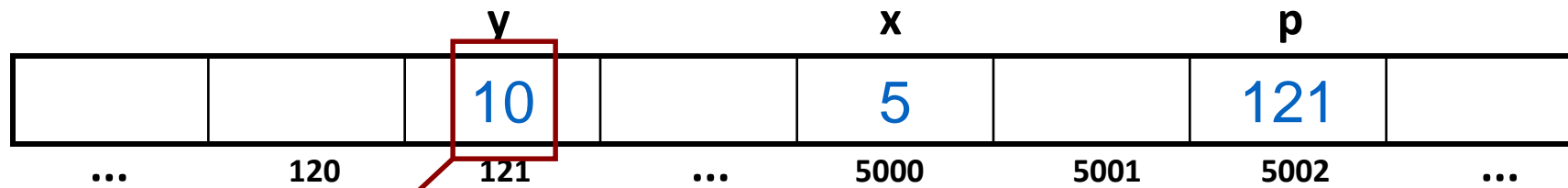


Um **apontador** serve, sobretudo, para permitir aceder a outras variáveis, através dos seus endereços.

Apontadores - Exemplos

Exemplo 4:

```
int x=5, y=10;  
int *p = &y;
```



Se **p** é um apontador, então ***p** permite-nos obter o valor que é apontado por p.

Logo: **y** \leftrightarrow ***p**

EXPRESSÃO	VALOR
x	5
&x	5000
y	10
&y	121
p	121
&p	5002
*p	10

Apontadores - Exercício

```
#include <iostream>
using namespace std;

void main(){

    int *p;
    int numero = 20, valp;
    p = &numero;
    valp = *p;

    cout << "O valor de valp: " << valp << endl;
    *p = 30;
    cout << "O valor de numero e: " << numero << endl;
    cout << "O valor apontado e: " << *p << endl;
}
```

```
O valor de valp: 20
O valor de numero e: 30
O valor apontado e: 30
Prima qualquer tecla para continuar . . .
```