Construtores

Profa. Karen Selbach Borges

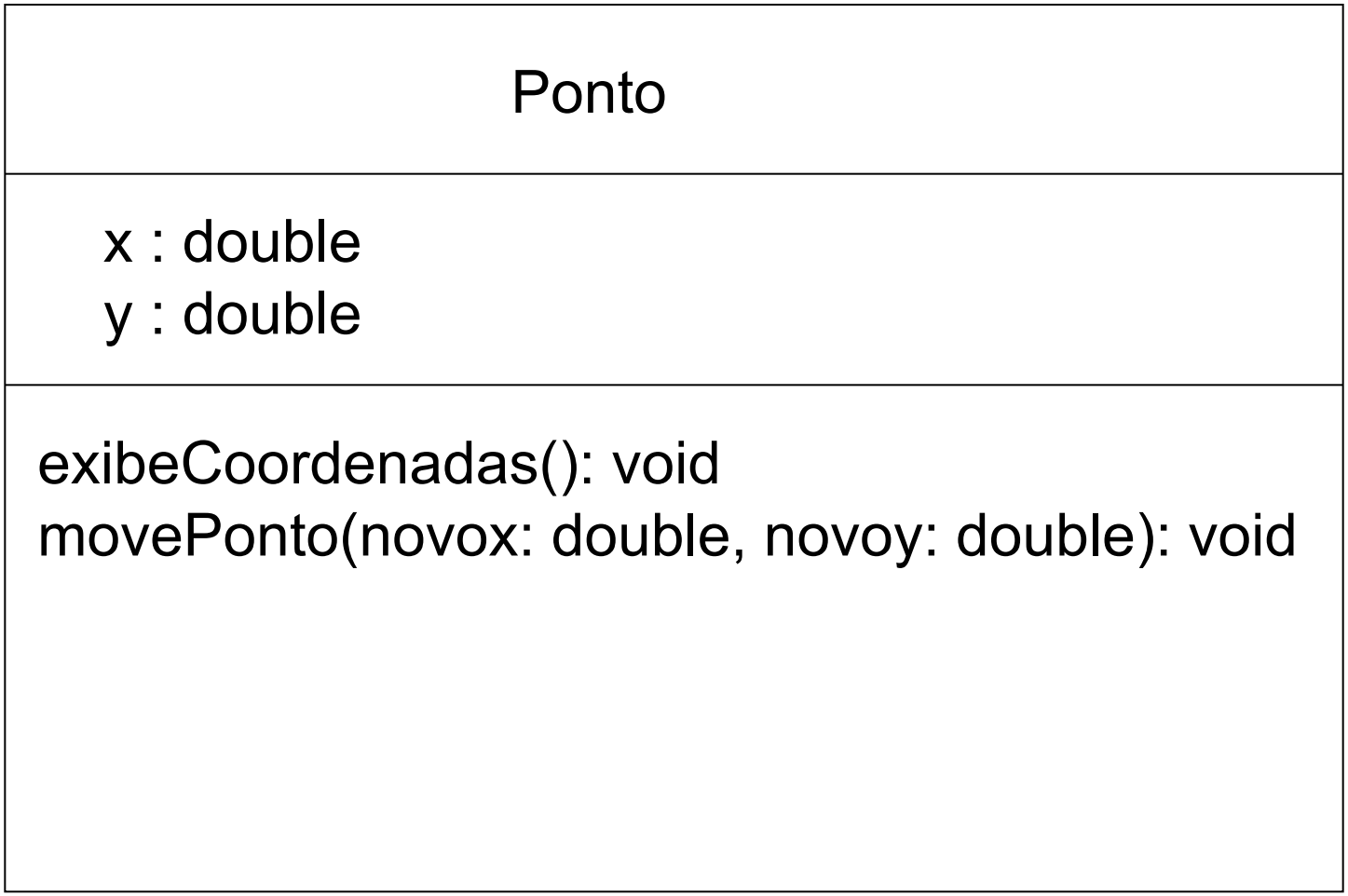
Construtores

- Serve para alocar espaço de memória para o objetos e inicializar os valores das variáveis de instância.
- Java define para cada classe um construtor *default*.

Construtores

- No caso de construtores default, as variáveis de instância são inicializadas com os valores padrão :
  - campos do tipo *boolean* : *false*.
  - campos do tipo *char :* caracter cujo código Unicode é zero e que é impresso como um espaço
  - campos do tipo inteiro (*byte, short, long, int*) ou de ponto flutuante (*float, double*) : valor zero, do tipo do campo declarado.
  - instâncias de qualquer classe, inclusive da classe *String : c*om *null*.

A Classe Ponto (com construtor default)



O Construtor Default

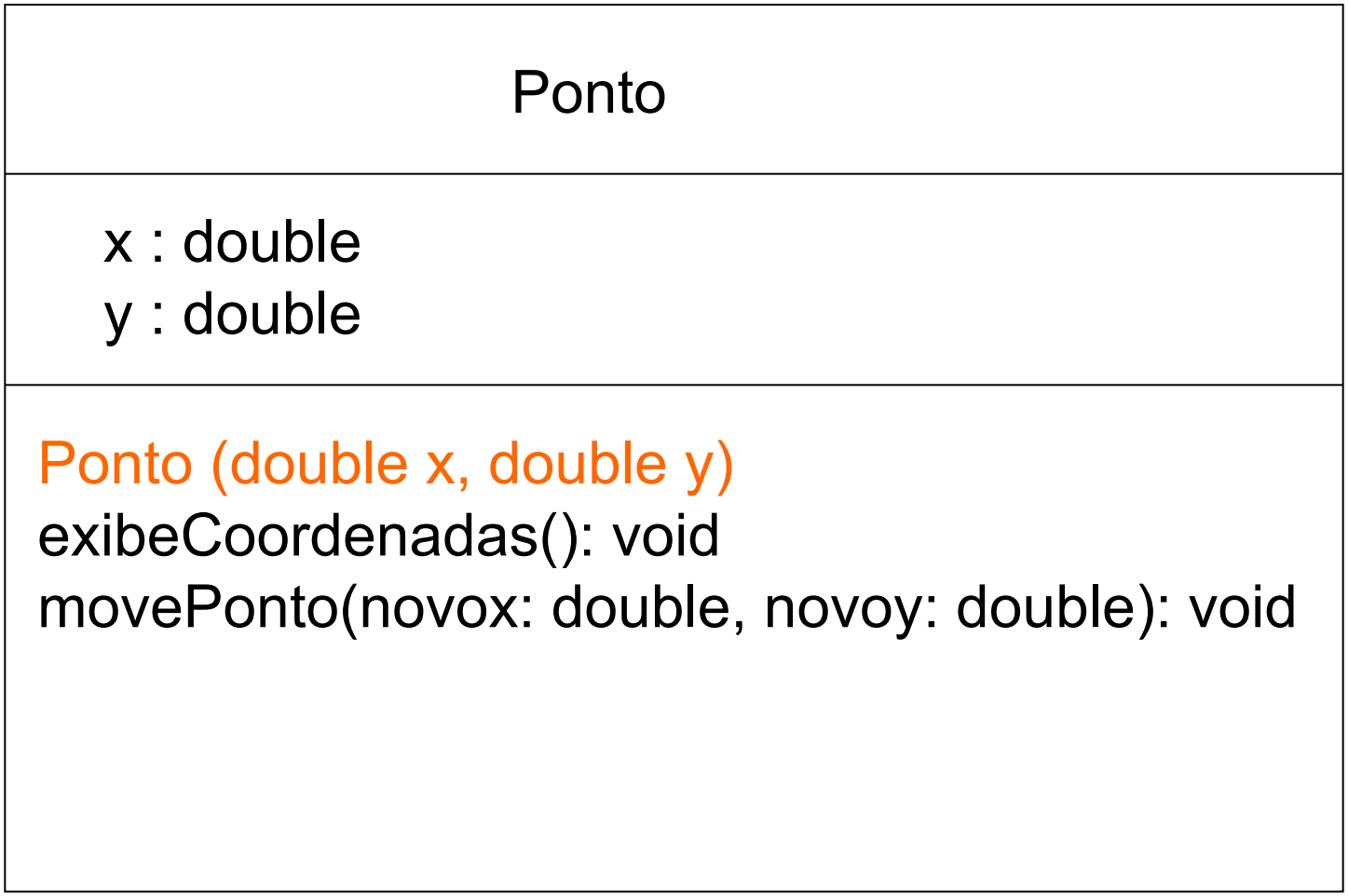
É assim que funciona, mas que você não vê

```
public Ponto ( ) {
    super ();    // referência ao construtor da classe mãe
    this.x = 0;
    this.y = 0;
}
```

Construtores Explícitos

- É possível criar construtores para uma classe
- Neste caso os construtores explícitos :
  - Possuem o mesmo nome da classe
  - Não possuem tipo de retorno, nem mesmo o *void*.

A Classe Ponto (com construtor explícito)



O Construtor Ponto

É assim que implementa

```
public Ponto (double x, double y ) {
    this.x = x;
    this.y = y;
}
```

Sobrecarga de Construtor

É possível ter mais de um construtor para a mesma classe:

```
public Ponto (double x, double y ) {
    this.x = x;
    this.y = y;
}

public Ponto (double x, double y, double z ) {   // Um ponto no plano 3D
    this(x,y);
    this.z = z;
}
```

Criando Objetos

- Em Java os objetos são criados utilizando-se a palavra reservada new.
  - Ex : Correspondencia c1 = new Correspondencia();
  - Observe que :
    - O objeto só passa a existir depois de executa a instrução new Correspondencia().
    - c1 é uma variável, que será utilizada para referenciar uma instância da classe Correspondencia.
    - c1 é um ponteiro para a área de memória que irá armazenar as informações a respeito do objeto.
    - Imprimir c1 irá exibir o OID do objeto

Manipulando Objetos

```
public class Principal {
    public static void main (String[ ] args) {
        Correspondencia c1 = new Correspondencia();
        System.out.println(“OID do objeto c1 = “ + c1);
        c1. peso = 10.5;
        Correspondencia c2 = new Correspondencia();
        System.out.println(“OID do objeto c2 = “ + c2);
        c2 = c1; // Cuidado !
        System.out.println(“OID do objeto c2 = “ + c2);
        c1 = new Correspondencia(); // Cuidado !
        System.out.println(“OID do objeto c1 = “ + c1);
    }
}
```

Manipulando Objetos

- Observações:
  - Atribuições entre variáveis que são referências à objetos (c2 = c1) corresponde a:
    - uma cópia do OID de c1 para c2
    - Em java : um redirecionamento dos ponteiros de memória (c2 e c1 passam a apontar para a mesma área e o objeto referenciado por c2 fica “perdido no espaço”)

Manipulando Objetos

- Observações:
  - Utilizar a mesma variável para a criação de outro objeto (c1 = new Correspondencia()) corresponde a:
    - Atribuição de um novo OID para c1
    - Em Java: a referência ao objeto inicial seja redirecionada à nova área de memória (o objeto referenciado por c1 fica “perdido no espaço”)