

# **Unidade I:**

## **Conceitos Básicos -**

### **Introdução a Orientação por Objetos**

**Prof. Max do Val Machado**



**PUC Minas**

Instituto de Ciências Exatas e Informática  
Curso de Ciência da Computação

# Classes e Objetos

- Motivação: Reaproveitamento de código
- Definições:
  - Classe: é um tipo, um conjunto de regras
  - Objeto: é uma variável do tipo classe

# Classes e Objetos

- Método construtor:
  - Normalmente Inicializa os atributos da classe
  - Apresenta o mesmo nome da classe
  - Não apresenta tipo de retorno
  - Sempre será público

# Classes e Objetos

- Visibilidade:

- *public* : acessíveis “dentro” e “fora” da classe
- *private* : acessíveis somente “dentro” da classe
- Qual é a vantagem disso?

# Classes e Objetos

## • Métodos *get* e *set*:

- Normalmente, temos um *get* e um *set* para cada atributo privado
- O método *get* retorna o conteúdo do atributo privado
- O método *set* altera o conteúdo do atributo privado
- O nome desses métodos serão *getNomeAtributo* e *setNomeAtributo*
- O tipo de retorno do método *get* é o mesmo do seu atributo e esse método não recebe parâmetros
- O método *set* retorna *void* e ele possui um parâmetro de entrada cujo tipo é o mesmo do atributo

# Classes e Objetos

- Estático:

- Atributos e métodos estáticos podem ser chamados sem que um objeto seja instanciado (por exemplo, *IO.println*)
  - Qual é a vantagem disso?
- O valor dos atributos estáticos é compartilhado por todos os objetos da classe
- Os métodos estáticos acessam somente os métodos ou atributos estáticos da classe

# Classes e Objetos

- Operador *this*:
  - Explicita que um método ou atributo pertence ao objeto corrente
  - Ele também pode ser utilizado para explicitar o objeto corrente

# Classes e Objetos

- Exemplo: Classe Retângulo

Ver código em: `LixaoRetangulo.java` e `Retangulo.java`



# Classes e Objetos

🔵 **Exercício:** Um aluno desenvolveu a classe abaixo e pediu sua ajuda para compilá-la. Para ajudar, você deve criar uma classe Ponto com as seguintes regras:

- Quatro atributos privados: double x, double y, int id e int nextID
- Os atributos id e nextID serão alterados somente por um método construtor.
- Implemente os métodos get e set tanto para o atributo x como para o y
- Na declaração do atributo nextID, o mesmo deve receber zero. Além disso, a alteração do valor desse atributo por um objeto sempre será compartilhada com qualquer objeto da classe ponto
- Implemente dois construtores sendo que o primeiro não recebe parâmetros e inicializa os valores de x e y com zero. O segundo recebe dois parâmetros (cujos nomes são obrigatoriamente x e y) e devem ser utilizados para inicializar os valores dos atributos x e y, respectivamente
- Os dois construtores devem atribuir o valor corrente do atributo nextID ao atributo id e incrementar o valor de nextID. Observe que cada objeto terá um ID distinto
- Implemente qualquer método que seja necessário para compilar a classe LixaoPonto

# Classes e Objetos

```
class LixaoPonto {  
    public static void main (String[] args){  
        Ponto p1 = new Ponto(4,3);  
        Ponto p2 = new Ponto(8,5);  
        Ponto p3 = new Ponto(9.2,10);  
  
        System.out.println("Distancia p1 entre e p2: " + p1.dist(p2));  
        System.out.println("Distancia p1 entre e (5,2): " + p1.dist(5,2));  
        System.out.println("Distancia (4,3) entre e (5,2): " + Ponto.dist(4,3,5,2));  
        System.out.println("P1, P2, P3 sao triangulo:" + Ponto.isTriangulo(p1,p2,p3));  
        System.out.println("Area retangulo:" + p1.getAreaRetangulo(p2));  
        System.out.println("ID de P1: " + p1.getID());  
        System.out.println("ID de P2: " + p2.getID());  
        System.out.println("ID de P3: " + p3.getID());  
        System.out.println("Next ID: " + Ponto.getNextID());  
    }  
}
```