

## INSTÂNCIA, ATRIBUTOS E REFERÊNCIAS

Reforçando os seus conhecimentos sobre **OBJETOS**, aprendendo mais sobre **INSTÂNCIA, ATRIBUTOS E REFERÊNCIAS**? Instanciar um objeto é o equivalente a criá-lo na memória do computador. Dessa forma, atributos correspondem as características do objeto que foi referenciado na memória a partir da sua instanciação.

Além disso, sabemos que o responsável por criar um objeto é o **MÉTODO CONSTRUTOR**. Ele o faz a partir da classe em questão, é somente através do construtor que é possível a instanciação de um objeto. E por conseguinte a manipulação efetiva dos atributos e métodos do mesmo.

Em paralelo a responsabilidade de criar um objeto, os construtores também podem inicializar objetos com valores pré-definidos. Em outros termos isso significa que um construtor também responde pela inicialização padrão que alguns objetos precisam ter. Isto é, cabe a ele a atribuição de valores iniciais dos objetos quando necessário.

E por fim, podemos assim sintetizar a definição dos construtores em Java. Trata-se de um método com o mesmo nome da classe, sem retorno e que pode ou não ter parâmetros.

De fato, o construtor pertence a classe que o originou e a sua principal função é criar objetos a partir desta classe. Logo, podemos assumir que o retorno de um construtor são os objetos do tipo da classe. Assim, é por este motivo que o construtor não precisa retornar nenhum valor.

Por padrão, em Java, mesmo que o desenvolvedor não defina um construtor para a classe, ela estará implícita. E terá como assinatura o mesmo nome da classe e nenhum parâmetro, como no exemplo:

```
public Aluno(){  
  
}
```

### Instância do Objeto

Naturalmente, acredito que não alimentamos mais nenhuma dúvida acerca do conceito de instância e/ou instanciação de objetos. Então vamos reforçar nosso conhecimento prático sobre a construção de objetos por meio do operador *new*.

Note que o construtor deverá ser definido dentro do escopo da classe, ou seja dentro do espaço delimitado por chaves. Espaço este que também pode ser entendido como a *workspace* da classe. Do outro lado, a instanciação do objeto se dá no escopo do método principal que é o método autoexecutável, *main*.

No exemplo acima, “**new Aluno()**” representa a instanciação/criação do objeto do tipo “Aluno” que está sendo guardado na variável “aluno01”. Na linha abaixo temos um segundo objeto que dentro da mesma lógica está sendo alocado na variável “aluno02”.

## Operado new

O operador `new` é o responsável pela instância de um objeto, isto é, por criar objetos a partir de uma classe. A título de conhecimento, classes e objetos representam um tipo de dados que são definidos pelo programador. Assim, é lógico pensar que eles possuem operadores específicos que atendam às suas necessidades específicas.

Portanto, é neste cenário que o operador `new`, que também é uma palavra reservada se insere. Visto que ele compreende um destes operadores específicos.

## Referência de um objeto

Quando instanciamos um objeto com o operador `new` e o guardamos em uma variável, o que de fato estamos fazendo? O que estamos fazendo é guardando na memória uma referência para aquele dado objeto e não os seus valores. Veja a ilustração deste conceito representado na imagem abaixo:

Em orientação a objetos, quando por exemplo criamos um objeto do tipo Aluno e o guardamos na variável “aluno01”. Estamos na verdade criando uma referência para que possamos acessar os atributos deste objeto na memória. No caso do exemplo com qual estamos trabalhando estes atributos são: nome, nomeMae, nomePai, numeroCPF, registroGeral e todos os que estabelecemos.

Dessa forma cada objeto que criamos é um novo objeto com uma nova posição a ser referenciada na memória. E para cada objeto criado podemos acessar dados distintos. Assim, alocar um dado em uma variável e o armazenar na memória do computador, nada mais é do que estabelecer um método. Isto é, simplesmente criar um meio de, quando necessário, acessarmos estes dados de forma rápida, fácil e prática.