

Parâmetro com tipo referência

Estudante é a classe de domínio é o modelo mais genético neste exemplo, já a classe com nome ImpressoraEstudante tem uma função específica que é imprimir os valores dos atributos da classe Estudante. Ambas estão no mesmo diretório.

Classe de domínio	Classe com função específica
<pre>public class Estudante { public String nome; public int idade; public char sexo; }</pre>	<pre>public class ImpressoraEstudante { public void imprimir(Estudante estudante) { System.out.println(estudante.nome); System.out.println(estudante.idade); System.out.println(estudante.sexo); } }</pre>

A classe EstudanteTeste possui o método main, é nela que os objetos serão criados e inicializados, mas considere os seguintes detalhes:

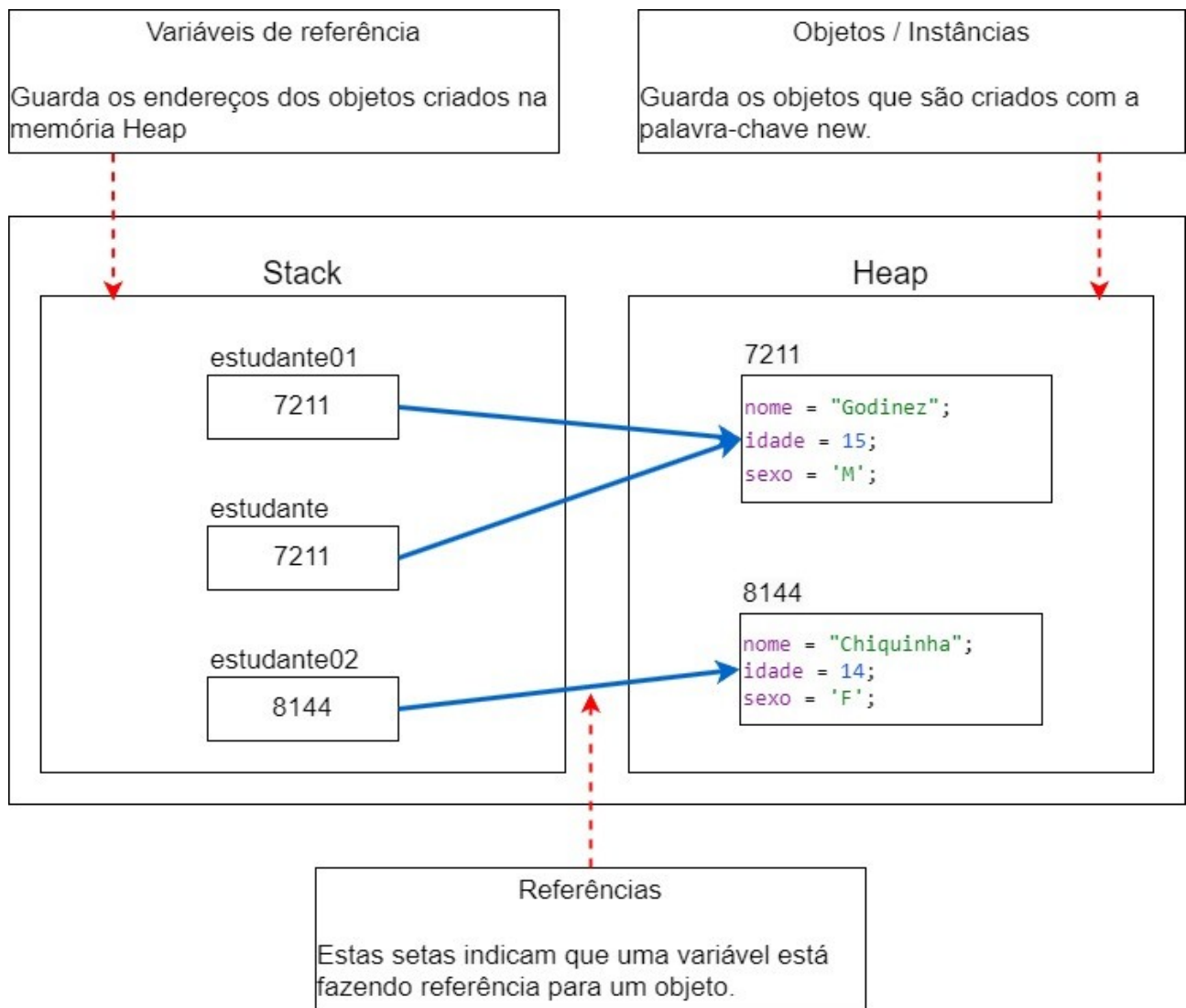
- **EstudanteTeste** cria os objetos, veja que só ela possui a palavra-chave **new**.
- **ImpressoraEstudante** faz referência ao objeto que é passado como argumento no seu método **imprimir**.

Classe de teste
<pre>public class EstudanteTeste { public static void main(String[] args) { Estudante estudante01 = new Estudante(); Estudante estudante02 = new Estudante(); ImpressoraEstudante impressora = new ImpressoraEstudante(); estudante01.nome = "Godinez"; estudante01.idade = 15; estudante01.sexo = 'M'; estudante02.nome = "Chiquinha"; estudante02.idade = 14; estudante02.sexo = 'F'; impressora.imprimir(estudante01); impressora.imprimir(estudante02); } }</pre>

Faz com que a referência de estudante01 também seja de estudante lá dentro do método.

É equivalente a: **estudante = estudante01**

Veja como tudo isso funciona na memória do computador, desenhar talvez possa ajudar a compreender melhor todos esses conceitos.



OBSERVAÇÃO

A variável `estudante` faz referência para um objeto somente quando o método `imprimir` é chamado, assim, a referência será a mesma do objeto passado como argumento.

Quando o método é chamado novamente e outro objeto é passado como argumento, a referência anterior é perdida e `estudante` passa a fazer referência ao novo objeto.

É importante observar que apenas a referência de `estudante` é perdida a cada chamada do método, as outras só serão perdidas quando o `escopo do main` terminar.

