

# Aula 05 – Prática – Introdução à Java

---

Crie um projeto Java no Eclipse chamado “tcp-aula05”, e siga os seguintes passos para a criação da aplicação **TestaExpressaoLogica**.

1. Crie uma interface chamada “ExpressaoLogica”
2. Crie duas classes abstratas que implementam a interface “ExpressaoLogica”
  - a. “ExpressaoUnaria”
    - i. Possui um atributo privado do tipo ExpressaoLogica
    - ii. Possui um getter para este atributo
    - iii. Possui um construtor que recebe como parâmetro uma ExpressaoLogica e armazena no atributo privado
  - b. “ExpressaoBinaria”
    - i. Possui dois atributos privados do tipo ExpressaoLogica
    - ii. Possui getters para estes atributos
    - iii. Possui um construtor que recebe como parâmetro duas ExpressaoLogica e armazena no atributos privados
3. Crie cinco classes (concretas), que sobrescrevem o método toString (da classe Object) retornando o valor indicado entre parênteses.
  - a. ExpressaoAtomica (variável)
    - i. Implementa a interface ExpressaoLogica
    - ii. Possui um atributo privado do tipo String representando uma variável
    - iii. Possui um getter para este atributo
    - iv. Possui um construtor que recebe como parâmetro um String e armazena no atributo privado
  - b. ExpressaoNot (“(not ” + expressão + “)”)
    - i. Estende a classe ExpressaoUnaria
    - ii. Implemente um construtor adequado
  - c. ExpressaoAnd (“(“ + expressão1 + “ and ” + expressão2 + “)”), ExpressaoOr (“(“ + expressão1 + “ or ” + expressão2 + “)”), ExpressaoImplicacao (“(“ + expressão1 + “ --> ” + expressão2 + “)”)
    - i. Estendem a classe ExpressaoBinaria
    - ii. Implementem um construtor adequado
4. Crie uma classe TestaExpressaoLogica com um método main que contenha:
  - a. Declaração de uma variável do tipo ExpressaoLogica
  - b. Instancie uma expressão lógica de forma aninhada para representar a seguinte expressão:  
$$((A \vee B) \rightarrow (C \wedge A)) \rightarrow (((\text{not } B) \wedge C) \vee (A \wedge D))$$
  - c. Imprima a expressão lógica instanciada
5. Adicione o seguinte método à interface ExpressaoLogica:  

```
public boolean eval(Map<String, Boolean> valoresVariaveis)
```

Implemente este método de forma adequada nas classes concretas

  - a. Se você chamar valoresVariaveis.get(umString), você tem como retorno um valor booleano. umString deve ser um nome de variável utilizado em uma expressão atômica.

6. Adicione a criação de um mapa ao método main da classe TestaExpressaoLogica, e imprima a chamada do método eval da variável do tipo ExpressaoLogica
  - a. 

```
Map<String, Boolean> valoresVariaveis = new HashMap<>();
valoresVariaveis.put("A", false);
valoresVariaveis.put("B", true);
valoresVariaveis.put("C", true);
valoresVariaveis.put("D", false);
```
7. Adicione um arquivo chamado README.txt no seu projeto e responda às seguintes perguntas:
  - a. O que aconteceu quando você criou uma das classes concretas e estendeu as classes abstratas?
  - b. O que aconteceu quando você adicionou o método eval à interface ExpressaoLogica?