

# Avaliação de programação

Imagine uma IDE (*Integrated Development Environment*) que permite ao programador trabalhar com diferentes linguagens de programação. Sempre que um arquivo de código fonte é aberto a IDE automaticamente reconhece a linguagem (**com base na extensão do arquivo**) e faz algumas coisas, tais como:

1. **colorir as palavras reservadas da linguagem** que está sendo utilizada no arquivo;
2. **indicar os erros sintáticos** de acordo com a sintaxe da linguagem utilizada no arquivo;
3. **autocompletar** as linhas digitadas pelo programador.

Nesta avaliação você deverá fazer a **implementação conceitual** de uma IDE, pois obviamente não é possível implementar uma IDE completamente funcional no período de uma aula. Por “implementação conceitual” entenda que você precisa programar apenas o suficiente para demonstrar a aplicação dos **padrões de projeto criacionais**, nada mais.

## O que você deve fazer

- Implemente o problema descrito anteriormente usando **pelo menos** os padrões **Simple Factory** e **Abstract Factory**.
- Implemente uma classe **IDE** que lida com vários tipos de arquivos de código fonte. Esta classe deve ter um método que abre um determinado arquivo de código. Depois de abrir o arquivo de código a IDE deve **verificar a sintaxe, colorir as palavras reservadas** (*highlight*) e fazer o **completamento de código**.
- As **classes** que verificam a sintaxe devem retornar a **lista de erros encontrados** (apenas uma lista de *Strings* é suficiente para esta avaliação).
- As classes que fazem o **completamento de código** devem retornar uma lista contendo nomes de métodos em função de uma *String* passada como parâmetro. Por exemplo, se “**System.out.pri**” for passada como parâmetro uma classe de completamento para a linguagem JAVA deveria retornar todos os métodos de **System.out** que iniciam com as letras **pri**. Lembre-se de que você não precisa implementar realmente o completamento de código, você precisa apenas definir as estruturas (classes, métodos, parâmetros, interfaces, etc.) adequadas para que os conceitos fiquem claramente demonstrados.
- Demonstre o funcionamento da sua solução para pelo menos **2 linguagens de programação diferentes**. Para isso, basta que o programa apresente mensagens como “colorindo código JAVA”, “verificando sintaxe JAVA”, “autocompletando JAVA”, etc.

1.

## Critérios de Avaliação

Critério	Peso
Identificação e implementação do Simple Factory	15%
Identificação e implementação do Abstract Factory	65%
Qualidade geral da implementação	20%