

Busca de Linguagens

Descrição

Desenvolva um programa que armazene e busque Linguagens de Programação. Este programa deverá conter duas classes: **Linguagem** e **Linguagens**.

A classe **Linguagem** conterá os seguintes atributos sobre uma linguagem de programação:

- int ano;
- String nome;
- String desenvolvedorChefe;
- String predecessores;

A classe deve ainda sobrescrever o método toString() para retornar uma String em JSON com os atributos da própria classe, como no exemplo abaixo:

```
{
    "ano": 1995,
    "nome": "Ruby",
    "desenvolvedorChefe": "Yukihiro Matsumoto",
    "predecessores": "Smalltak;Perl"
}
```

Para retornar a classe em JSON você deve [serializar a classe usando a biblioteca Jackson](#). Utilize o [Maven para gerenciar as bibliotecas e construir sua aplicação](#).

A classe **Linguagens** deverá armazenar várias linguagens usando um List<Linguagem> e disponibilizará métodos para busca de linguagens na lista baseado em suas propriedades. Todos estes métodos de busca deverão utilizar **Streams do Java** para selecionar a linguagem de acordo com o critério de busca. Os métodos da classe Linguagens são:

- void adicionarLinguagem(Linguagem l): adiciona uma linguagem à lista de linguagens
- void removerLinguagem(Linguagem l): remove uma linguagem da lista de linguagens

- *Linguagem* **buscarLinguagemPorNome**(*String* nome): busca uma linguagem pelo **nome** utilizando o método `startsWith()` da *String* como método de comparação
- *Linguagem* **buscarPorAno**(*int* ano): busca uma linguagem pelo **ano de criação** utilizando o operador de igualdade como método de comparação
- *Linguagem* **buscarPorDesenvolvedorChefe**(*String* nomeDesenvolvedorChefe): busca uma linguagem pelo **nome do desenvolvedor chefe** utilizando o método `contains()` da *String* como método de comparação
- *List<Linguagem>* **buscarPorPredecessor**(*String* nomePredecessor): busca uma linguagem pelo **nome de um predecessor** utilizando o método `contains()` da *String* como método de comparação
- *void* **imprimirLinguagens**(): imprime na tela todas as linguagens cadastradas em JSON

OBSERVAÇÃO: Os métodos devem retornar **null** caso não achem uma linguagem de acordo com os parâmetros de busca.

Utilize as informações abaixo para desenvolvimento e testes do programa:

Ano	Nome	Desenvolvedor Chefe	Predecessores
1990	AMOS Basic	François Lionet, Constantin Sotiropoulos.	STO Basic
1991	Visual Basic	Alan Cooper	QuickBasic
1995	Ruby	Yukihiro Matsumoto	SmallTalk; Pearl
2009	Go	Google	C; Oberon; Limbo; SmallTalk

