

# UPskill – Java+.NET

Programação Orientada a Objetos – Genéricos

### EXERCÍCIO MATRIZ GENÉRICA

**Objectivos específicos:** Classes, variáveis e métodos genéricos. Parâmetros de tipo referência sem restrições e com restrições. Tipos **wildcard**. Cláusula **extends**. Herança e **tipos genéricos**. **Arrays genéricos**.

#### Exercício

Pretende-se uma classe genérica, chamada **MatrizGenerica**, para representar matrizes **bidimensionais irregulares** e **dinâmicas**. Estas matrizes devem permitir, em tempo de execução, adicionar e remover linhas. As linhas podem ter comprimentos diferentes.

Para isso siga os seguintes passos:

1. Descarregar o **projeto base** fornecido no *moodle*.
2. No **IDE** (e.g. Apache Netbeans ou IntelliJ) abrir o projeto.
3. Analisar o **código** do projeto importado.
4. Na classe **MatrizGenerica**, criar os **métodos** para:
  - a) **Obter** o **elemento** da matriz com os índices de linha e coluna recebidos por parâmetro.
  - b) **Adicionar uma linha** no fim da matriz, com elementos que sejam subtipos do tipo genérico da **MatrizGenerica**. Este método recebe como parâmetro uma qualquer coleção nativa do Java com os elementos a adicionar.
  - c) **Verificar** se a matriz **contém** o elemento recebido por parâmetro.
  - d) **Substituir** o elemento da matriz (com os índices de linha e coluna passados por parâmetro) pelo elemento também passado por parâmetro.
  - e) **Remover** da matriz a **linha** com o índice recebido por parâmetro.
  - f) **Retornar** um **array** contendo todos os elementos da **coluna** especificada, pela mesma ordem (i.e., do primeiro para o último elemento). Esse **array** retornado pelo método deve ser recebido por parâmetro. Se o **array** não tiver espaço suficiente para todos os elementos da coluna indicada, tem de ser redimensionado. Se sobrar espaço no **array**, a posição imediatamente a seguir à última posição preenchida deve ficar com *null*.
5. O **código** na classe **MainMatrizGenerica** deve incluir as seguintes operações:
  - a) Instanciação de uma **MatrizGenerica** com elementos de tipo **TrabalhadorPeca**.
  - b) Instanciação de uma **MatrizGenerica** com elementos de tipo **primitivo**.
  - c) Instanciação de uma **MatrizGenerica** com elementos de tipo **interface**.
  - d) Atribuição de uma instância **MatrizGenerica<TrabalhadorPeca>** a uma variável do tipo **MatrizGenerica<Trabalhador>**, sendo a classe **TrabalhadorPeca** uma subclasse de **Trabalhador**.