

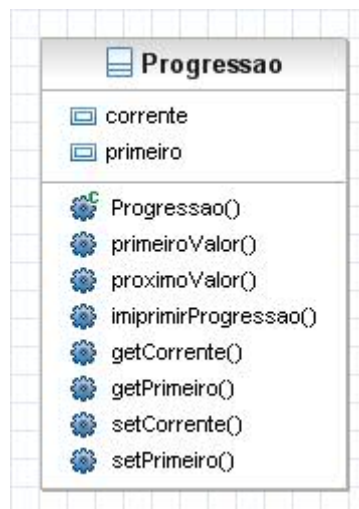
EXERCÍCIO DE APLICAÇÃO 03

Herança e Polimorfismo

Para tornar as noções de herança e polimorfismo mais concretas, vamos analisar alguns exemplos simples em Java.

Particularmente, utilizamos alguns exemplos de classes que percorrem e imprimem progressões numéricas. Uma progressão numérica é uma seqüência de números em que o valor de cada um depende de um ou mais valores anteriores. Por exemplo, uma progressão aritmética determina o próximo número através de adição, e uma progressão geométrica, por multiplicações. Em qualquer caso, uma progressão exige uma forma determinativa do valor inicial, bem como uma maneira de identificar o valor corrente.

Iniciaremos definindo uma classe, ***Progressao***, apresentada no diagrama de classe abaixo, que estabelece os campos e métodos genéricos de uma progressão numérica.



O método ***imprimirProgressao*** não retorna nenhum valor, enquanto que os métodos ***primeiroValor*** e ***proximoValor*** retornam ambos inteiros longos.

A classe ***Progressao*** também inclui o construtor ***Progressao()***, que inicializam as variáveis de instância no momento da criação do objeto. A classe ***Progressao*** visa a ser uma super-classe genérica a partir da qual classes especializadas são derivadas, de maneira que o código do construtor será incluídos nos construtores de cada uma que estende a classe ***Progressao***.

Classe de progressão aritmética

Essa classe define uma progressão em que cada valor é determinado pela adição de um incremento fixo, ***incremento***, ao valor anterior. Ou seja, ***ProgressaoAritmetica*** define uma progressão aritmética, apresentada no diagrama abaixo.

A classe ***ProgressaoAritmetica*** herda aos campos ***primeiro*** e ***corrente***, bem como os métodos ***primeiroValor*** e ***imprimirProgressao*** da classe ***Progressao***. Adiciona um novo atributo, ***incremento*** para armazenar o incremento e dois construtores para inicializá-lo.

Sobrescreve o método *proximoValor* para adequá-lo à forma pela qual obteremos o próximo termo de uma progressão aritmética.



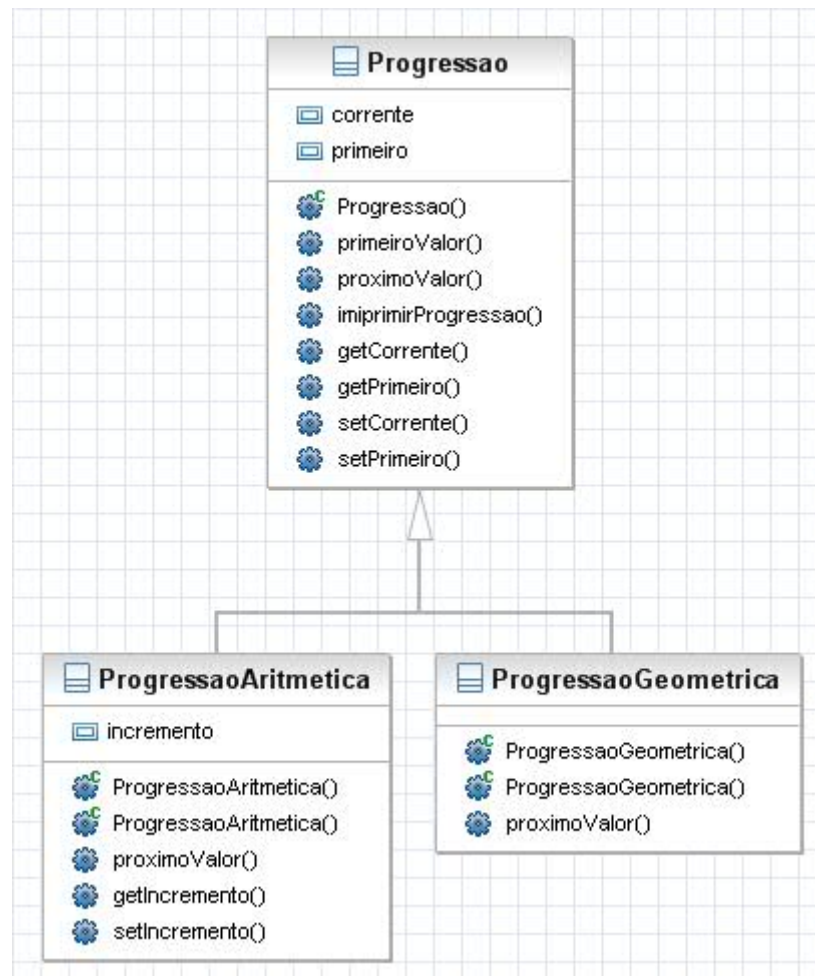
O polimorfismo está presente neste caso. Quando uma referência para *Progressao* aponta para um objeto da classe *ProgressaoAritmetica*, então os métodos *primeiroValor* e *proximoValor* da *ProgressaoAritmetica* serão usados. Este polimorfismo também é real na versão herdada de *imprimirProgressao*, pois as chamadas dos métodos *primeiroValor* e *proximoValor* são aqui implícitas para o objeto atual, que neste caso corresponderá à classe *ProgressaoAritmetica*.

Na definição da classe *ProgressaoAritmetica*, adicionamos dois métodos construtores, um construtor default, sem parâmetros, e uma versão parametrizada que recebe um inteiro para ser usado como incremento da progressão. O construtor default na verdade deve chama o construtor parametrizado passando 1 (um) como parâmetro.

Classe de progressão geométrica

Vamos agora definir uma classe *ProgressaoGeometrica*, apresentada no diagrama abaixo, que permite tanto navegar através de uma progressão geométrica como imprimi-la, sendo esta determinada pela multiplicação do valor prévio por uma base *b*. Uma progressão geométrica é como uma progressão aritmética, exceto pela forma como se determina o próximo valor. Desta forma, *ProgressaoGeometrica* é declarada como subclasse da classe *Progressao*. Da mesma forma que *ProgressaoAritmetica*, a classe

ProgressaoGeometrica herda os atributos *primeiro* e *corrente*, bem como os métodos *primeiroValor* e *imprimirProgressao* da classe *Progressao*.



Na definição da classe *ProgressaoGeometrica*, adicionamos dois métodos construtores, um construtor default, sem parâmetros, e uma versão parametrizada que recebe um inteiro para ser usado como base da progressão. O construtor default na verdade deve chama o construtor parametrizado passando 2 (dois) como parâmetro.

Para completar nosso exercício, vamos definir uma classe chamada *AppProgressao*, que executa um teste simples com cada uma das duas classes. Nesta classe, deve existir uma variável *progressão* que é polimórfica durante a execução do método *main*, uma vez que referencia, alternadamente, objetos das classes *ProgressaoAritmetica* e *ProgressaoGeometrica*.

A aplicação *AppProgressao* é propositadamente pequena, mas mostra um exemplo simples do uso de herança em Java.

A classe *Progressao*, suas subclasses têm uma serie de problemas. Um dos problemas é que as progressões geométricas crescem rapidamente e não existe previsão de tratamento para o estouro inevitável dos inteiros longos envolvidos. Por exemplo, uma vez que $3^{40} > 2^{63}$, uma progressão geométrica de base 3 irá estoura a capacidade de um inteiro longo após 40 iterações.