

Semana 2

Cálculo de Polinómios com
coeficientes racionais



Unidade Curricular de
Laboratório de Programação

2017/2018

Objectivos

- Manipulação de vectores;

Antes de Começar

Neste projecto pretende-se que os alunos exercitem o uso de classes e de vectores para implementar Polinómios com coeficientes racionais. Para isso devem recordar as operações básicas em vectores, manipulação e representação de frações e polinómios.

Enunciado

Neste trabalho pretende-se construir uma ferramenta para manipulação de polinómios com coeficientes racionais. Será então necessário que implemente as classes `Fracao` e `Polinomio` com as operações que se apresentam a seguir. Deve ainda criar uma classe `RunSemana2.java` que permita ler valores do *Standard Input*, construir objectos da classe `Fracao` e `Polinomio` e testar algumas das funcionalidades pedidas.

`Fracao`:

Esta classe deve representar uma fração na sua forma irredutível, ou seja, de modo a que o máximo divisor comum entre numerador e denominador seja 1. Por exemplo, a fração 8/6 é representada por 4/3.

- `Fracao(int num, int den)` -- cria uma nova fração com o numerador e denominador dado assumindo que `den != 0`.
- `boolean ehZero()` -- verifica se esta fração representa o número zero.
- `boolean ehInteira()` -- verifica se esta fração representa um número inteiro.
- `boolean ehPositiva()` -- verifica se esta fração representa um número positivo.
- `Fracao inversa()` -- devolve uma nova fração que é a inversa desta fração (e.g., 2/3 é o inverso de 3/2 e vice-versa) assumindo que a fração não é zero.
- `Fracao soma (Fracao f)` -- devolve uma fração que é a soma desta fração com a fração `f` dada.
- `Fracao produto (Fracao f)` -- devolve uma fração que é o produto desta fração com a `Fracao f` dada.
- `Fracao divisao(Fracao f)` -- devolve uma fração que é a divisão desta fração com a fração `f` dada, assumindo que `f` não é zero.
- `Fracao copia()` -- devolve uma fração que é cópia desta fração.
- `boolean equivalente(Fracao f)` -- devolve `true` se e só se esta fração representa o mesmo número que a fração dada `f`.
- `String toString()` -- devolve uma representação textual desta fração na sua forma irredutível na forma `num/den` se for positiva e `-num/den` se for negativa. Se o denominador for 1 então deve ser omitido, isto é, se representar um inteiro é representada apenas pelo numerador. Por exemplo se a fração for 4/1 deve apresentar "4", se for 4/6 deve devolver "2/3", e se for por exemplo a fração 4/-6 deve apresentar "-2/3").

- `static Fracao makeFromString(String f)` -- este método de classe deve ser capaz de interpretar uma *String* nos formatos "num" ou "num/den" (e.g. "6", "141/89", "-5/3", "-2/-3", "5/-1") e devolver a fração correspondente, assumindo que a *String* dada está sempre num formato válido.

Polinomio:

Esta classe permite representar e operar polinómios com coeficientes que são frações de modo a que este seja representado por um vector com o tamanho do grau do polinómio.

- `Polinomio(Fracao[] coefs)` -- cria um novo polinómio com os coeficientes dados. Se designarmos por *Grau* o grau do polinómio, o coeficiente do monómio de grau *i* do polinómio encontra-se na posição *Grau - i* do vector. Por exemplo, se o vector dado for [3 -2/3 4], o polinómio correspondente é $3x^2 - 2x/3 + 4$.⁽¹⁾ Note que, se o polinómio não for constante (isto é com *Grau* = 0), o primeiro elemento do vector (isto é, o elemento na posição 0) não pode ser 0.
- `int grau()` -- devolve o grau deste polinómio.
- `boolean ehZero()` -- verifica se este polinómio é o polinómio zero.
- `boolean ehConstante()` -- verifica se este polinómio é um polinómio constante.
- `boolean ehIgual(Polinomio p)` -- que diz se este polinómio é igual ao polinómio *p* dado.
- `Polinomio escalar(Fracao f)` -- devolve um polinómio que corresponde à multiplicação deste polinómio por um escalar.
- `Polinomio simetrico()` -- devolve um polinómio que corresponde ao polinómio simétrico deste polinómio (e.g., $-3x^2 + 2x/3 - 4$ é o simétrico de $3x^2 - 2x/3 + 4$).
- `Polinomio soma(Polinomio p)` -- devolve um polinómio que corresponde à soma deste polinómio com o polinómio *p* dado.
- `Polinomio subtracao(Polinomio p)` -- devolve um polinómio que corresponde a subtrair a este polinómio o polinómio *p* dado.
- `Polinomio produto(Polinomio p)` -- devolve um polinómio que corresponde ao produto deste polinómio com o polinómio *p* dado.
- `Fracao avalia(Fracao x)` -- devolve uma fração que é o resultado de avaliar este polinómio em *x* (e.g., $x^2 + 1/2$ com $x = 5$ resulta em $51/2$).
- `Polinomio deriva()` -- devolve um polinómio que corresponde à derivada deste polinómio, (e.g., $6x - 2/3$ é o polinómio derivado de $3x^2 - 2x/3 + 4$).
- `Polinomio copia()` -- devolve uma cópia deste polinómio.

¹ Por exemplo não deve ter um vector do tipo [0 1 1 1] a representar o polinómio $x^2 + x + 1$.

- `String toString()` -- devolve uma representação textual do polinómio da seguinte forma: $a_n x^n + \dots + a_1 x + a_0$, em que a_i é a representação do coeficiente do termo x^i . De notar que se o coeficiente for 0 o termo deve ser omitido da representação e caso o coeficiente seja negativo, o sinal + deve ser substituído por -. Por exemplo o polinómio `[3 -2/3 0 1 4]` deve ser representado por $3 x^4 - 2 / 3 x^3 + 1 x + 4$. O polinómio constante igual a zero deve ser representado por 0.

O que fazer

Deve completar as classes `Fracao` e `Polinomio` cujos esqueletos são dados com as operações descritas acima, sem alterar as respectivas assinaturas. Deve implementar a classe `RunSemana2.java` para verificar a funcionalidade das classes anteriores.

Atenção

Antes de submeter o trabalho, recorde o protocolo de submissão dos trabalhos, certificando-se que **TODOS** os passos descritos no documento são cumpridos.

O que entregar

Deve criar o ficheiro **semana2.zip**, contendo todos os ficheiros `.java` (incluindo os ficheiros de teste). Para tal use o ambiente gráfico para o criar ou a instrução

zip semana2.zip *.java

numa pasta que contenha todos os ficheiros. Posteriormente, deve seguir as indicações dadas no documento com os procedimentos de submissão disponíveis de LabP.