

Programação III

Semestre de Inverno de 2022-2023

2º Trabalho prático

Data de Entrega: 14 de dezembro de 2022

OBJETIVO: Praticar a utilização de *streams* de texto, do mecanismo de tratamento de exceções, da construção de tipos e métodos genéricos, as interfaces funcionais e expressões lambda, e os contentores da *framework* de coleções do JAVA.

Grupo 1

1. Realize o método publico estático, com a seguinte assinatura:

```
public static boolean validate( Reader in ) throws IOException
```

que recebendo por parâmetro um *stream* de texto, contendo código fonte na linguagem Java, retorna **true** se para todo o carácter ‘}’ existiu anteriormente um carácter ‘{’, caso contrário **false**. Considere que os literais do tipo **String** e os comentários não contêm chavetas.

2. Realize um método com a seguinte assinatura:

```
public static void copyCom( BufferedReader in, PrintWriter out ) throws IOException
```

que recebendo por parâmetro um *stream* de texto, contendo código fonte na linguagem Java, copia para o *stream* de texto **out** os comentários de linha indicando em que linha ocorrem. Assume que nem os literais do tipo *string* nem os comentários de bloco contêm a sequência de caracteres que define o início de um comentário de linha (`"//"`).

3. Realize os seguintes métodos estáticos públicos:

```
a) <T> void mapper( BufferedReader in,
                  Function<String, R> mapping,
                  BiConsumer<String, R> action ) throws IOException
```

Que por cada linha lida do *stream in* execute a ação binária **action** passando por parâmetro a linha e o resultado da operação **mapping** sobre a linha. Para obter o mapeamento duma linha deve ser chamado o método **apply** sobre o objeto função **mapping** passando-lhe por parâmetro a linha. Executar a ação binária é chamar o método **accept** passando dois parâmetros: a linha lida e *string* que resultou do mapeamento.

b) **Integer evaluate(String expression)**

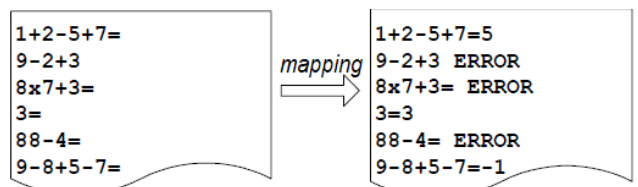
Que retorna o resultado da avaliação da expressão aritmética recebida por parâmetro. Uma expressão aritmética é constituída, por um carácter dígito seguido de uma sequência de zero ou mais pares (operador, carácter dígito), seguidos do carácter ‘=’. Os operadores são o ‘+’ ou ‘-’. Caso a expressão esteja incorreta retorna **null**;

```
<expression> ::= <digit> {<operator> <digit>}* '='
<operator> ::= '+' | '-'
```

c) **void evaluate(String filenameIn, BiConsumer<String,Integer> action) throws IOException**

Que recebendo por parâmetro um ficheiro de texto com nome **filenameIn**, em que cada linha contém uma expressão aritmética, execute a **action** sobre cada linha do ficheiro e o resultado do cálculo da expressão.

Utilize o método da alínea a) e implemente um **Function<String, Integer>**.



d) **void copyEvaluate(String filenameIn, String filenameOut) throws IOException**

Que recebendo por parâmetro um ficheiro de texto com nome **filenameIn**, em que cada linha contém uma expressão aritmética, escreva no ficheiro de texto com nome **filenameOut** a expressão com o resultado do cálculo.

Utilize o método da alínea c) e implemente um **BiConsumer<String,String>**.

e) `void copyEvaluate(BufferedReader in, Writer out) throws IOException`

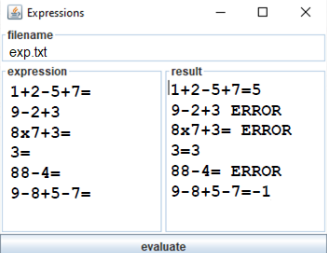
Que recebendo por parâmetro o *stream* de texto *in* em que cada linha contém uma expressão aritmética, escreva no *stream* de texto *out* a expressão com o resultado do cálculo.

Utilize o método da alínea a) e implemente uma `Function<String, Integer>` e um `BiConsumer<String, String>`

f) `String evaluate(String expression) throw ArithmeticException`

Que recebendo por parâmetro uma *string* contendo uma expressão retorne uma *string* com a expressão seguida do resultado do cálculo. Utilize o método da alínea e).

4. Faça uma aplicação com o aspeto da figura. Quando for premido o botão “*evaluate*” chama o método ***evaluate*** da alínea b), para atualizar as duas áreas de texto “*expression*” com as linhas do ficheiro, e “*result*” com a expressão e o resultado do cálculo. Em caso de existirem exceções deve ser escrito na área de texto “*result*” a mensagem associada à exceção lançada.



expression	result
1+2-5+7=	1+2-5+7=5
9-2+3	9-2+3 ERROR
8x7+3=	8x7+3= ERROR
3=	3=3
88-4=	88-4= ERROR
9-8+5-7=	9-8+5-7=-1

Grupo 2

brevemente ...

Bom trabalho