- Faça login no computador seguindo as instruções do docente.
- No directório *Desktop* vai encontrar um conjunto de ficheiros úteis para o exame.
- Utilize o executável `./run-jar` como complemento na especificação do programa.
  Exemplo: `./run-jar p1.txt`
- Pode consultar a documentação das classes `Java` usando o comando `view-javadoc`.
  Exemplo: `view-javadoc ParseTreeProperty`
- Tem à sua disposição os comandos de apoio à programação em `ANTLR4`:
  `antlr4`, `antlr4-build` (ou `build`), `antlr4-run` (ou `run`), `antlr4-clean` (ou `clean`), `antlr4-main` (ou `main`), `antlr4-test` (ou `a4-test`)
- Utilize o enunciado como rescunho, e no final **entregue-o com o cabeçalho preenchido**.
- Caso pretenda desistir deve indicar essa decisão no enunciado e executar o comando: `desisto`

Problema: Pretende-se implementar um interpretador para cálculo vectorial simples. Como exemplo inicial, considere o seguinte programa:

```
# p1.txt
show [1,2];      # escreve na consola o vector [1.0,2.0]
[3] -> v1;       # guarda na variável v1 o vector [3.0]
show v1;         # escreve na consola o valor armazenado na variável v1
show 3;          # escreve na consola o escalar 3.0
5.5 -> e1;       # guarda na variável e1 o valor escalar 5.5
show e1;         # escreve na consola o valor armazenado na variável e1
```

Nota 1: partindo das instruções exemplificadas, tente tornar a linguagem o mais genérica possível.
Nota 2: considere que os vectores não podem ser elementos de outros vectores.
Nota 3: os identificadores para variáveis contêm apenas letras minúsculas e dígitos, não podendo começar com um dígito.
Nota 4: um vector é um array (ou uma lista), não vazio, de valores reais (ex: array nativo ou `ArrayList`).
Nota 5: considere que o números escalares literais são números inteiros ou reais de vírgula fixa.
Nota 6: não se esqueça das verificações semânticas. Existem ficheiros `err?.txt` para o ajudar nesse fim.

a) Implemente em `ANTLR4`, uma gramática `Vector` para esta linguagem. [4 valores]

b) Implemente um interpretador que faça a verificação semântica e execute as instruções desta linguagem. [4 valores]

c) Altere a gramática e o interpretador por forma a permitir a realização das seguintes operações sobre vectores[1]: [6 valores]

- soma/subtracção de escalares ou de vectores (operadores $+$ $-$): tem como resultado um escalar ou (respectivamente) um vector (outra combinação é um erro semântico).
- operadores prefixos unários ($+$ $-$): aplicáveis a qualquer valor (escalar ou vector) resultando, no caso do operador $-$ no(s) valor(es) simétrico(s) (sem alteração no outro caso). Este operador deve ser prioritário relativamente ao anterior.
- parêntesis: este operador serve para impor prioridades na realização de operações sobre vectores.

```
# p2.txt
show  [1,2]+[3,4];                # escreve na consola o vector [4.0,6.0]
[1,1]+[2,2]-([7,8]-[8,7]) -> x;   # guarda na variável x o vector [4,2]
show x-([2,2]);                   # escreve na consola o vector [2.0,0.0]
show 1+2-3;                       # escreve na consola o escalar 0
```

d) Acrescente ainda os seguinte operadores: [6 valores]

[1] Em caso de dúvida sobre o significado de algum destes operadores, sugere-se a execução do ficheiro jar.

- multiplicação de escalares por vectores (ou vice-versa) com o operador $*$: tem como resultado um vector cujos elementos são multiplicados pelo valor escalar (a multiplicação de vectores deve ser considerada um erro semântico).
- multiplicação de escalares também com o operador $*$ (com o resultado aritmético óbvio).
- produto interno de vectores (operador .): tem como resultado um escalar que é o somatório do produto dos valores de cada vector[2].

Considere as seguinte prioridades entre operadores (por ordem decrescente): operadores unários prefixos, multiplicação de valores, produto interno e soma/subtracção de valores.

```
#   p3.txt
2*[1,2] -> x;        # guarda na variável x o valor [2.0,4.0]
show x*(0.5*1);      # escreve na consola o valor [1.0,2.0]
show [1,2].[2,1];    # escreve na consola o valor 4.0
show x+[5,5]*3;      # escreve na consola o valor [17.0,19.0]
```

25 de Junho de 2019