- Faça login no computador com a conta sessaux e passaux. 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 (num terminal aberto no Desktop); ./run-jar pi.txt
 - sejam aceites (p1.txt para as alineas a) e b), etc.).
 - 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-test (a4-test), antlr4-visitor (visitor), antlr4-listener (listener), antlr4-main (main), antlr4-build (build), antlr4-run (run), antlr4-clean (clean)
 - Utilize o enunciado como rescunho, e no fim 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 com números inteiros de precisão ilimitada. Nesse sentido, será necessário a utilização da classe BigInteger da biblioteca java¹ (java.math.BigInteger). Como exemplo inicial, considere o seguinte programa:

```
716257652762576157162 -> c; # guarda um número muito grande na variável c
                      # escreve na consola o número armazenado na variável c
show c;
```

Nota 1. partindo das instruções exemplificadas, tente tornar a linguagem o mais genérica possível.

Nota 2: os identificadores para variáveis contêm apenas letras e dígitos, não podendo começar com um dígito.

Nota 3: A classe BigInteger contém métodos para todas as operações requeridas neste interpretador (existem métodos para todas as operações aritméticas requeridas e há um construtor que faz o equivalente ao Integer.parseInt). Nota 4: Não se esqueça das verificações semânticas. Existem ficheiros err?.txt para o ajudar nesse fim

- (a) Implemente em ANTLR4, uma gramática BigIntCalc para esta linguagem (exemplificada em p1.txt). [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 números (ver programa p2.txt): [6 valores]
 - soma/subtracção/multíplicação/quociențe e resto da divisão de números inteiros (respectivamente, operadores * / * div mod), com as precedências aritméticas naturais.
 - operadores prefixos unários (+ -): aplicáveis a qualquer expressão. Este operador deve ser prioritário relativamente a todos os anteriores.
 - parêntesis: este operador serve para impor prioridades na realização de operações aritméticas.

Il negate 11

¹Ver documentação com: view-javadoc BigInteger

```
# pd mm

*have 100*1000-500*1800;

2 -5 *21; *show *21;

*thine *22;

*thine *22;

*thine *21; *21:1214 div *22 = *21**21 and 1000;

*thine *21:1214 div *22 = *21**21 and 1000;

*thine *21:1214 div *22 = *21**21 and 1000;

*thine *22:122 = *1000 div *10) -> *22;

*thine *22:123 div *5 *mod *2;

*thine *23:4 div *5 *mod *2;
```

J d) Acrescente ainda uma instrução condicional em que a condição é falsa se for o número pero, e verdadeira em todos os outros casos (ver programa p3, txt).

sero, e verdadeira em todos os outros casos (ver propusar un qualquer número de instruções (in-A instrução condicional deve poder seleccionar um qualquer número de instruções (inchindo, pero), e permitir optativamente a existência da selecção de instruções quando a condição é falsa.

23 de Junho de 2020