

# Enunciados Tipo para o Teste Prático

## Layout do teste prático:

1. A primeira consiste em alterar um comando já realizado ou em acrescentar uma nova expressão composta à linguagem suportada pelo projecto; este ponto consiste em alterar ou criar um novo ficheiro para a expressão, e adicionar um novo “case” no **NewParser.parseCompositeExpression()**.
2. A segunda corresponde a criar uma nova opção relativamente simples; cria-se a nova opção, os métodos necessários para o evaluate() dessa opção e adiciona-se essa opção no seu menu.
3. Na terceira pergunta será também necessário concretizar uma nova opção, mas agora com uma complexidade maior; mesma coisa que o anterior, mas possivelmente exige uma vista de olhos para a documentação Java.
4. A quarta questão corresponde simplesmente aos alunos mostrarem que sabem **compilar** e **executar** programas java a partir da linha de comandos utilizando os comandos *javac* e *java*. **Esta pergunta é eliminatória!!!!!!!!!!**  
Se quiserem compilar/correr o projecto todo, mudem a directoria através do Terminal para uma acima do projecto (assim que vejam a directoria “pex”) e para:
  - Compilar: `javac -cp ./path/to/support/packages/* `find pex -name "*.java"``
  - Correr: `java -cp ./path/to/support/packages/* pex.app.App`

## Enunciado 1

Este enunciado é o oficial da página Fénix de PO. Se já se sentem à vontade com este, saltem para o próximo.

1. Acrescente uma nova expressão unária **sqrt** à linguagem suportada pelo seu programa. Esta expressão calcula a raiz quadrada do avaliação do seu argumento. O literal inteiro devolvido apenas tem em conta a parte inteira do resultado. Para realizar esta instrução pode utilizar o método estático **double sqrt(double)** presente na classe **java.lang.Math**.
2. Acrescente uma nova opção ao menu de manipulação de programas que pede um número ao utilizador e remove a expressão do programa que está guardada nessa posição. Caso seja dada uma posição inválida, o comando criado deve lançar a excepção **pex.app.BadPositionException** e o programa não é alterado.
3. Acrescente uma nova opção ao menu principal. Esta opção deve listar todos os programas do interpretador. Cada programa deve ser apresentado indicando primeiro o seu nome e só depois deve ser apresentado o seu conteúdo.

## Enunciado 2

1. Acrescente uma nova expressão **nop** à linguagem suportada pelo seu programa. Esta expressão não faz absolutamente nada, nem aceita nenhum argumento.  
O valor da expressão é 0.
2. Acrescente uma nova opção ao menu principal que pede o nome de um programa ao utilizador (através de **requestProgramId()**) e dispõe o número de linhas (ou seja, o número de expressões) que o dado programa contém.  
Se o programa não existir, é comunicado o erro através de **noSuchProgram()**.
3. Acrescente uma nova opção ao menu principal que mude o nome de um programa. Comece por pedir o nome do programa a alterar, seguidamente pelo novo nome. Comunicar erro se o programa não existir.

## Enunciado 3

1. Acrescente uma nova expressão binária **repeat**, onde o primeiro argumento deve devolver um inteiro. Esta expressão faz o mesmo que o **while**, a diferença sendo que avalia o segundo argumento o número de vezes determinado pelo primeiro.  
O valor da expressão é o valor do primeiro argumento, portanto é o número de vezes que corre o ciclo.
2. Acrescente uma nova opção ao menu de manipulação de programas que pede um número ao utilizador e dispõe a expressão do programa que está guardada nessa posição. Caso seja dada uma posição inválida, o comando criado deve lançar a exceção **pex.app.BadPositionException**.
3. Acrescente uma nova opção ao menu de manipulação de programas que mostre os identificadores disponíveis no programa juntamente com os respectivos valores. Um identificador não definido devolve 0.

## Enunciado 4

1. Acrescente uma nova expressão ternária **shift** que imita o shift aritmético. Esta expressão recebe como primeiro argumento um valor booleano que determina se o shift de bits deve ser feito para a esquerda com *false* (`== 0`) ou para a direita com *true* (`!= 0`); o segundo argumento será o valor ao qual irá efectuar o shift; finalmente, o terceiro argumento será o número de posições a deslocar.  
O valor da expressão será o resultado do shift aritmético.
2. Acrescente uma nova opção ao menu de manipulação de programas que limpe o programa todo, ou seja que apague todas as expressões que contenha.
3. Acrescente uma nova opção ao menu principal que acrescenta um programa ao fim de outro. Será pedido ao utilizador os nomes dos dois programas, dos quais o primeiro será alterado. Se algum dos programas não existir, é comunicado o erro através de **noSuchProgram()**.