Exercício 2

Recursão

Unidade Curricular de
Laboratórios de Programação
2020/2021

Objetivos

- Aplicação do conceito de recursão a problemas concretos.

Antes de Começar

De modo a poder realizar este projeto deverá já ter lido o guião sobre recursão e recordado o que foi ensinado em AED sobre o assunto.

O que fazer

- Descarregar o ficheiro `alunosExercicio2.zip` disponível na página de LabP.
- No Eclipse, escolher `File` → `Import` → `General` → `Existing Projects into Workspace`, e em seguida `Select archive file` e selecionar esse ficheiro.
  Deverá passar a ter um projeto chamado `Exercicio2` contendo:
  - Na pasta `src`, o ficheiro `RunRecursive.java`
  - Na pasta `tests`, o ficheiro `TestsRecursive.java`, com vários testes `JUnit`
- Criar, na pasta `src`, uma classe `Recursive.java` com **implementações recursivas** dos métodos abaixo indicados.

Para testar as suas implementações pode usar a classe `RunRecursive` e a classe `TestsRecursive` de testes `JUnit`.

Segue-se uma descrição detalhada de cada um dos métodos a implementar, que deverão ser todos métodos de classe (métodos `static`):

1. `long product(int n)` que, dado um inteiro positivo `n` devolve o produto de todos os inteiros entre `1` e `n`.
2. `int numberDigits(int n)` que recebe um inteiro positivo `n` e devolve o número de dígitos de `n`.
3. `int[] minimumMaximum(int[]v)` que recebe um vetor de inteiros `v` e devolve um vetor de dois inteiros tal que a componente de índice 0 é o menor valor em `v` e a de índice 1 é o maior valor em `v`. Assuma que `v` é um vetor com pelo menos um inteiro.

4. `boolean isPalindrome(String s)` que recebe uma *string* `s` e verifica se `s` é um palíndromo (i.e. `s` é igual a essa mesma *string* invertida, como por exemplo as *strings* `"rapar"` ou `"oddo"`). Assuma que `s` é uma *string* não nula e não vazia.
5. `String histogram(int[] v)` que devolve uma *string* representando um histograma ou gráfico de barras, com as barras na horizontal, onde cada barra corresponde a `v[i]` asteriscos, para 0 ≤ `i` < comprimento de `v`.
   Assume-se que todos os elementos de `v` são não negativos.
   Por exemplo, a execução das instruções seguintes:

   ```
   int[] valores = {1,0,5,7,5,13,2};
   String s = histogram(valores);
   System.out.println(s);
   ```

   deverá resultar na escrita das linhas seguintes:

   ```
   |*
   |
   |*****
   |*******
   |*****
   |*************
   |**
   ```

   Note que o método `histogram` não deve escrever nada, deve apenas retornar uma *string* que, se for escrita, terá o aspeto acima ilustrado.

   ATENÇÃO: Se pretender uma tarefa mais desafiante, pode implementar uma variante deste método, descrita no fim deste enunciado.

6. `long sequence(int n)` que calcula o `n`-ésimo elemento de uma sequência. Os primeiros três elementos dessa sequência são iguais a `1` e, para qualquer `n > 3, sequence(n) = sequence(n-2) + sequence(n-3)`.
   Assume-se que `n` é um inteiro positivo.
   Deve procurar apresentar uma implementação eficiente.

Antes de Entregar

Antes de entregar, certifique-se da correção da formatação e inclua comentários *javadoc* para todos os métodos.

Na linha do ficheiro contendo a *tag* `@author` indique o seu nome e número de aluno.

Entrega

Deve submeter um ficheiro `t2fcxxxxx.zip`, onde `xxxxx` é o seu número de aluno, contendo o ficheiro `Recursive.java`

Variante do exercício 5 (opcional)

Este exercício é uma generalização do exercício 5, atrás descrito.

Defina uma função com assinatura

`String histogram(String header, String c[], int ident, int[] v)`
que devolve uma *string* representando um histograma, ou gráfico de barras, relativo a um vetor de categorias `c` que têm associados os valores no vetor `v`. O parâmetro `header` é o título a incluir no início da *string*, ao qual se seguem as linhas com as barras do histograma, cada uma delas constituída pelo nome da categoria, `c[i]`, o separador `'|'` e uma sequência com `v[i]` asteriscos.
Assume-se que `c` e `v` têm a mesma dimensão, superior ou igual a 1.
Por exemplo, a execução das instruções seguintes:

```
String[] grupos = {"Até 12", "De 13 a 29", "De 30 a 59",
                   "De 60 a 79", "de 80 a 99", "100 ou mais"};
int[] valores = {1, 5, 7, 5, 13, 2};
int ident = 13;
String s = histogram("Pacientes em UCI por idades:",
                              grupos, ident, valores);
System.out.println(s);
```

deverá resultar na escrita das linhas seguintes onde, em cada linha, `ident` é o número de caracteres antes de `'|'`:

```
Pacientes em UCI por idades:

Até 12        |*
De 13 a 29    |*****
De 30 a 59    |*******
De 60 a 79    |*****
De 80 a 99    |*************
100 ou mais   |**
```