

Laboratório 2 - Algoritmos e Estruturas de Dados

Prof. Inês Almeida

19 de Abril de 2022

Deve fazer clone do repositório para a máquina de trabalho. Deve manter o repositório GitHub atualizado com o seu trabalho.

Datas

- Enunciado: 19 de abril de 2022;
- Data de disponibilização dos testes: 22 de abril de 2022;
- Entrega no e-learning e GitHub: 28 de abril de 2022, 23:59;

Entrega

O código produzido deverá estar disponível no repositório GitHub gerado pelo GitHub Classroom. Deve sempre existir um branch main, onde a versão final deverá ficar disponível.

Documentos Disponibilizados

No repositório são disponibilizados todos os documentos necessários para a utilização de Listas Ligadas (LinkedList) e respetivos nós. É disponibilizado no diretório models as seguintes classes:

1. `LinkedList`
2. `Node`

Parte 1 - Sistema de armazenamento de nomes de países - Linked List

Pretende-se que implemente um programa em que utilizador possa armazenar e consultar nomes de países. O programa tem de ser implementado através de Listas Ligadas (Linked List). No repositório deste laboratório são disponibilizados as classes: LinkedList e Node. Para realizar este exercício é necessário recorrer ao métodos disponibilizados por estas classes. Devem ser implementadas as seguinte funcionalidades:

- Registrar País no Início da Lista (RPI):

O país introduzido pelo utilizador deve ser inserido no início da lista.

– Entrada:

`RPI_país_novo`

– Saida com sucesso (São apresentados os países que se encontram na lista, sendo que o país que se registou deve ser apresentado em primeiro lugar):

`país_novo`

`país_1`

`país_2`

`...`

`país_n`

- Registrar País no Fim da Lista (RPF):

O país introduzido pelo utilizador deve ser inserido no fim da lista.

1. Entrada:

`RPF_país_novo`

2. Saida com sucesso: (São apresentados os países que se encontram na lista, sendo que o país que se registou deve ser apresentado em último lugar)

Exemplo:

`país_1`

`país_2`

`país_3`

`...`

`país_novo`

- Registrar País Depois de outro Elemento já Registrado (RPDE):
O novo país introduzido pelo utilizador deve ser inserido depois do país já registado e indicado pelo utilizador.

1. Entrada:

RPDE_país_novo_país_registado

2. Saída com sucesso:

país_1

país_2

...

país_registado

país_novo

Exemplo: Considerando que Espanha já se encontra na lista, o novo país indicado pelo utilizador (França) é introduzindo na posição posterior onde está Espanha. Entrada :

RPDE_França_Espanha

Saída de Sucesso:

Portugal

Itália

Espanha

França

Japão

- Registrar País Antes de outro elemento já registado (RPAE):
O novo país introduzido pelo utilizador deve ser inserido antes do país já registado e indicado pelo utilizador.

1. Entrada:

RPAE_país_novo_país_registado

2. Saída com sucesso:

país_1

país_2

...

país_novo

país_registado

Exemplo: Considerando que Espanha já se encontra na lista, o novo país indicado pelo utilizador (França) é introduzindo na posição anterior onde está Espanha.

Entrada:

RPDI_França_Espanha

Saída de Sucesso:

Portugal

Itália

França

Espanha

Japão

- Registrar País num determinado Índice (RPII):
O novo país introduzido pelo utilizador deve ser inserido no índice indicado pelo mesmo.

1. Entrada:

RPI_país_novo_índice

2. Saida com sucesso:

país_1

país_2

...

país_novo

3. Exemplo:

Entrada

RPI_Argentina_1

Saida:

Argentina

Portugal

Itália

França

Espanha

Japão

- Verificar Número de elementos da lista (VNE):
Deve ser retornado o número de elementos presentes na lista

1. Entrada:

VNE

2. Saida com sucesso:

0_número_de_elementos_são_número_elementos.

- Verificar se um País se encontra na lista (VP):

1. Entrada:

VP_nome_país

2. Saida com sucesso:

0_país_nome_país_encontra-se_na_lista.

3. Saida com insucesso:

0_país_nome_país_não_se_encontra_na_lista.

- Eliminar o primeiro elemento da lista (EPE):

1. Entrada:

EPE

2. Saida com sucesso:

0_país_nome_país_eliminado_foi_eliminado_da_lista.

- Eliminar o último elemento da lista (EUE):

1. Entrada:

EUE

2. Saida com sucesso:

```
O_país_<nome_país_eliminado>_foi_eliminado_da_lista.
```

- Eliminar um determinado país da lista (EP):

1. Entrada:

```
EP_nome_país_
```

2. Saida com sucesso:

```
O_país_<nome_país_eliminado>_foi_eliminado_da_lista.
```

3. Saida com insucesso: Caso o país não se encontre na lista deve ser apresentado:

```
O_país_<nome_país_eliminado>_não_se_encontra_na_lista.
```

0.1 Regras

O programa deve ser implementado recorrendo apenas a listas ligadas Python (Linked List). O programa deve ser implementado com recurso a funções e de acordo com a arquitetura Model View Controller (MVC). Deve criar o diretório task1 e criar os seguintes ficheiros:

- view.py onde são invocadas todas as funções necessárias para a implementação do programa.
- model.py onde são definidas as estruturas de dados que representam a informação.
- controller.py onde são implementadas todas as funções necessárias para implementação do programa.
- program.py com main. **Deve ser indicado em comentário quais são os requisitos do programa e qual o resultado esperado.**

0.2 Avaliação

Esta parte será avaliada com base em duas componentes: quantitativa (A), e qualitativa (B). A nota final da parte 1 é determinada por $(0.8 \times A) + (0.2 \times B)$.

Avaliação quantitativa: A avaliação quantitativa será baseada nos testes unitários disponíveis no repositório GitHub deste laboratório.

Instruções	Peso
RPI	2
RPF	2
RPDE	2
RPAE	2
RPII	2
VNE	2
VP	2
EPE	2
EP	4

Avaliação qualitativa: A avaliação qualitativa irá considerar que existem várias formas de resolver o problema descrito, mas exige-se a:

- Separação entre interface, dados, e lógica da aplicação;
- Indicação, no ficheiro `program.py`, dos requisitos do programa e respetivo resultado esperado;
- Justificação clara para as variáveis e operações implementadas;
- Adequação da escolha de estruturas de dados e algoritmos para a resolução do problema.