Laboratório 3 - Algoritmos e Estruturas de Dados

Prof. Inês Almeida

2 de maio de 2022

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

Datas

- Enunciado: 2 de maio de 2022;
- Entrega no e-learning e GitHub: 8 de maio 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 documentos de apoio para a utilização de Listas Duplamente Ligadas (DoublyLinkedList) e respetivos nós. É disponibilizado no diretório models as seguintes classes:

1. `DoublyLinkedList`
2. `Node`

Parte 1 - Sistema de armazenamento de cidades - Doubly Linked List

Pretende-se que implemente um programa em que utilizador possa armazenar e consultar nomes de cidades. O programa tem de ser implementado através de Listas Duplamente Ligadas (Doubly Linked List). No repositório deste laboratório são disponibilizados as

classes: DoublyLinkedList e Node. Para realizar este exercicio é necessário recorrer ao métodos disponibilizados por estas classes. Devem ser implementadas as seguinte funcionalidades:

- Registar Cidade no Fim da Lista (RCF):
  A cidade introduzida pelo utilizador deve ser inserida no fim da lista.

  1. Entrada:

     `RCF␣cidade_nova`

  2. Saida com sucesso: (São apresentadas as cidades que se encontram na lista, sendo que a cidade que se registou deve ser apresentada em último lugar)

     `cidade_1`

     `cidade_2`

     `cidade_nova`

- Registar Cidade no Início da Lista (RCI):
  Para esta funcionalidade deverá implementar um método na classe DoublyLinkedList que permita inserir a cidade no ínicio da lista.

  – Entrada:

    `RCI␣cidade_nova`

  – Saida com sucesso (São apresentados as cidades que se encontram na lista, sendo que a cidade que se registou deve ser apresentahttps://www.overleaf.com/project/62684a189c787543 em primeiro lugar):

    `cidade_nova`

    `cidade_1`

    `idade_2`

    `...`

    `cidade_n`

- Apresentar os elementos da lista do início para o fim (AEI):

  1. Entrada:

     `AEI`

  2. Saida com sucesso:

```
cidade_1
cidade_2
...
cidade_n
```

- Apresentar os elementos da lista do fim para o início (AEF):

  1. Entrada:

     ```
     AEF
     ```

  2. Saida com sucesso:

     ```
     cidade_n
     ...
     cidade_2
     cidade_1
     ```

- Eliminar a primeira cidade da lista (ECI):
  Para esta funcionalidade deverá implementar um método na classe DoublyLinkedList que permita eliminar o primeiro elemento da lista.

  1. Entrada:

     ```
     ECI
     ```

  2. Saida com sucesso: São apresentados os elementos da lista (sem a cidade eliminada).

0.1 Regras

O programa deve ser implementado recorrendo apenas a listas duplamente ligadas Python (Doubly 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 x A) + (0.2 x B).

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

Instruções	Peso
RCF	3
RCI	5
AEI	3
AEF	3
ECI	6

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.