



UNIVERSIDADE FEDERAL DE MINAS GERAIS
GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO
Linguagens de Programação

Implementação do Tipo de Dado Abstrato Conjunto em Python

Rodrigo Viana Souza, 2020054803

BELO HORIZONTE
2023

1) Introdução

Para implementar o tipo de dados conjunto, foi utilizado o paradigma imperativo na linguagem python. Diante disso, foi utilizado a lista que é uma estrutura dinâmica que permite armazenar elementos de maneira ordenada, assim, ela foi escolhida para representar os conjuntos, onde cada elemento é único.

2) Implementação

A estrutura de dados, como já dito antes, utilizada para implementar o tipo abstrato "Conjunto" foi a lista `list`. No Python, a lista é uma estrutura de dados dinâmica que permite armazenar elementos de maneira ordenada e acessar esses elementos por meio de índices. No contexto deste código, a lista é utilizada para representar os elementos presentes no conjunto. Cada elemento na lista é único, refletindo a característica de conjuntos, onde não há repetição de elementos.

1. `ehVazio`:

- Algoritmo: Retorna `True` se o comprimento da lista de elementos for zero, indicando que o conjunto está vazio.

2. `uniaoConj`:

- Algoritmo: Cria um novo conjunto e adiciona os elementos únicos de ambos os conjuntos à nova lista.

3. `intersConj`:

- Algoritmo: Cria um novo conjunto e adiciona os elementos que são comuns a ambos os conjuntos à nova lista.

4. `diffConj`:

- Algoritmo: Cria um novo conjunto e adiciona os elementos que estão em A, mas não em B, à nova lista.

5. `diffSimConj`:

- Algoritmo: Utiliza as operações de união, interseção e diferença para calcular a diferença simétrica.

6. `addConj`:

- Algoritmo: Verifica se o elemento já está na lista e o adiciona apenas se não estiver presente.

7. `remConj`:

- Algoritmo: Verifica se o elemento está na lista e o remove apenas se estiver presente.

8. `tamConj`:

- Algoritmo: Retorna o comprimento da lista de elementos, representando o número de elementos no conjunto.

9. `ehElemConj`:

- Algoritmo: Verifica se o elemento está na lista, indicando se ele pertence ao conjunto.

3) Utilização do Código

O código pode ser utilizado da seguinte maneira:

1. Criação de Conjuntos:

```
conjuntoA = Conjunto()  
conjuntoB = Conjunto()
```

2. Adição de Elementos:

```
conjuntoA.addConj(1)  
conjuntoA.addConj(2)  
conjuntoA.addConj(3)  
conjuntoA.addConj(5)
```

3. Impressão:

```
print("Conjunto A: ", conjuntoA.lista)  
print("Conjunto B: ", conjuntoB.lista)
```

4. Operações entre Conjuntos:

python

```
uniao = conjuntoA.uniaoConj(conjuntoB)  
print("União: ", uniao.lista)  
intersecao = conjuntoA.intersConj(conjuntoB)  
print("Interseção: ", intersecao.lista)  
diferenca = conjuntoA.diffConj(conjuntoB)  
print("Diferença: ", diferenca.lista)  
dif_sim = conjuntoA.diffSimConj(conjuntoB)  
print("Diferença Simétrica: ", dif_sim.lista)
```

5. Verificação e Manipulação:

```
print("É vazio? ", conjuntoA.ehVazio())  
print("Tamanho: ", conjuntoA.tamConj())  
print("Contém o elemento 2? ", conjuntoA.ehElemConj(2))  
conjuntoA.remConj(2)
```

```
print("Conjunto A: ", conjuntoA.lista)
```

Esse código acima foi colocado no arquivo main.py que contém um import que importar a classe Conjunto com as suas operações, assim basta executar o main.py pra ver uso das funções da classe conjunto e as suas implementações. Para isso, foi criado o teste.sh que faz a execução desses códigos, com o comando:

- chmod +x teste.sh
- ./teste.sh

4) Conclusão

Nesse projeto, foi implementado um tipo de dado abstrato "Conjunto" em Python, seguindo o paradigma imperativo, que permite uma manipulação simples e eficiente do código. A estrutura de dados utilizada foi a lista que representou os conjuntos, explorando as operações básicas fornecidas por esta estrutura.

Cada função, como a verificação de conjunto vazio, união, interseção, diferença e outras, foi implementada de maneira apropriada, utilizando algoritmos simples e eficazes. O código final oferece funcionalidades básicas para manipular conjuntos, respeitando as restrições e diretrizes fornecidas no enunciado do projeto.

O script de teste em shell fornece uma execução do main.py tem contém uma série de operações com os conjuntos que funcionam corretamente.