

LISTAS

1. Conjuntos e Sequências

Uma sequência é uma enumeração, uma indexação de um conjunto de elementos.

É uma organização de dados que permite identificar uma ordem: primeiro, segundo terceiro, quarto etc.

Os conjuntos **não** possuem o atributo ordem, mas **uma sequência é caracterizada pela ordem** de seus elementos. Essa ordem é definida pela indexação, isto é, pela associação de cada número natural positivo a um único elemento do conjunto.

Um conjunto **não** possui elementos repetidos, todos os seus elementos são distintos, mas **uma sequência pode ter elementos repetidos** como por exemplo a sequência 1,2,2,3,3,3,4,4,4,4,5,5,5,5 ou a sequência de letras ARARAQUARA, que tem os índices 1, 3, 5, 8 e 10 associados a uma mesma letra.

Uma sequência não necessariamente tem uma lei de formação que captura uma regularidade entre seus elementos. Há sequências conhecidas principalmente pela sua ordem, mas na verdade não possuem uma regra que permita obter ou “calcular” cada elemento. É o caso do alfabeto, uma sequência de símbolos (chamados letras) - A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z – que memorizamos nesta ordem (ordem alfabética).

A sequência W,I,R,K,L,I,C,H também é constituída de letras, tem outra ordem e tem repetição de letras.

A sequência numérica 5.0, 7.5, 5.5, 5.2, 8.0, 2.5, 3.7, 9.2, 10.0, 3.5, 3.5, 7.0, 7.0, 8.0, 9.5 tem por elementos somente números pertencentes ao intervalo [0,10], mas não tem nenhuma lei de formação que permita determinar o elemento seguinte na sequência.

Os nomes de diversas pessoas presentes em uma sala podem ser enumerados pela localização dessas pessoas na sala resultando numa sequência diferente da sequência em ordem alfabética.

Nem mesmo todas as sequências numéricas conhecidas têm obrigatoriamente uma fórmula que permite calcular seus elementos.

Uma sequência bastante conhecida é a dos números primos – 2, 3, 5, 7, 11, 13, 17... – e não há nenhuma fórmula conhecida que dê uma lei de formação para essa sequência.

2. Tipos Abstratos de Dados

Um dos passos essenciais no desenvolvimento de um programa computacional é a organização dos dados para o processamento. Para diversas aplicações, a escolha da estrutura de armazenamento de dados é a única grande decisão envolvida na implementação, pois uma vez escolhida, em geral os algoritmos são simples.

Uma estrutura de dados não pode ser somente um local de armazenamento, deve ser acompanhada de operações que permitam o processamento dos dados armazenados.

Esse par, valores + operações, é o que chamamos de Tipo de Dados.

Um tipo de dados é um conjunto de valores com uma coleção de operações para aplicar sobre esses valores.

Todo sistema computacional é baseado em camadas de abstração. Do bit à máquina, da linguagem de máquina ao programa em Java, são camadas de abstrações que permitem o estudo dos problemas e o desenvolvimento de novos mecanismos. Para desenvolver uma nova camada de abstração, precisamos definir os objetos abstratos que devem ser processados e suas operações. É preciso também representar esses dados em alguma estrutura de dados e implementar essas operações.

Um Tipo Abstrato de Dados (TAD) é um tipo de dados que somente pode ser acessado através de uma interface.

O programa que usa um TAD é um cliente e o programa que realiza as funcionalidades para o tipo de dados é uma implementação. A interface é opaca – o cliente não pode ver a implementação e podemos mudar a representação dos dados sem que seja necessário fornecer qualquer informação ao cliente.

Um tipo de dados necessário na grande maioria dos processamentos é a sequência de dados, uma vez que as comunicações e representações no mundo real são em grande parte organizadas sequencialmente.

Vamos definir um Tipo de Dados denominado **Lista** para capturar as necessidades básicas de um usuário desse tipo de dados.

3. Lista

Uma Lista é um conjunto finito de dados organizados sequencialmente, de modo que:

- É possível identificar a ordem dos elementos da lista, isto é, o primeiro, segundo etc.
- O número de elementos do conjunto é a ordem do último elemento da sequência.

As operações básicas de um objeto desse tipo de dados são:

- 1) Criar lista – representa a criação de um conjunto vazio ou sequência nula, isto é, sem elementos.
- 2) Obter tamanho da lista.
- 3) Verificar se a lista é vazia.
- 4) Inserir – tem por objetivo acrescentar um novo elemento à lista.
- 5) Remover – tem por objetivo remover um elemento localizado na lista.
- 6) Buscar – tem por objetivo localizar um elemento na lista.
- 7) Ordenar – tem por objetivo ordenar os elementos da lista em uma determinada ordem.
- 8) Obter elemento da lista que se encontra em uma determinada posição.

3. Representação de uma Lista por alocação sequencial (Tabela)

Nessa representação, todos os registros são de um mesmo tipo e encontram-se alocados em posições adjacentes, formando um bloco. Por essa razão, dizemos que o acesso aos dados é direto, uma vez que todos os endereços no bloco são determinados diretamente. Por exemplo:

memória					
6	1 2 1	2 5 2	1 5 3	2 1 1	4 4 3
4 1 2	2 4 4	Ana	Carlos	João	Pedro
3	10	42	H	Caio	5
10.0	8.6	7.5	4.0	2.5	567

Tabela A = (1 2 1, 2 5 2, 1 5 3, 2 1 1, 4 4 3, 4 1 2),

Tabela B = (Ana, Carlos, João)

Tabela C = (10.0, 8.6, 7.5, 4.0, 2.5).

Se a localização do primeiro registro (local onde se encontra o primeiro elemento da lista) é conhecida e se o tamanho de cada registro também é conhecido, as localizações de todos os demais registros são obtidas por um simples cálculo.

Toda lista armazenada em uma Tabela será nomeada com um nome de variável. O acesso a qualquer dos registros da lista será indicado pelo nome da Tabela seguido de um índice que indica a localização do registro na Tabela e é o número de ordem do elemento na sequência. Esse número não é armazenado, é usado somente para referenciar o local e a variável seguida desse índice é chamada variável indexada.

Por exemplo, a lista Ana, Carlos, João pode ser armazenada na Tabela A e os registros dessa Tabela são acessados pelas variáveis A[1], A[2], A[3]. O comprimento da lista é 3 e esse número também deve ser armazenado na estrutura de armazenamento de dados. Vamos designar por nA o local onde é armazenado o comprimento da lista. A notação usada em um algoritmo para a declaração de uma lista contendo nomes, armazenada em uma Tabela A, é: A(lista de string). A implementação deve capturar essa informação, de modo que ao acessarmos uma lista, automaticamente tenhamos acesso ao seu comprimento.