#### Estruturas Homogêneas

**Introdução**

Ao longo desta disciplina, aprenderemos que uma **ESTRUTURA DE DADOS** é uma **abstração** de objetos do mundo real, que temos interesse de representar na concepção de soluções baseadas em sistemas de informação. Uma **Estrutura de Dados** é composta:

1. Pela estrutura de armazenamento dos **atributos** relevantes do objeto em pauta; e
2. Pelas **operações** sobre estes atributos.

A estrutura de armazenamento representa os **aspectos estáticos** do objeto modelado, ou seja, **o que o objeto é,**  ou seu **estado**. Já as operações representam os **aspectos dinâmicos** do objeto, ou seja, **o que o objeto faz e como é feito,** ou seu **comportamento** .

Numa linguagem estruturada, como em C, os atributos são implementados por meio de **variáveis de tipos primitivos ou estruturados** (**registros** ( ***struct*** ) e **vetores**) enquanto que as operações são implementadas por ***funções***. Já numa linguagem orientada a objetos, como o Java ou Python, **Estruturas de Dados** são implementadas por meio de **classes**, onde os atributos são representados por **variáveis de instância** (de tipos primitivos ou estruturados) e as operações são implementadas por **métodos**.

Deste modo, inicia-se o presente curso buscando entender melhor o conceito de variável e de tipos de dados, primitivos (*built-in*) e estruturados.

Ao declarar uma variável (primitiva ou estruturada), o programador deve indicar:

. seu nome (que será utilizado pelo programador para referenciar a variável em seu programa); e

. seu tipo de dado (primitivo ou estruturado, que indica o conjunto de valores que pode ser armazenado nesta variável).

Um tipo de dado é chamado **primitivo** (ou ***built-in***) se ele já é fornecido nativamente pela linguagem de programação. Obviamente, o conjunto de tipos primitivos varia de tecnologia para tecnologia. Para efeito deste curso, considera-se os seguintes tipos primitivos:

- **int**: valores numéricos inteiros;

- **float**: valores numéricos fracionários ou reais;

- **char**: armazena um caractere (tabela ASCII);

- **string**: armazena uma cadeia de caracteres;

- **boolean**: armazena um valor lógico (V/F).

Já os tipos de dados **estruturados** são basicamente 2:

1. **Estruturas Homogêneas**: são estruturas formadas por dados de mesmo tipo, ou seja, **arranjos** (vetores – uma dimensão e matrizes – duas ou mais dimensões)

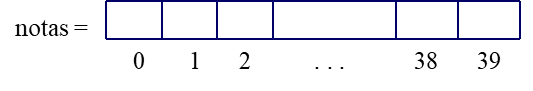
Para declarar um vetor, pode-se usar a seguinte sintaxe:

nome\_var [tamanho] : tipo\_dado\_base

Exemplo:

notas[40]: float

Este exemplo cria um vetor chamado notas, com 40 elementos do tipo float, indexados de 0..39, como abstraído na figura a seguir:

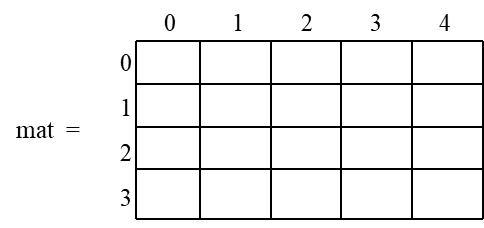


Para declarar uma matriz de duas ou mais dimensões, basta informar dentro dos colchetes o tamanho de cada dimensão, separados por vírgula.

Por exemplo, para declarar uma matriz de nome mat, com dimensão 4x5 (quatro linhas por cinco colunas), armazenando strings, basta declarar:

mat[4, 5] : string

Esta declaração cria uma estrutura abstraída na figura abaixo:



1. **Estruturas Heterogêneas**: permitem agrupar um ou mais elementos de diferentes tipos, primitivos ou estruturados, formando registros. Na próxima atividade, este tipo de dado será mais detalhado e explorado.

**Problemas a serem resolvido**

Para diagnosticar sua habilidade na manipulação de estruturas de dados homogêneos, contrua algoritmos para resolver os seguintes problemas:

1. Algoritmo que leia as notas de uma prova de determinada turma com 50 alunos, calcule a média da turma e mostre na tela as notas maiores que a média e suas respectivas posições no vetor.
2. Algoritmo que leia um conjunto X com N elementos reais e calcule a diferença entre o maior e o menor elemento existente, bem como as posições que os mesmos ocupam no conjunto. Esta diferença entre o maior e menor elementos de um conjunto de valores é denominada *amplitude*.
3. Algoritmo que leia um vetor X[20] e após, troque o 1o elemento pelo 20o, o 2o pelo 19o, e assim por diante. Ao final, imprima X.
4. Algoritmo que, considerando uma matriz 20x10 já carregada com valores inteiros, efetue a soma de cada linha e escreva as somas obtidas. Mostrar também os números das linhas que apresentarem a maior e menor somar.
5. Algoritmo para gerar a seguinte matriz M:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 01 | 02 | 03 | 04 | 05 |
| 06 | 07 | 08 | 09 | 10 |
| 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 |

1. Algoritmo que, dada uma matriz quadrada de ordem 10, verifique se esta é simétrica. Uma matriz M é simétrica se M[i, j] = M[j, i].