

## Unions e Enums:

### → Union:

Também como na struct é um agrupamento de variáveis, porém membros não podem coexistir, ou seja, usar um trecho de memória para tudo.

Union NomeUnião {

tipo membro;

tipo outro;

};

→ define a union

Union União nome;  
↕  
declara a união.

O espaço alocado na memória é do maior tipo que há na union.  
Uma atribuição na union, destrói outro membro já declarado, ou seja, sobrescreve os dados.

Unions economizam memória

Ela ainda pode ser usada com o typedef e criar um novo tipo.

### → Enum:

Define valores que uma variável pode assumir.

Se declara: enum nomeEnum { valores ... };

O compilador reserva como constante e atribui um inteiro.

Pode-se definir o valor inicial do contador: valor = x ou ainda  
fazer a requisição de redistribuição de variáveis.

Cinto sem pole - e mais adequado para os enuns.

Aluno: Gabriel F.F. de Souza

R. A: 2669480

Professor: Muriel Galvão