

Engenharia de Computação

Fundamentos de Programação

Aula 15 – União e Enumeração

Prof. Muriel de Souza Godoi
muriel@utfpr.edu.br

União

- Agrupamento de variáveis de tipos distintos que **NÃO podem coexistir**;
- Diferença entre **struct** e **union**:
- Estrutura (Struct)
 - Armazena, **simultaneamente** todos os seus atributos.
- União (Union)
 - Armazena apenas **um membro de cada vez**
 - Todos os membros compartilham o mesmo espaço de memória.

União (Union)

Sintaxe:

```
union NomeUniao{  
    tipo membro;  
    tipo membro;  
}; //union
```

Exemplo:

```
union Uniao{  
    char letra;  
    int numero;  
}; //union
```

- O código acima apenas define o tipo union, não declara nenhuma variável, como em uma struct
- Após sua definição, pode-se criar uma variável do tipo union

```
union Uniao nome;
```

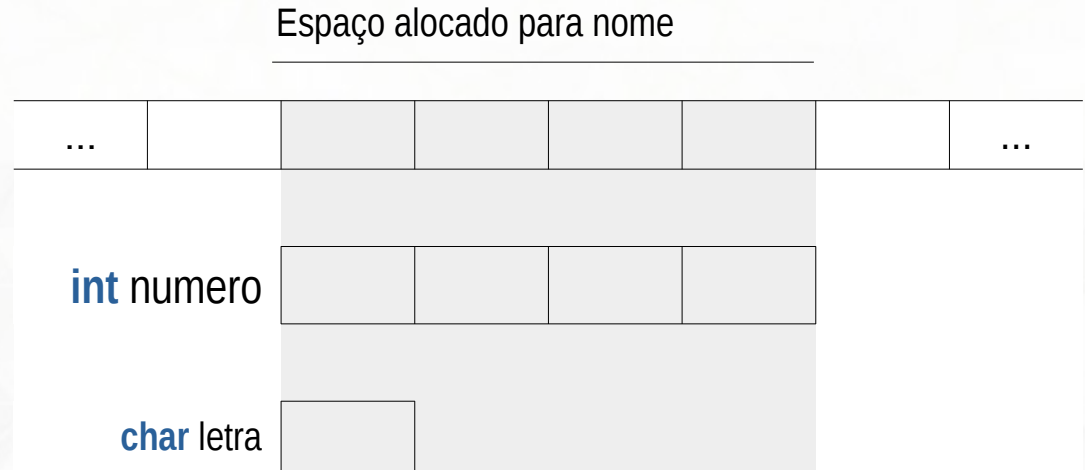
- Qual a quantidade de memória utilizada?

União (Union)

- O espaço usado por uma **union** ocupa o espaço necessário para armazenar o maior de seus campos

```
union Uniao{  
    int numero;  
    char letra;  
}; //union
```

```
union Uniao nome;
```



União (Union)

```
union Numero{
    int inteiro;
    float real;
};

int main(){
    union Numero numero;

    numero.real = 3.14;
    numero.inteiro = 15;

    printf("%f", numero.real);
    printf("%d", numero.inteiro);

    ...
}
```

- **Importante:**

- Os membros inteiro e real **compartilham a mesma posição** na memória
- Logo, a atribuição de um valor real **destrói** o valor inteiro e vice versa!

União (Union)

- Quando utilizar?
 - Quando não sabemos qual tipo de dados que uma variável irá armazenar;
 - Quando precisamos economizar memória.

Enumeração - Enum

- Define que uma variável poderá receber apenas um **conjunto restrito de valores** (constantes)
- É um tipo de dado definido pelo usuário
- **Sintaxe:**

```
enum nomeEnum { VALOR1, VALOR2, VALOR3, VALOR4, ... };
```

- Automaticamente o compilador lista as constantes como um valor inteiro, isto é:
 - VALOR1 = 0
 - VALOR2 = 1
 - VALOR3 = 2
 - VALOR4 = 3

Enumeração - Exemplo

```
enum Semana {DOMINGO, SEGUNDA, TERCA, QUARTA, QUINTA, SEXTA, SABADO};

int main(){

    enum Semana hoje;
    hoje = QUINTA;

    if(hoje == SEGUNDA)
        ...
    else if(hoje == TERCA)
        ...
    else if(hoje == QUARTA)
        ...

    return 0;
} //main
```


Enumeração - Exemplo

```
enum FormaPagamento {DINHEIRO, CARTAO, VALE_REFEICA0};
```

```
int main(){  
    enum FormaPagamento op;  
    op = DINHEIRO;
```

```
    switch(op){  
        case DINHEIRO:  
            ...  
        case CARTAO:  
            ...  
        case VALE_REFEICA0:  
            ...  
        default:  
            ...  
    }//switch  
    ..  
}//main
```

Equivalentes



```
    op = 0;  
  
    switch(op){  
        case 0:  
            ...  
        case 1:  
            ...  
        case 2:  
            ...  
        default:  
            ...  
    }//switch  
    ..
```

Enumeração - Enum

- Tecnicamente, declara nomes para valores inteiros:

```
enum nomeEnum { VALOR1, VALOR2, VALOR3, VALOR4, VALOR5 };
```

0 1 2 3 4

- Podemos começar a partir do 30:

```
enum nomeEnum { VALOR1=30, VALOR2, VALOR3, VALOR4, VALOR5 };
```

30 31 32 33 34

- Ou quaisquer valores desejados:

```
enum nomeEnum { VALOR1=15, VALOR2, VALOR3=45, VALOR4, VALOR5=70 };
```

15 16 45 46 70

Juntando tudo...

- Usar uma **Enum** para saber o valor salvo em uma **Union** que é membro de uma **Struct** definida usando **typedef**

```
typedef enum {PESO, VOLUME} TipoQuantidade;

typedef struct {

    char descricao[30];
    TipoQuantidade tipo;
    union {
        double volume;      // litros
        unsigned int peso;  // gramas
    } quantidade;

} Produto;
```

No início pode parecer confuso,
mas depois de um tempo de
prática você estará dominando
esses conceitos!



Exercício

- **1)** Crie uma enumeração representando os meses do ano. Agora, escreva um programa que leia um valor inteiro do teclado e exiba o nome do mês correspondente e a quantidade de dias que ele possui.
- **2)** Crie um programa de cadastro que receba, armazene, e em seguida, exiba os dados de 5 pessoas.
 - Cada pessoa possui: nome, idade, peso, data de nascimento, brasileiro ou estrangeiro. Caso seja Brasileiro, armazene o CPF, caso estrangeiro, armazene o passaporte.
 - **Regra:** Utilize structs, typedef, union e enum.