



UNIPAC

Universidade Presidente Antônio Carlos

Bacharelado em Ciência da Computação

Introdução a Programação

Material de Apoio

Parte XI – Struct e Enum

Prof. Nairon Neri Silva

naironsilva@unipac.br

1º sem / 2020

Material cedido pela professora Livia

Estruturas

- Coleção de variáveis referenciadas por um nome, fornecendo uma maneira conveniente de se ter informações relacionadas agrupadas. As variáveis que compreendem a estrutura são chamadas de membros, elementos ou campos.

Estruturas

- Sintaxe:

```
struct identificador {  
    tipo nome_variável;  
    tipo nome_variável;  
    ...  
} variáveis_estrutura;
```

- Onde: *variáveis_estrutura* são declarações de variáveis do tipo da estrutura.

Estruturas

- Exemplo:

```
#include <stdio.h>
void main (void)
{
    struct endereco
    {
        char rua;
        int num;
        char bairro, cidade;
    } pessoaA;

    pessoaA.num = 100;
    printf("%d", pessoaA.num);
}
```

Pode-se criar as structs fora do main(), neste caso as variáveis serão globais

Para acessar cada um dos campos da estrutura utiliza-se (ponto)
<nomedavariável>.<nomedocampo>

Estruturas

- Exemplo:

```
struct endereco
{
    char rua;
    int num;
    char bairro, cidade;
};
```

```
#include <stdio.h>
void main (void)
{
    struct endereco pessoaA;
    pessoaA.num = 100;
    printf("%d", pessoaA.num);
}
```

Definindo tipos - typedef

- A palavra reservada typedef nada mais é do que um atalho em C para que possamos nos referir a um determinado tipo existente com nomes sinônimos.
- Por exemplo, com o typedef, em vez de termos que nos referir como 'struct Aluno', poderíamos usar somente 'Aluno' para criar structs daquele tipo.

Definindo tipos - typedef

- Em vez de escrever sempre 'struct Funcionario', poderíamos escrever apenas 'Funcionario' e então declarar várias structs do tipo 'Funcionario'.
- Embora possamos criar atalhos com typedef para outros tipos, o typedef é comumente usado com structs.

Exemplo

```
struct {  
    int x;  
    int y;  
} ponto; //declara uma variável ponto (que é um par ordenado de inteiros)
```

Precedendo a declaração de um typedef:

```
typedef struct {  
    int x;  
    int y;  
} ponto;
```

```
void main() {  
    ponto a;  
    ...  
}
```

Agora ponto passa a ser o nome de um novo tipo. Esse tipo pode ser usado para declarar variáveis.

Exemplo

```
typedef struct{  
    int x;  
    int y;  
} ponto;
```

```
int main()  
{  
    ponto a;  
    a.x = 100;  
    a.y = 200;  
    printf("Pontos (x,y): (%d,%d) ", a.x, a.y);  
    return 0;  
}
```

Exercício não avaliativo

- 1) Construa uma estrutura com nome, número de matrícula e curso. Leia do usuário dados de 5 alunos, armazene em um vetor do tipo da estrutura criada e imprima os dados na tela.

Enumerações

- É um conjunto de constantes inteiras que especifica todos os valores legais que uma variável desse tipo pode ter.
- Sintaxe:

enum identificador { lista de enumeração };

Enumerações

- Cada símbolo representa um valor inteiro. Assim, podem ser usados em qualquer lugar em que um inteiro pode ser usado. A cada símbolo é dado um valor maior em uma unidade do precedente. O valor do primeiro símbolo da enumeração é 0.

Enumerações

- Exemplo:

```
#include <stdio.h>
#include <stdlib.h>
```

```
enum dias_da_semana {segunda, terça, quarta, quinta,
    sexta, sabado, domingo};
```

```
int main() {
    enum dias_da_semana d1,d2;
    d1=segunda;
    d2=sexta;
    if (d1==d2) {
        printf ("O dia e o mesmo.");
    }else{
        printf ("São dias diferentes.");
    }
    return(0);
}
```

Enumerações

```
enum meses_do_ano {Janeiro = 1, Fevereiro, Marco, Abril,
    Maio, Junho, Julho, Agosto, Setembro, Outubro, Novembro,
    Dezembro} meses;

int main(){
    printf("Digite o numero do mes: ");
    scanf("%d",&meses);
    if((meses >= Janeiro) && (meses <= Dezembro)){
        //switch que determina qual mes será impresso na tela
        switch(meses){
            case Janeiro:
                printf("%d - Janeiro",meses);
                break;
```

Enumerações

```
        case Fevereiro:
            printf("%d - Fevereiro", meses);
            break;
        ...
    }
}
else //senão estiver na faixa válida exibe mensagem
{
    printf("Valor INVALIDO!!!\n");
    printf("Os valores validos são \n\n");
    for(meses = Janeiro; meses <= Dezembro; meses++)
        printf("Mes: %d \n", meses);
}
return 0;
}
```

Loop que exibe a faixa de valores válidos (os valores da enum são na realidade inteiros)

Exercícios durante a aula (em grupo)

- 1) Escreva um algoritmo para fazer a criação de 2 novos tipos de dados conforme solicitado abaixo:
 - Data: composto de dia, mês e ano.
 - Compromisso: composto de uma data e texto que descreve o compromisso.

No método `main()` solicite ao usuário os dados do compromisso, armazene em uma variável do tipo compromisso e ao final imprima os dados presentes na variável.