

SISTEMAS COMPUTACIONAIS

Linguagem C: estruturas

typedef

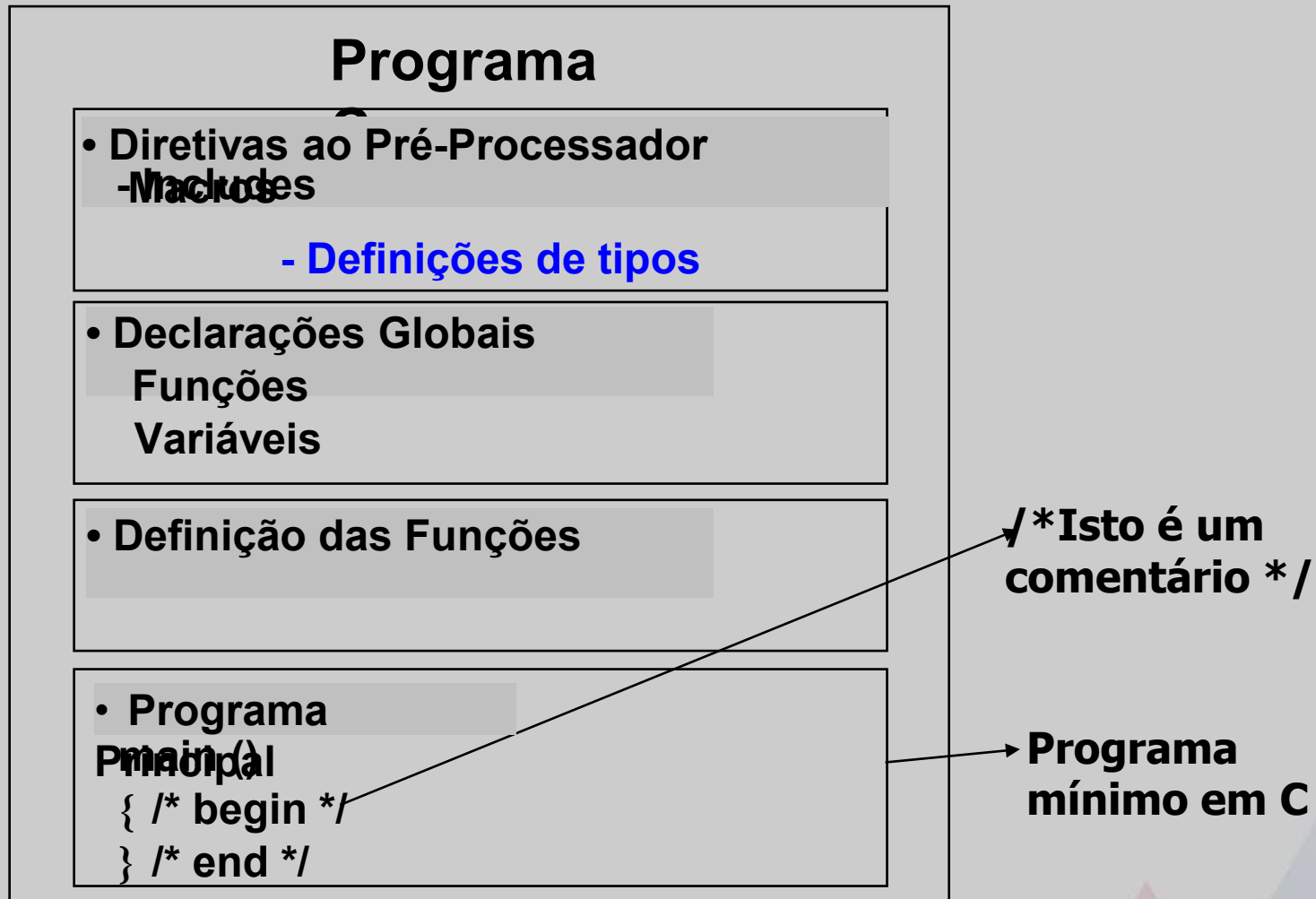
- Permite compor novos tipos de dados
- É composto a partir de tipos pré-existentes
- Não cria um novo tipo de dado!
- Forma geral:

typedef tipo novo_nome;

tipo: qualquer tipo válido em C

novo_nome: um identificador válido em C

typedef



EXEMPLO

```
#include <stdio.h>
typedef float num_real;
typedef int medida;
typedef medida altura;
altura alt=20;
int x=4, i;
int main (void){
    i = alt / x;
    return(0);
}
```

ESTRUTURA

- **É uma coleção de variáveis, possivelmente de diferentes tipos, organizadas em um único conjunto**
- **As variáveis que compõem uma estrutura são comumente chamadas de elementos (ou campos)**

DEFINIÇÃO

```
struct pessoa{  
    char nome[30];  
    int idade;  
};
```

A definição permite declarar variáveis cujo tipo seja uma pessoa

DECLARAÇÃO

```
struct pessoa pai, mae, tio, irmao;
```

ou

```
struct pessoa {  
    char nome[30];  
    int idade;  
}pai, mae, tio, irmao;
```

DECLARAÇÃO COM TYPEDEF

```
struct a{  
    int x;  
    char y; };  
typedef struct a MyStruct;  
MyStruct b; /*declaração da var b, cujo tipo é MyStruct*/
```

ou

```
typedef struct a {  
    int x;  
    char y;  
}MyStruct;  
MyStruct b; /*declaração da var b, cujo tipo é MyStruct*/
```


ACESSO A DADOS

```
struct a
{
    int x;
    char y;
}MyStruct;
```

```
int main (void){
    int num;
    MyStruct.x = 10;
    MyStruct.y = 'a';
    num = MyStruct.x;
}
```

ATRIBUIÇÃO

```
Struct S{
    int x;
    char y;
}a, b;
int main (void){
    a.x = 10;
    b = a;
    printf(“%d”, b.x); }
struct S b, a = {10, 'a'};
int main (void){
    b = a;
    printf(“%d %c”, b.x, b.y);
}
```

OPERAÇÕES

```
struct S{  
    int x;  
    char y;  
};  
struct S b, a = {10,'a'};  
int main (void){  
    b.x = a.x * 2;  
    b.y += 3;  
}
```

COMPOSIÇÃO

Estruturas podem ser campos de estruturas

```
struct ponto{
```

```
    int x;
```

```
    int y;
```

```
};
```

```
struct retangulo{
```

```
    struct ponto p1;
```

```
    struct ponto p2;
```

```
}ret;
```

Acesso aos dados:

```
ret.p1.x = 10;
```

```
ret.p1.y = 10;
```

```
ret.p2.x = 50;
```

```
ret.p2.y = 20;
```

ESTRUTURA E PONTEIRO

Estruturas podem ser autoreferenciadas:

```
struct ponto{  
    int x;  
    int y;  
    struct ponto * p;  
};
```

BIBLIOGRAFIA

- **SCHILDT, H., C Completo e Total, Makron Books, 1997. – Capítulo 2.**
- **KELLEY, A.; POHL, I., A Book on C, Addison-Wesley, quarta edição, 1998. – Cap. 1, 2 e 4.**
- **Damas, L. Linguagem C. Editora Grupo GEN. 10/2006. VBID 9788521632474**
- **Soffner, R. Algoritmos e Programação em Linguagem C, 1ª edição. Editora Saraiva. 08/2013. VBID 9788502207530**

SISTEMAS COMPUTACIONAIS

Linguagem C: estruturas