



# Estruturas de Dados Heterogêneas

**struct**

- *É um tipo criado pelo programador que permite combinar os tipos de dados distintos, nativos da linguagem.*
- As variáveis do tipo **struct** devem ser declaradas utilizando a palavra **struct** antes do nome do tipo definido pelo usuário.

- Exemplo:

```
struct Cavaleiro{  
    char nome[80];  
    int idade;  
    char armadura[80];  
    int atingiuSetimoSentido;  
};
```

*Cavaleiro*

*nome[80]*

*armadura[80]*

*atingiuSetimoSentido*

*idade*

# Campos da estrutura

- Cada informação dentro da ***struct*** é chamada de **CAMPO**.
- Os campos na ***struct*** podem ser acessados utilizando o operador ponto (.).
- Exemplo:  

```
struct Cavaleiro c;  
strcpy(c.armadura, "Pegasus");
```

# Parâmetros e retorno

- **struct** podem ser enviadas como parâmetros passados por **valor**, além de definirem o tipo de **retorno** para funções.

- Exemplo:

```
struct Cavaleiro Batalha(  
    struct Cavaleiro c1,  
    struct Cavaleiro c2  
) ;
```

- O comando ***typedef*** redefine o nome de um tipo. Pode ser utilizado para simplificar a declaração de ***struct***.
- Exemplo:  

```
typedef struct {  
    char nome[80];  
    int idade;  
    char armadura[80];  
    int atingiuSetimoSentido;  
} Cavaleiro;
```

# Comparação

```
struct Cavaleiro {  
    char nome[80];  
    int idade;  
    char armadura[80];  
    int atingiuSetimoSentido;  
};
```

```
void ExibirArmaduras  
    (struct Cavaleiro[12]);
```

## Vetor passado por referência

```
int main() {  
    struct Cavaleiro ouro[12];  
    . . .  
    ExibirArmaduras(ouro);  
    . . .  
}
```

```
typedef struct {  
    char nome[80];  
    int idade;  
    char armadura[80];  
    int atingiuSetimoSentido;  
} Cavaleiro;
```

```
void ExibirArmaduras  
    (Cavaleiro[12]);
```

```
int main() {  
    Cavaleiro ouro[12];  
    . . .  
    ExibirArmaduras(ouro);  
    . . .  
}
```



# Exemplo – Números complexos

- Exemplo:

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

```
typedef struct{
    float a, b;
    char forma; // 'c' ou 'p'
} Complexo;
```

```
Complexo LerComplexo();
Complexo CartesianoParaPolar(Complexo);
Complexo PolarParaCartesiano(Complexo);
Complexo SomarComplexos(Complexo, Complexo);
Complexo MultiplicarComplexos(Complexo, Complexo);
void ExibirComplexo(Complexo);
```

# Exemplo – Números complexos

```
int main()
{
    Complexo z1, z2, zS, zM;

    z1 = LerComplexo();
    z2 = LerComplexo();

    zS = SomarComplexos(z1, z2);
    ExibirComplexo(zS);

    zM = MultiplicarComplexos(z1, z2);
    ExibirComplexo(zM);

    return 0;
}
```

# Exemplo – Números complexos

```
Complexo LerComplexo()  
{  
    Complexo z;  
  
    printf("Forma (c)artesiana ou (p)olar: ");  
    fflush(stdin);  
    scanf("%c", &z.forma);  
  
    if (z.forma == 'c')  
        printf("Digite as componentes RE e IM:\n");  
    else  
        printf("Digite o modulo e a fase:\n");  
    scanf("%f", &z.a);  
    scanf("%f", &z.b);  
  
    return z;  
}
```

# Exemplo – Números complexos

```
Complexo CartesianoParaPolar(Complexo z)
{
    Complexo aux;
    aux.forma = 'p';
    aux.a = sqrt(pow(z.a,2.0) + pow(z.b,2.0));
    aux.b = atan(z.b/z.a);
    return aux;
}
```

```
Complexo PolarParaCartesiano(Complexo z)
{
    Complexo aux;
    aux.forma = 'c';
    aux.a = z.a*cos(z.b);
    aux.b = z.a*sin(z.b);
    return aux;
}
```

# Exemplo – Números complexos

```
Complexo SomarComplexos(Complexo z1, Complexo z2)
{
    Complexo aux;

    if (z1.forma == 'p')
        z1 = PolarParaCartesiano(z1);

    if (z2.forma == 'p')
        z2 = PolarParaCartesiano(z2);

    aux.forma = 'c';
    aux.a = z1.a + z2.a;
    aux.b = z1.b + z2.b;
    return aux;
}
```

# Exemplo – Números complexos

```
Complexo MultiplicarComplexos(  
    Complexo z1, Complexo z2)  
{  
    Complexo aux;  
  
    if (z1.forma == 'c')  
        z1 = CartesianoParaPolar(z1);  
  
    if (z2.forma == 'c')  
        z2 = CartesianoParaPolar(z2);  
  
    aux.forma = 'p';  
    aux.a = z1.a * z2.a;  
    aux.b = z1.b + z2.b;  
    return aux;  
}
```

# Exemplo – Números complexos

```
void ExibirComplexo (Complexo z) {  
    if (z.forma == 'c')  
        printf("%.4f%cj%.4f\n",  
                z.a,  
                z.b>0? '+' : '-',  
                fabs(z.b)  
        );  
    else  
        printf("%.4f<%.4f",  
                z.a,  
                z.b  
        );  
}
```

1.234-j5.678

3.210<0.123