



## Estruturas

SCC120 - Introdução à Ciência da Computação  
Profa. Roseli Ap. Francelin Romero

## Estruturas

### Variáveis Compostas Heterogêneas (1/12)

- As variáveis vistas até agora eram:
  - simples: definidas por tipo integer, real, char;
  - compostas homogêneas: definidas por array.
- Estruturas podem possuir variáveis de tipos diferentes e são declaradas da seguinte forma:

```
struct{  
    tipo1 campo1;  
    tipo2 campo2;  
    ...  
    tipon campon;  
}
```

WAN & SOR

## Estruturas

### Variáveis Compostas Heterogêneas (2/12)

- Uma estrutura pode ser visto como uma ficha de dados. Por exemplo,

INSCRIÇÃO	NOME	NÚMERO
pode ser integer, real, string, ...		

WAN & SOR

2

## Estruturas

### Variáveis Compostas Heterogêneas (2/12)

- Uma estrutura pode ser visto como uma ficha de dados. Por exemplo,

INSCRIÇÃO	NOME
	pode ser um vetor de 256 caracteres

WAN & SOR

3

## Estruturas

### Variáveis Compostas Heterogêneas (2/12)

- Uma estrutura pode ser visto como uma ficha de dados. Por exemplo,

INSCRIÇÃO	NOME	NÚMERO
	RUA	
		pode ser ser integer, vetor de 4 caracteres, ...

WAN & SOR

4

## Estruturas

### Variáveis Compostas Heterogêneas (2/12)

- Uma estrutura pode ser visto como uma ficha de dados. Por exemplo,

INSCRIÇÃO	NOME	NÚMERO
	RUA	
		pode ser ser integer, vetor de 4 caracteres, ...

- Esta ficha é representada na linguagem C da seguinte forma:

```
struct ficha{  
    int inscricao;  
    char nome[256];  
    char rua[64];  
    int numero;  
};
```

WAN & SOR

5

## Estruturas

### Variáveis Compostas Heterogêneas (3/12)

➤ Dada uma declaração de um novo tipo:

```

struct produto{
    char cod[8];
    char nome[30];
    int quant;
    float custo ;
};

produto prod;
  
```

➤ Como é feito o acesso às variáveis?

```

prod.cod = '12345678';
prod.nome = 'arroz';
prod.quant = 1000;
prod.custo = 0.82;
  
```

WAN & SOR 6

## Estruturas

### Variáveis Compostas Heterogêneas (5/12)

➤ Se, ao invés de uma única ficha, quisermos cadastrar n fichas?

➤ SOLUÇÃO: criar um vetor de estruturas!

➤ Como seria a declaração de um vetor do tipo ficha?

```

ficha fichas[5];
  
```

WAN & SOR 7



## Estruturas

### Variáveis Compostas Heterogêneas (6/12)

➤ Observações:

- ⊗ Estruturas (struct): define um "conjunto" de variáveis que podem ser de tipos diferentes;
- ⊗ Vetor (array): é uma "lista" de elementos de mesmo tipo. Por ex., a seguinte declaração:

```

ficha fichas[5];
  
```

declara uma variável lista contendo elementos do tipo ficha.

WAN & SOR 8

## Estruturas

### Variáveis Compostas Heterogêneas (7/12)

fichas

[1]	10547	JOSÉ DA SILVA	1840
[2]	27490	MARIA JOSÉ	8484
[3]	13859	MELISSA SOUTO	4393
[4]	9221	VALTER MERCADO	2424
[5]	23784	JOÃO FIGUEIRA	1234

WAN & SOR 9

## Estruturas

### Variáveis Compostas Heterogêneas (7/12)

produtos

[1]	10547	JOSÉ DA SILVA	1840
[2]	27490	MARIA JOSÉ	8484
[3]	13859	MELISSA SOUTO	4393
[4]	9221	VALTER MERCADO	2424
[5]	23784	JOÃO FIGUEIRA	1234

Como referenciar este campo?

WAN & SOR 10

## Estruturas

### Exercício 1 (8/12)

➤ Faça um programa para ler 10 fichas de empregados de uma empresa. Cada ficha possui o seguinte formato:

NUMEMP	NOMEMP		
ENDEREÇO	DATAEMP	SEXO	SALÁRIO

➤ Observações:

- ⊗ O campo DATAEMP deve ser dividido em 3 variáveis: dia, mês, ano;
- ⊗ O campo NUMEMP é do tipo inteiro;
- ⊗ O campo NOMEMP, ENDEREÇO é do tipo string;
- ⊗ O campo SEXO é do tipo caracter;
- ⊗ O campo SALÁRIO é do tipo real.

WAN & SOR 11

## Estruturas

### Exercício 1 - RESPOSTA (9/12)

```

struct ficha{
    int numemp;
    char nomemp[20], endereco[20];
    int dia, mes, ano; char sexo;
    float salario;
}
ficha fichas[10];
int main(){
    int i;
    for (i=0;i<10;i++){
        printf("\nentre com o numero do empregado");
        scanf("%d",&fichas[i].numemp);
        printf("\nentre com o nome do empregado");
        scanf("%s", fichas[i].nomemp);
        printf("\nentre com o endereco do empregado");
        scanf("%s",fichas[i].endereco);
        getchar(); //de string para inteiro eh necessario
        printf("\nentre com o dia da admissao");
        scanf("%d",&fichas[i].dia);
        printf("\nentre com o mes da admissao");
        scanf("%d",&fichas[i].mes);
        printf("\nentre com o ano da admissao");
        scanf("%d",&fichas[i].ano);
        printf("\nentre com o sexo do empregado");
        scanf("%c", &fichas[i].sexo); getchar();
        printf("\nentre com o salario do empregado");
        scanf("%f",&fichas[i].salario);
    }
}

```

WAN & SOR

12

```

for (i=0;i<10;i++){
    printf("%d\n",fichas[i].numemp);
    printf("%s\n", fichas[i].nomemp);
    printf("%s\n",fichas[i].endereco);
    printf("%d\n",fichas[i].dia);
    printf("%d\n",fichas[i].mes);
    printf("%d\n",fichas[i].ano);
    printf("%c\n", &fichas[i].sexo);

    printf("%f\n",fichas[i].salario);
}
getch();
}

```

WAN & SOR

13

```

struct dataemp{
    int dia;
    int mes;
    int ano;
};
struct ficha{
    int numemp;
    char nomemp[20], endereco[20];
    dataemp data;
    float salario;
};
ficha fichas[10];

```

WAN & SOR

14

```

int main(){
    int i;
    for (i=0;i<10;i++){
        printf("\nentre com o numero do empregado");
        scanf("%d",&fichas[i].numemp);
        printf("\nentre com o nome do empregado");
        scanf("%s", fichas[i].nomemp);
        printf("\nentre com o endereco do empregado");
        scanf("%s",fichas[i].endereco);
        getchar(); //de string para inteiro eh necessario
        printf("\nentre com o dia da admissao");
        scanf("%d",&fichas[i].data.dia);
        printf("\nentre com o mes da admissao");
        scanf("%d",&fichas[i].data.mes);
        printf("\nentre com o ano da admissao");
        scanf("%d",&fichas[i].data.ano);
        // getchar(); //de string para inteiro e real eh necessario
        printf("\nentre com o salario do empregado");
        scanf("%f",&fichas[i].salario);
    }
}

```

WAN & SOR

15

```

for (i=0;i<10;i++){
    printf("%d\n",fichas[i].numemp);
    printf("%s\n", fichas[i].nomemp);
    printf("%s\n",fichas[i].endereco);
    printf("%d\n",fichas[i].data.dia);
    printf("%d\n",fichas[i].data.mes);
    printf("%d\n",fichas[i].data.ano);
    printf("%f\n",fichas[i].salario);
}
getch();
}

```

WAN & SOR

16

## Vantagens de usar estruturas

```

#include <stdio.h>
struct dataemp{
    int dia;
    int mes;
    int ano;
};
struct ficha{
    int numemp;
    char nomemp[20], endereco[20];
    dataemp data;
    char sexo;
    float salario;
};
ficha fichas[10];
int main(){ int i;
    for (i=0;i<10;i++){
        printf("\nentre com o numero do empregado");
        scanf("%d",&fichas[i].numemp);
        getchar();
        printf("\nentre com o nome do empregado");
        scanf("%s", fichas[i].nomemp);
        gets(fichas[i].nomemp);
        printf("\nentre com o endereco do empregado");
        gets(fichas[i].endereco);
    }
}

```

```

getchar(); //de string para inteiro eh necessario
printf("\nentre com o dia da admissao");
scanf("%d",&fichas[i].data.dia);
printf("\nentre com o mes da admissao");
scanf("%d",&fichas[i].data.mes);
printf("\nentre com o ano da admissao");
scanf("%d",&fichas[i].data.ano);
printf("\nentre com o sexo do empregado");
scanf("%c",&fichas[i].sexo);
// getchar(); //de string para inteiro e real eh necessario
printf("\nentre com o salario do empregado");
scanf("%f",&fichas[i].salario);
}
// voce pode imprimir a estrutura de uma vez

```

### Vantagens de se usar estrutura

```

for (i=0;i<l;i++){
printf("%d\n",fichas[i].numemp);
printf("%s\n", fichas[i].nomemp);
printf("%s\n",fichas[i].endereço);
// Imprimindo a estrutura data de uma vez
printf("%d / %d / %d \n",fichas[i].data);
printf("%f\n",fichas[i].salario);
}
getch();
}

```

**Esto é válido para leitura? Verifique!!!!**

### Registros

Exercício 2 (10/12)

➤ Observação:

- ☒ Cada elemento do vetor fichas é um elemento do tipo ficha declarado anteriormente.

WAN & SOR 20

### Registros

Variáveis Compostas Heterogêneas (12/12)

➤ Uma atribuição que tenha a seguinte forma:

```

for (i=1;i<=9;i++){
fichas2[i] = fichas[i];
}

```

➤ pode ser feita? Sendo que fichas2 é do tipo ficha?


➤ Atribuições entre estruturas só podem ser feitas quando os campos são IGUAIS!

WAN & SOR 21

➤ Revisado por:

- ☒ Roseli Romero.

WAN & SOR 22



WAN & SOR 23