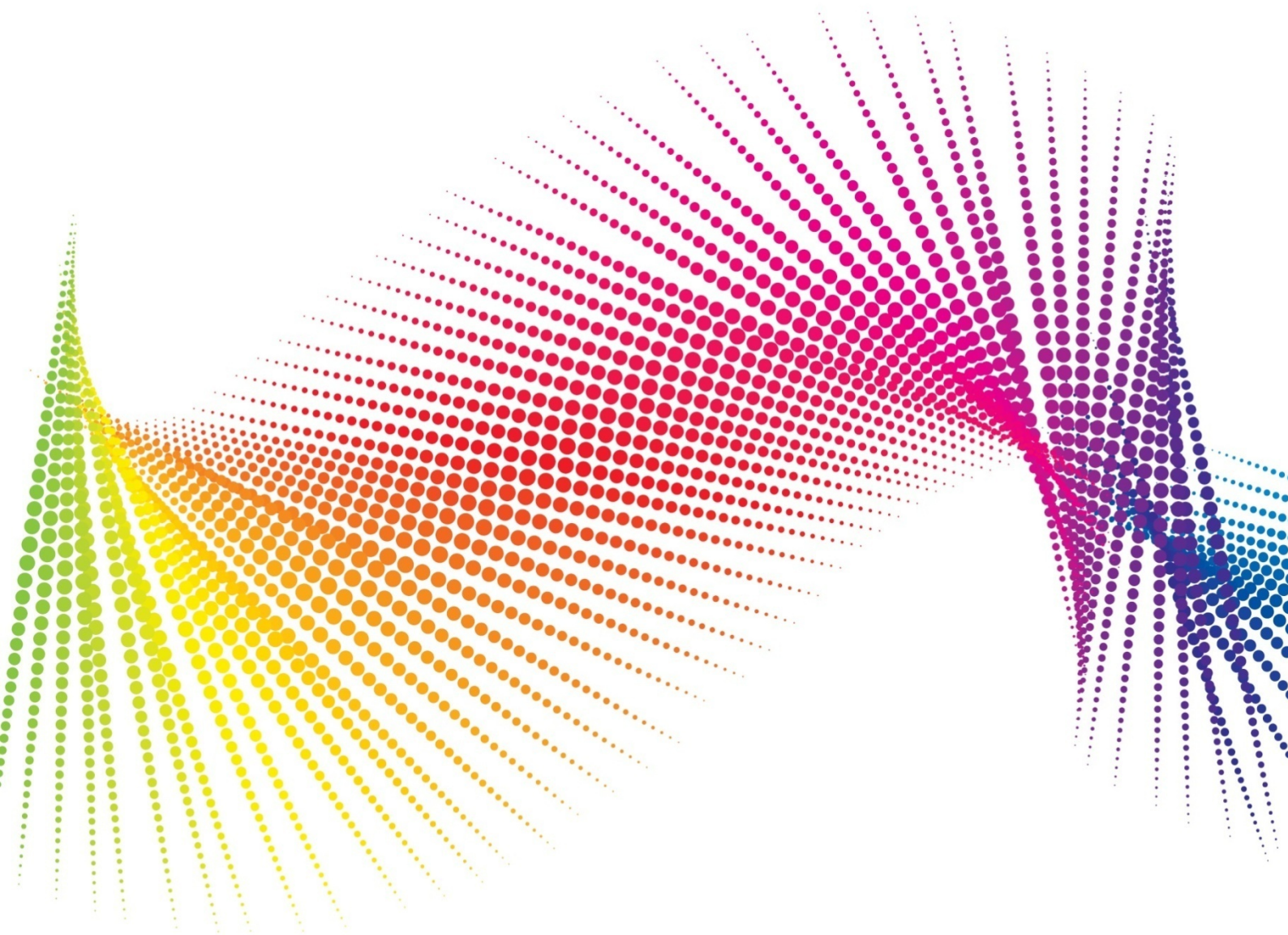


Estrutura de Dados

Aula 13



Este material é parte integrante da disciplina oferecida pela UNINOVE.

O acesso às atividades, conteúdos multimídia e interativo, encontros virtuais, fóruns de discussão e a comunicação com o professor devem ser feitos diretamente no ambiente virtual de aprendizagem UNINOVE.

Uso consciente do papel.

Cause boa impressão, imprima menos.

Aula 13: Estruturas (*struct*)

Objetivo: Estudar as estruturas, capazes de armazenar várias variáveis de tipos diferentes.

Conceitos básicos sobre tipos estruturados e estruturas

Caro aluno, antes de começarmos a tratar o tema desta aula, que são as estruturas, vamos recordar alguns conceitos importantes? Alguns deles você já deve saber, outros talvez ainda tenha dúvidas... Então, vamos lá: acesse o AVA e assista à animação. Esta animação faz parte da sequência desta aula e, portanto, é essencial para a sua aprendizagem.

Declaração de uma estrutura

Estruturas são agrupamentos de variáveis que podem ser de tipos diferentes, mas são referenciadas por um nome comum.

Podemos declarar uma variável como sendo do tipo estrutura. Também podemos declarar uma estrutura como sendo um novo tipo.

Sintaxe:

```
{ struct [nome] { tipo nome_variável1;  
                  tipo nome_variável2;  
                  . . .  
                  tipo nome_variáveln;  
                } [lista_de_variáveis];
```

e/ou

```

{
    struct [nome] [lista_de_variáveis];

    typedef struct [nome] {
        tipo nome_variável1;
        tipo nome_variável2;
        . . .
        tipo nome_variáveln;
    } [nome_tipo];

    [nome_tipo] [lista_de_variáveis];
}

```

Exemplo de definição da estrutura e declaração de variáveis após a definição

```

struct informacoes
{
    char nome[30];
    int dianascimento;
    int mesnascimento;
    int anonascimento;
};

struct informacoes info1, info2;

```

Exemplo de definição da estrutura e declaração de variáveis

```

struct informacoes
{
    char nome[30];
    int dia_nascimento;
    int mes_nascimento;
    int ano_nascimento;
} info_sobre_voce, info_sobre_mim;

```

Exemplo de definição de um tipo estrutura e declaração de variáveis

```

typedef struct f {
    char nome[20];
    int idade;
    float altura;
} ficha;

ficha ficha1, ficha2, ficha3;

```

Outro exemplo:

```
struct aluno { char nome[30];  
               float notaprova1;  
               float notaprova2;  
               float media;  
               } aluno1, aluno2;
```

Referência aos campos de uma estrutura

Os campos de uma estrutura são referenciados utilizando-se o “operador ponto”.

Exemplos:

```
printf("\nDigite o nome do aluno: ");  
gets(aluno1.nome);  
ficha1.idade = 18;  
info1.dianascimento = 11;
```

Inicialização de estruturas

Uma estrutura pode ser inicializada: a) na declaração dispondo-se os valores a serem atribuídos a cada campo entre chaves na ordem de declaração dos campos dentro da estrutura; b) campo a campo usando o operador ponto; c) usando o comando de atribuição, igualando-a a outra estrutura.

Exemplos:

```
struct informacoes info3 = {"Maria", 9, 9, 1999};  
  
strcpy(info2.nome, "Jose da Silva");  
info2.dianascimento = 31;  
info2.mesnascimento = 8;  
info2.anonascimento = 1991;  
info1 = info3;
```

Esta atribuição equivale a:

```
strcpy(info1.nome, info3.nome);  
info1.dianascimento=info3.dianascimento;  
info1.mesnascimento=info3.mesnascimento;  
info1.anonascimento=info3.anonascimento;
```

Declaração e inicialização de estruturas aninhadas

Um campo de uma estrutura pode ser ele próprio uma estrutura.

Exemplos:

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

// Uma reta é definida por dois pontos

```
struct reta  
{    struct ponto pt1,pt2;  
};
```

```
struct informacoes  
{    char nome[80];  
    struct {    int dia;  
                int mes;  
                int ano;  
            } datanascimento;  
} info4;
```

```
struct reta reta1, reta2;
```

```
reta1.pt1.x= 1;  
reta1.pt1.y=10;  
reta1.pt2.x= reta1.pt2.y=0;  
reta2=reta1;
```

```
info4.datanascimento.dia = 31;  
info4.datanascimento.mes = 12;
```

Exemplo de programa simples usando *struct*

```
/* este programa pega os dados:nome,telefone,altura e idade de uma pessoa
*/

#include <stdio.h>
#include <stdlib.h>
struct pessoa{
    char nome[50],tel[15];
    float altura;
    int idade;
};
main()
{
    struct pessoa dados;
    int i=0;
    printf("\n Digite os dados da pessoa:");
    printf("\n Nome:");
    fflush(stdin);
    gets(dados.nome);
    printf("\n Telefone:");
    fflush(stdin);
    gets(dados.tel);
    printf("\n Altura:");
    fflush(stdin);
    scanf("%f",&dados.altura);
    printf("\n Idade:");
    fflush(stdin);
    scanf("%d",&dados.idade);
    printf("\n Voce entrou com:");
    printf("\n Nome; %s",dados.nome);

    printf("\n Nome; %s",dados.tel);

    printf("\n Nome; %.2f",dados.altura);

    printf("\n Nome; %d",dados.idade);

    system("pause >> log");
}
```

Agora, caro aluno, vamos praticar resolvendo os exercícios propostos. **Leia a lista, resolva os exercícios e verifique seu conhecimento.** Caso fique alguma dúvida, leve a questão ao Fórum e divida com seus colegas e professor.

REFERÊNCIAS

SCHILDT, H. *C Completo e Total*. São Paulo: Makron Books, 1997.

TENEMBAUM, Aaron M., et al. *Estruturas de Dados usando C*. São Paulo: Pearson Makron Books, 1995.