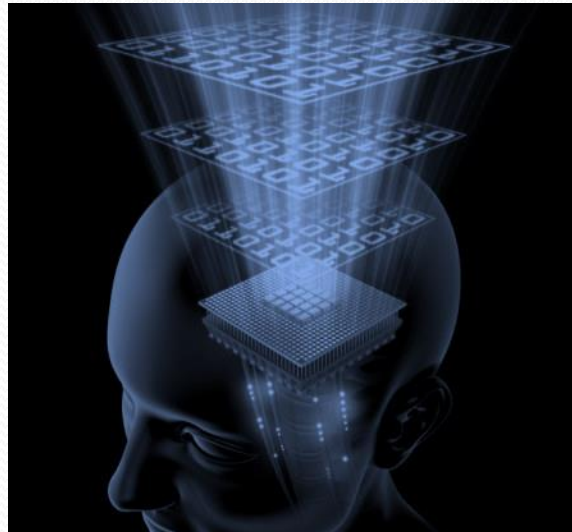




Disciplina: Algoritmos e Programação

Márcia Boell

marcia.boell@uffs.edu.br



Motivação

- Faça um programa que armazene os seguintes dados sobre um aluno:
 - Nome
 - Notas em 3 provas
 - Média final
- Provavelmente teremos:
 - `char nome[50]`
 - `float notas[3]`
 - `float media`
- Não existe nenhuma relação explícita entre as variáveis!

Estruturas

- Arranjos agrupam dados *homogêneos*
- **Estruturas** agrupam dados *heterogêneos*
 - Dados que não são do mesmo tipo
 - Mas se referem a **mesma entidade** (“coisa”)
- Dados são chamados de campos (ou *atributos*)

Estruturas: declaração

- Definição de uma estrutura:

```
struct <nome_estrutura>
{
    <tipo> <variavel_1>;
    <tipo> <variável_2>;
    ...
    <tipo> <variável_n>;
};
```

Campo 1
Campo 2
Campo *n*

- Estruturas são **tipos** complexos:
 - struct** <nome_estrutura> <nome_variavel_estrutura>;

Exemplo

```
struct Aluno
{
    char   nome[50];
    float  notas[3];
    float  media;
};
```

```
struct Aluno aluno1;
struct Aluno aluno2;
struct Aluno joaozinho;
```

Na memória...

aluno1

nome	nota			media
	[0]	[1]	[2]	

aluno2

nome	nota			media
	[0]	[1]	[2]	

Acesso

- Acesso aos campos é feito com o *operador ponto*
 - `<nome_variavel_estrutura>.<campo>`
- Usamos como se fosse um outra variável qualquer

```
#import <stdio.h>
#import <string.h>

struct Aluno
{
    char  nome[50];
    float notas[3];
    float media;
};

int main(){
    float mediaGeral;
    struct Aluno aluno1;
    struct Aluno aluno2;

    strcpy(aluno1.nome, "Maria");
    aluno1.notas[0] = 10;
    aluno1.notas[1] = 9;
    aluno1.notas[2] = 8;
    aluno1.media = (aluno1.notas[0] + aluno1.notas[1] + aluno1.notas[2]) / 3;

    gets(aluno2.nome);
    scanf("%f", &aluno2.notas[0]);
    scanf("%f", &aluno2.notas[1]);
    scanf("%f", &aluno2.notas[2]);
    aluno2.media = (aluno2.notas[0] + aluno2.notas[1] + aluno2.notas[2]) / 3;

    mediaGeral = (aluno1.media + aluno2.media)/2;

    return 0;
}
```


Memoria

aluno1

nome	nota			media
	[0]	[1]	[2]	

aluno2

nome	nota			media
	[0]	[1]	[2]	

Estruturas - Exemplo

Comando: `strcpy(aluno1.nome, "Maria");`

aluno1

nome	nota			media
	[0]	[1]	[2]	
Maria				

aluno2

nome	nota			media
	[0]	[1]	[2]	

Estruturas - Exemplo

Comando: `aluno1.nota[0] = 10;`

aluno1

nome	nota			media
	[0]	[1]	[2]	
Maria	10			

aluno2

nome	nota			media
	[0]	[1]	[2]	

Estruturas - Exemplo

Comando: `aluno1.nota[1] = 9; aluno1.nota[2] = 8;`

aluno1

nome	nota			media
	[0]	[1]	[2]	
Maria	10	9	8	

aluno2

nome	nota			media
	[0]	[1]	[2]	

Estruturas - Exemplo

Comando: `aluno1.media=(aluno1.nota[0]
+aluno1.nota[1]+aluno1.nota[2])/3;`

aluno1

nome	nota			media
	[0]	[1]	[2]	
Maria	10	9	8	9

aluno2

nome	nota			media
	[0]	[1]	[2]	

Estruturas em C - Utilização

- Ler dados de estrutura aluno2:
 - `gets(aluno2.nome);`
 - `scanf("%f", &aluno2.nota[0]);`
 - `scanf("%f", &aluno2.nota[1]);`
 - `scanf("%f", &aluno2.nota[2]);`
- Calcular média de aluno2:
 - `aluno2.media = (aluno2.nota[0] + aluno2.nota[1] +
aluno2.nota[2])/3;`

Estruturas - Exemplo

aluno1

nome	nota			media
	[0]	[1]	[2]	
Maria	10	9	8	9

aluno2

nome	nota			media
	[0]	[1]	[2]	
José	7	8	6	7

Atribuição

- Podemos fazer atribuição de structs
- Exemplo
 - `aluno1 = aluno2;`
 - Todos os campos de `aluno2` serão copiados para `aluno1`
- Atenção: *structs não são vetores!*

Vetores de Estruturas

- Podem ser utilizados vetores de estruturas
- Exemplo: *vetor com 4 alunos*
 - `struct Aluno alunos[4];`

Exemplo: média dos alunos

- Exemplo para cálculo das Médias dos alunos:

alunos

	nome	nota			media	
		[0]	[1]	[2]		
[0]	Maria	10	9	8		→ Registro 0
[1]	José	7	8	6		→ Registro 1
[2]	Pedro	7.5	8	7		→ Registro 2
[3]	Olir	5	8	9.5		→ Registro 3

Exemplo: média dos alunos

```
int i,n;

for(i=0; i<4; i++) {
    //alunos
    alunos[i].media=0;
    for(n=0; n<3; n++){
        //notas
        alunos[i].media = alunos[i].media + alunos[i].notas[n];
    }
    alunos[i].media= alunos[i].media / 3;
}
```

Exemplo

- Faça um programa que utilize a estrutura `Aluno` para armazenar dados sobre **100 alunos** de uma turma.
 - Os nomes dos alunos devem ser lidos do teclado
 - As notas dos alunos nas três avaliações devem ser lidas do teclado
 - As médias devem ser calculadas e armazenadas.
- Após, os nomes e as médias dos alunos da turma devem ser apresentados.

```

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

#define N_ALUNOS 100

struct Aluno {
    char nome[50];
    float nota[3];
    float media;
};

int main() {
    struct Aluno alunos[N_ALUNOS];

    int i,j;

```

```

    for(i=0; i<N_ALUNOS; i++) {
        printf("---%do. aluno\n", i+1);
        printf("Nome: ");
        gets(alunos[i].nome);

        alunos[i].media=0;
        for(j=0; j<3; j++) {
            printf("Nota da %da. avaliacao: ",j+1);
            scanf("%f", &alunos[i].nota[j]);
            alunos[i].media = alunos[i].media +
                               alunos[i].nota[j];
        }
        alunos[i].media=alunos[i].media/3;
    }

    printf("\n---Medias dos alunos da turma\n");

    for(i=0; i<N_ALUNOS; i++){
        printf("Nome: %s - ", alunos[i].nome);
        printf("Media: %.1f\n", alunos[i].media);
    }

    return 0;
}

```

Exercício

- Faça um programa que controle a tabela de pontos Campeonato Brasileiro.
 - Armazenar *nome* e *pontos obtidos* de cada time (20 times);
 - Determinar e exibir:
 - Campeão (maior número de pontos)
 - Último rebaixado (menor número de pontos)
- Crie uma estrutura para receber os nomes de clubes de futebol e seus respectivos pontos obtidos.

```

#include <stdio.h>
#include <stdlib.h>
#define N_TIMES 20

struct Time {
    char nome[50];
    int pontos;
};

int main(){
    struct Time times[N_TIMES];
    int i, imaior, imenor;

    for(i=0;i<N_TIMES;i++) {
        printf("Nome do clube
        (%i):",i+1);
        gets(times[i].nome);

        printf("Numero de pontos:");

        scanf("%d",&times[i].pontos);
    }

```

```

        imaior=0;
        imenor=0;

        for(i=0; i<N_TIMES; i++){
            if (times[i].pontos > times[imaior].pontos){
                imaior = i;
            }
            if (times[i].pontos < times[imenor].pontos){
                imenor = i;
            }
        }
        printf("\nTime Campeao: %s (obteve %d
        ponto(s))\n", times[imaior].nome,
        times[imaior].pontos);

        printf("\nTime Rebaixado: %s (obteve %d
        ponto(s))\n", times[imenor].nome,
        times[imenor].pontos);

        return 0;
    }

```