

Registros

Otávio Lemos

Pense num personagem em um jogo (Ex. Sonic)

- Como escrever a descrição do Sonic?
- Quero armazenar: argolas, tempo, vidas, continues:

```
int argolas;  
int tempo;  
int vidas;  
int continues;
```

- Ok, mas pode ficar melhor...

Registros



- Ideia: agrupar todos os dados em uma única variável.
- Usar registro. O que é isso?
- Uma estrutura de dados mais complexa do que os tipos primitivos → assim como vetor possui vários campos

Registros

Declaração

```
struct <tipo>{  
    <variaveis internas  
    pertencentes ao registro>  
} <variaveis>;
```

Registros

Exemplo

```
struct jogador {  
    int argolas;  
    int tempo;  
    int vidas;  
    int continues;  
} sonic;
```

Registros

Exemplo

```
struct data {  
    int dia;  
    int mes;  
    int ano;  
};  
  
struct data x;  
struct data y;
```

Registros

Como usar?



- Cada variável interna do registro é chamada de campo.
- No exemplo do sonic temos quatro campos: argolas, tempo, vidas e continues.
- O registro é acessado por meio de seus campos, nunca pela estrutura inteira.

Registros

Como acessar os campos?



- Usar a seguinte forma:

`<variavel de registro>.<campo>`

- Exemplo do sonic:

```
sonic.argolas = 0;  
sonic.tempo = 120;  
sonic.vidas = 3;  
sonic.continues = 1;
```


Registros

Como acessar os campos?



```
x.dia = 31;  
x.mes = 8;  
x.ano = 1998;
```

Registros

Exemplo

```
struct data fim_evento
    (struct data datainicio, int duracao) {
    struct data datafim;
    . . .
    . . .
    datafim.dia = ...
    datafim.mes = ...
    datafim.ano = ...
    return datafim;
}
```

Registros

Vetor de registro

- Registro é um tipo de variável como todos os outros → pode ser utilizado em um vetor
- Exemplo:

```
struct pessoa{  
    char nome[50];  
    char sexo;  
    int idade;  
} alunos[66];
```

Registros

Vetor de registro – acesso

- Sé se pode acessar um campo de um elemento do vetor de cada vez
- Exemplos:

```
alunos[3].sexo = 'M';  
sprintf(alunos[1].nome, "Giovane");
```

- Acessos inválidos:

```
alunos[2] = alunos[3];  
alunos.idade = 10;
```

Exercícios



- 1 Crie um tipo de registro que contenha os seguintes campos: Nome, RA, nota e frequencia;
- 2 Crie um vetor de variaveis deste registro;
- 3 Faça um programa que permita inserir remover, visualizar e buscar elementos do registro pelo RA.

Exercícios

- 4 Complete o código da função `fim_evento` acima;
- 5 Escreva uma função que receba dois registros do tipo `data`, cada um representando uma data válida, e devolva o número de dias que decorreram entre as duas datas.
- 6 Escreva uma função que receba um número inteiro que representa um intervalo de tempo medido em minutos e devolva o correspondente número de horas e minutos (por exemplo, converte 131 minutos em 2 horas e 11 minutos). Use um registro como o seguinte:

```
struct hm {  
    int horas;  
    int minutos;  
};
```