

|  |
| --- |
| Relatório de Estudo |
|  |
| 26 de abril  Professor: Adriano Rivolli,  Aluno: Thalles Garbelotti |

# Struct

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| As structs em C são estruturas heterogêneas que podem ser comparadas a um objetos de outras linguagens. São também conhecidas como registros e possuem campos que podem ser formados por qualquer tipo de dado.  Para utilização de structs em C é necessário seguir alguns passos:   1. Primeiro é necessário definir a estrutura e os campos que formam o tipo de dado desejado:  * A estrutura declarada deve ser iniciada com a palavra **struct** * O nome deve ser informado logo após a declaração da **struct** * Agora é possível abrir o bloco referente à estrutura criada e informar seus respectivos campos * Os campos da **struct** podem ser de qualquer tipo (desde os tipos nativos até vetores, matrizes e outras structs) e funcionam como variáveis simples em C  1. Após a estrutura ter sido declarada é possível instancia-la através de uma variável:  * A struct deve ser instanciada como uma variável do tipo definido * Para acessar um campo da struct utiliza-se o operador “."  Funções  |  | | --- | | Funções em C também são utilizadas para agrupar comandos com a finalidade de reaproveitar treixos de código. Esses métodos podem ser declarados através de **protótipos de função**, logo após os imports (includes) mas também pode-se trabalhar com funções de forma semelhante a outras linguagens como Java e Javascript, declarando-as em qualquer lugar do código exceto dentro do corpo de outra função.  Algumas características e boas práticas para funções em C foram explicadas em aulas e slides do professor Adriano Rivolli e adaptadas para estudo neste relatório:   * Funções devem possuir uma única tarefa * Podem ser declaradas dentro de outras funções * Os parâmetros, também chamados de argumentos das funções são entrada de dados de funções * O retorno corresponde a saída de dados * As variáveis de uma função são locais e só podem ser acessadas dentro desta * Ao chamar uma função o programa é “interrompido” até que a execução seja finalizada pois o código da função só é executado quando a função é instanciada * Uma função pode conter de zero a muitos parâmetros e todos estes parâmetros declarados devem ser passados para a instancia da função * Variáveis de qualquer tipo podem ser passadas como argumentos de funções * A instrução return termina a execução de uma função e volta ao programa que a instanciou * Caso a função não seja do tipo void, o tipo de retorno deve ser do mesmo tipo definido no cabeçalho da função. Além disso, esse valor pode ser armazenado em variáveis ou passados como valores para outras funções * Uma função pode conter várias instruções de retorno, porém, apenas uma instrução será executada pela função |  Ponteiros  |  | | --- | | Na linguagem C os ponteiros são variáveis responsáveis por armazenar endereços de memória de outras variáveis. Sua sintaxe é dada pelo uso de “\*” na frente do nome de uma variável e assim como toda variável o ponteiro possuí um tipo que é do mesmo tipo da variável referenciada pelo endereço que o ponteiro armazena (endereço da variável é acessado por “&” seguido do nome da variável).  Uma das principais finalidades do conceito de ponteiros é permitir que seja passado para as funções, os parâmetros que referenciam endereços de memória, substituindo assim a manipulação direta de variáveis globais.  Uma outra técnica importante da linguagem C que se utiliza de ponteiros é a Alocação Dinâmica de Memória. Essa técnica insiste em alocar blocos de memória em tempo de execução, passando como argumento para o malloc o tamanho em bytes a ser alocado. O valor contendo o tamanho em bytes passado para a função malloc é retornado pela função sizeof. A alocação dinâmica retorna então o endereço de memória alocado e é aí que entra o uso de ponteiros, para armazenar os endereços de memória que foram alocados. | |  |  Título  |  | | --- | | Texto do Subtítulo Aqui Para começar imediatamente, toque em qualquer texto de espaço reservado (como este) e comece a digitar para substituí-lo por um texto próprio.  Deseja inserir uma imagem de seus arquivos ou adicionar uma forma, caixa de texto ou tabela? É fácil! Na guia Inserir da faixa de opções, toque na opção desejada. | | *“Encontre ferramentas ainda mais fáceis de usar na guia Inserir, tais como para adicionar um hiperlink ou inserir um comentário”* | | Para começar imediatamente, toque em qualquer texto de espaço reservado (como este) e comece a digitar para substituí-lo por um texto próprio.  Deseja inserir uma imagem de seus arquivos ou adicionar uma forma, caixa de texto ou tabela? É fácil! Na guia Inserir da faixa de opções, toque na opção desejada. | |