Fatec
Araras

Estrutura de Dados

Prof. Orlando Saraiva Júnior
orlando.nascimento@fatec.sp.gov.br

"The most important single aspect of software development is to be clear about what you are trying to build."

Bjarne Stroustrup

Estrutura de Dados

Função

Funções

A experiência vem mosttrando que a melhor maneira de desenvolver e manter um programa grande é contruí-lo a partir de partes menores, ou **módulos**, ca um mais facilmente administrável que o programa original. Essa técnica é conhecida como **dividir e conquistar**.

Módulos em C e C++ são chamados de **funções**. Os programas nestas linguagens normalmente são escritos combinando-se novas funções com funções 'pré-definidas', disponíveis na **biblioteca-padrão**.

Funções

Fatec Araras

Você pode escrever funções que definam tarefas específicas. Estas chamadas são **funções definidas pelo programador**.

As funções são **chamadas** (ou **invocadas**) por uma chamada de função, que especifica o nome da função e oferece informações (como argumentos) de que a função chamada precisa.

A maioria das funções possui uma lista de **parâmetros** que oferecem meios de transmissão de informações entre as funções. Todas as variaveis descritas nas definições de função são **variáveis locais**. Os parâmetros de uma função também são variáveis locais desta função.

Funções

Fatec Araras

Você pode escrever funções que definam tarefas específicas. Estas chamadas são **funções definidas pelo programador**.

As funções são **chamadas** (ou **invocadas**) por uma chamada de função, que especifica o nome da função e oferece informações (como argumentos) de que a função chamada precisa.

A maioria das funções possui uma lista de **parâmetros** que oferecem meios de transmissão de informações entre as funções. Todas as variaveis descritas nas definições de função são **variáveis locais**. Os parâmetros de uma função também são variáveis locais desta função.

Protótipos de funções

O protótipo de função diz ao compilador o tipo de dado retornado pela função, o número de parâmetros que a função espera receber, os tipos dos parâmetros e a ordem esperada destes parâmetros.

Pilha de chamada de funções

Fatec Araras

Para entender como C/C++ realizam chamadas de função, precisamos considerar uma estrutura de daados conhecida como pilha. Quando um programa chama uma função, a função chamada precisa saber como retornar ao chamador, de forma que o endereço de retorno da função chamadora é colocado na **pilha de execução do programa**.

A pilha de execução do programa também contém a memória para as variáveis locais usadas em cada chamada de função durante a execução do programa.

Stack overflow

A quantidade de memória em um computador é finita. Se houver mais chamadas de função do que é possível armazenar nos registros de ativação da pilha de execução do programa, um erro conhecido como **estouro de pilha** (**stack overflow**) ocorrerá.

Chamando funções por valor e referência

Em muitas linguagens de programação, existem duas maneiras de se chamar funções – a **chamada por valor** e a **chamada por referência**.

Quando os argumentos são passados por valor, uma cópia do valor do argumento é feita e passada para a função chamada. As mudanças na cópia **NÃO** afetam o valor original da variável.

Quando um argumento é passado por referência, o chamador permite que a função chamada modifique o valor da variável original.

Recursão

Fatec Araras

Uma **função recursiva** é uma função que chama a si mesma direta ou indiretamente, por meio de outra função.

Fatec
Araras

Dúvidas

Prof. Orlando Saraiva Júnior
orlando.nascimento@fatec.sp.gov.br

Exercícios

1) Escreva um programa com uma função chamado soma. Este programa deve somar todos os valores entre 0 e n, onde n é o parâmetro passado para a função.

Exemplo, n= 5, o retorno da função será 15 (5+4+3+2+1)

Utilize laço de repetição.

2) Escreva um programa com uma função chamado soma. Este programa deve somar todos os valores entre 0 e n, onde n é o parâmetro passado para a função.

Exemplo, n= 5, o retorno da função será 15 (5+4+3+2+1)

Utilize recursividade