Aula – Programação II (INF 09330)

UFES
Informática

Subprogramas e Funções

Prof. Thiago Oliveira dos Santos
Departamento de Informática
Universidade Federal do Espírito Santo

2015

Visão Geral da Aula

UFES
Informática

- Introdução
- Subprogramas
- Chamada de Subprogramas
- Retorno de Subprogramas

Introdução

UFES
Informática

Resolvendo por Partes

- Problemas complexos podem ser divididos em partes menores
- Problemas menores são mais fáceis de resolver
- Com a solução dos problemas menores
  - O problema complexo se torna mais simples
- Abstração de conceitos
- Esse tipo de abordagem é denominado modularização
  - Se baseia na técnica de “dividir para conquistar”

Introdução

UFES
Informática

Exemplo

- Problema de montagem de um carro
  - É composto de várias etapas (funilaria, montagem, estofamento ...)
  - Poderia ser feito por um robô só
    - Porém, ele seria muito complexo
  - Outra solução seria utilizar vários robôs
    - Cada um com uma especialidade
    - Capaz de resolver um dos problemas
    - Esse robô seria muito mais simples e mais barato de se construir

Introdução

UFES
Informática

O que são subprogramas?

- Trecho de programa que realiza um operação computacional
- Executa parte de uma tarefa de um algoritmo maior
- Em C, subprogramas existem na forma de função

Por que usar?

- Separar código de acordo com a funcionalidade
- Abstrair conceitos
  - Focar no resultado sem pensar na implementação
- Facilitar leitura do código
- Facilitar a manutenção do código
- Aumentar a produtividade de criação de programas complexos

Introdução

UFES
Informática

Funções

- Similar ao conceito matemático
- Recebe variáveis (argumentos e parâmetros)
- Retorna valor

Subprogramas

UFES
Informática

Partes de um Subprograma

- Cabeçalho
- Dicionário de dados
- Corpo

Subprogramas

UFES
Informática

Partes de um Subprograma

- Cabeçalho
- Dicionário de dados
- Corpo

Sintaxe

- <tipo_de_retorno> Cabeçalho
  <nome_da_funcao>(<lista_de_parametros>){
  <variáveis> Dicionário de Dados
  <comandos> Corpo
  }

Cabeçalho

UFES
Informática

- Define
  - O nome do subprograma
    - Serve para identificá-lo
    - Assim como as variáveis, seus nomes devem ser auto explicativos
  - Seus parâmetros de entrada
    - Fazem a comunicação (entrada de dados) com o corpo da função
    - São os dados iniciais da função
  - E o tipo de seu retorno
    - Faz a comunicação (saída de dados) com o exterior da função
    - É o dado final (resultado) da função

Cabeçalho

UFES
Informática

- Exemplo
  - *float CalculaMedia(float a, float b)*
- Função para calcular a média entre dois números
- Recebe dois números reais *a* e *b*
- Retorna um número real (float)

Dicionário de Dados

UFES
Informática

- Onde se faz declaração das variáveis internas ao subprograma
  - São variáveis locais ao subprograma
  - Não são visíveis fora dele, mesmo que tenha o mesmo nome

Corpo

UFES
Informática

- Onde se descreve as instruções a serem executadas
- Funciona como um programa
  - Porem, realiza uma tarefa específica
- Funciona como bloco discreto de código
  - Só permite acesso as seus parâmetros e suas variáveis locais
  - Código da função é privativo da função
- Responsável por retornar o valor da função
  - Caso necessário

Corpo

UFES
Informática

- Parâmetros de entrada no corpo na função
  - Recebe dados de fora da função
  - Se comportam como variáveis locais a função
    - Criados na entrada da função
    - Já vem preenchidos com dados de fora da função
    - Destruídos na saída
  - Podem ser manipuladas localmente

Corpo

UFES
Informática

- Retorno de dados
  - É feito com o comando *return*
  - Faz a comunicação (saída de dados) com o exterior da função
  - É o dado final (resultado) da função
  - Uma função não é obrigada a retornar dados
    - Se ela não foi definhada para tal
    - Chamada do comando *return* pode ser sem valor

Corpo

UFES
Informática

- Exemplo de uma função com corpo definido
  - Cálculo da distancia entre dois pontos

```
float CalculaDistancia(float x1, float y1, float x2, float y2){
        float dx, dy, dist;

        dx = x2 – x1;
        dy = y2 – y1;
        dist = sqrt(dx*dx + dy*dy);
        return dist;
}
```

Chamada de Subprogramas

UFES
Informática

- Quando um (sub)programa solicita serviços de outro subprograma, diz-se que foi feita uma:
  - “chamada ao subprograma”
- Funções (Subprogramas) são chamados digitando:
  - O nome da função
  - Os dados (argumentos) a serem passados como parâmetros
  - Seu retorno de dados pode ou não ser utilizado
- Requer que o tipo dos argumentos seja igual ao declarado
  - Caso contrário, conversão implícita de tipo é feita

Chamada de Subprogramas

UFES
Informática

- Exemplo 1

```
int main() {
   float xa, xb, ya, yb, dist;

   printf("Forneca as coordenadas x e y (em km) das cidades:\n");
   scanf("%f%f%f%f", &xa, &xb, &ya, &yb);
   dist = CalculaDistancia(xa, ya, xb, yb);
   printf("Distancia (Km) entre as cidades: %f\n", dist);

   printf("Forneca as coordenadas x e y (em metros) dos moveis:\n");
   scanf("%f%f%f%f", &xa, &xb, &ya, &yb);
   dist = CalculaDistancia(xa, ya, xb, yb);
   printf("Distancia (metros) entre os moveis: %f\n", dist);
   return 0;
}
```

Chamada de Subprogramas

UFES
Informática

- Exemplo 2 (passagem de parâmetros por valor, cópia)

```
void ImprimeDobrado(int num, float numf){
    num = num*2;
    numf = numf*2;
    printf("Esse eh inteiro: %d, e esse eh float: %f!\n", num, numf);
}

int main()
{
   int x = 21;
   float y = 0.5;
   ImprimeDobrado(x, y);
   printf("Esse eh inteiro: %d, e esse eh float: %f!\n", x, y);
   return 0;
}
```

Chamada de Subprogramas

UFES
Informática

- Exemplo 3

```
void ImprimeAlgoSemParametro(){
    printf("Imprimindo nada de interessante\n");
}
void ImprimeAlgoComParametro(int num){
    num = num*2;
    printf("Imprimindo algo dobrado: %d\n", num);
}
void ImprimeAlgoDiferenteComParametro(int num){
    num = num*3;
    printf("Imprimindo algo triplicado: %d\n", num);
}
int main()
{
   int num = 2;
   ImprimeAlgoComParametro(num);
   ImprimeAlgoSemParametro();
   ImprimeAlgoDiferenteComParametro(num);
   return 0;
}
```

Tipos de Passagens de Parâmetros

UFES
Informática

- Em programação em geral
  - Por valor
    - Copia os dados de fora para dentro da função
    - Parâmetros alterados dentro da sub-rotina
      - Não alteram o valor do argumento fora da sub-rotina
  - Por referência
    - Trabalha com o dado alocado fora da função
    - Parâmetros alterados dentro da sub-rotina
      - Alteram o valor do argumento fora da sub-rotina

Tipos de Passagens de Parâmetros

UFES
Informática

- Em C, a passagem é sempre por valor
- Passagem por referência requer recursos adicionais
  - Visto mais adiante na disciplina
- Caso necessário alterar o valor de fora
  - Utilizar a variável de entrada para receber o retorno

```
int DobraValor(int num){
    return num*2;
}
int main()
{
   int x = 10;
   printf("X: %d\n", x);
   x = dobraValor(x);
   printf("X: %d\n", x);
   return 0;
}
```

Retorno de Funções Sem Valor

UFES
Informática

O Comando “return”

- Permite retornar da função
  - Com um valor, quando for o caso
- Exemplo de retorno sem valor

```
void TesteReturn(int num){
    if (!num)
       return;
    printf("Num dentro: %d\n", num);
}
int main()
{
   TesteReturn(0);
   TesteReturn(1);
   TesteReturn(2);
   return 0;
}
```

Retorno de Antecipado Funções

UFES
Informática

O Comando “return”

- Permite terminar uma função prematuramente
- Exemplo de retorno prematuro

```
int EhDivisor(int x, int y){
    if (!y)
      return -1;

    if (x % y){
       return 0;
    }
    return 1;
}
```

Protótipo de Funções

UFES
Informática

Características

- Força a verificação de tipos em tempo de compilação
- Permite referenciar funções antes de sua implementação

Exemplo

```
int TestPrototipo(int num); //Sem essa linha: RESULTADO IMPREVISIVEL
int main()
{
   TestPrototipo();
   TestPrototipo(0);
   return 0;
}

int TestPrototipo(int num){
    printf("%d\n", num);
    return 0;
}
```

Exercício

UFES
Informática

- Faça uma função que receba dois números inteiros positivos definindo os limites mínimo e máximo de um intervalo, e imprima o fatorial dos números pares entre eles (inclusive).

Exercício

UFES
Informática

- Faça uma função que receba dois números inteiros positivos definindo os limites mínimo e máximo de um intervalo, e imprima o fatorial dos números pares entre eles (inclusive).

```
void ImprimeFatorialDosParesEntreAeB(int a, int b){
   int i, fat;

   for(i = a; i <= b; i = i + 1){
      if ( ehPar(i) ){
         fat = fatorial(i);
         printf("O fatorial de %d eh %d\n", i, fat);
      }
   }

}
```

Exercício

UFES
Informática

- Faça uma função que receba dois números inteiros positivos definindo os limites mínimo e máximo de um intervalo, e imprima o fatorial dos números pares entre eles (inclusive).

```
int fatorial(int x){
   int fat;

   for(fat = 1; x > 1; x = x - 1)
      fat = fat * x;

   return fat;
}

int ehPar(int x){
   return !(x % 2);
}
```

Perguntas???

UFES
Informática

- Fazer exercícios da lista 3 e refazer lista 2 usando conceito de função!