Aula 6: Subrotinas (Funções e Procedimentos)

Professor(a): João Eduardo Montandon (103)
Virgínia Fernandes Mota (106)
jemaf.github.io
http://www.dcc.ufmg.br/∼virginiaferm

INTRODUÇÃO A PROGRAMAÇÃO - SETOR DE INFORMÁTICA

Subrotinas

- Um conceito simples: Subrotina é um parcela de código computacional que executa uma tarefa bem definida, sendo que essa tarefa pode ser executada (chamada) diversas vezes num mesmo programa.
- Motivação:
  - Necessidade de dividir um problema computacional em pequenas partes.
  - Os programadores verificaram que muitas destas pequenas partes se repetiam.
  - Ex: Impressão de mensagens, zerar um vetor, fazer uma operação matricial, etc.

Utilização de subrotinas

- Utilização de subrotinas:
  - Utilizar uma parte do código em várias partes do programa;
  - Vários programas irão utilizar os mesmos códigos (bibliotecas);
  - Abstrair a complexidade e facilitar o entendimento do programa.

Utilização de subrotinas

- Facilita a programação estruturada
  - dada as fases previstas nos refinamentos sucessivos decompõe-se o programa em módulos funcionais
  - tais módulos podem ser organizados/programados como subrotinas
  - **ou seja: viabiliza a modularização**

Características das subrotinas

- Executam uma tarefa bem definida
- Não funcionam sozinhas: devem ser chamadas por um programa principal ou por outra subrotina
- Permite a criação de variáveis próprias e a manipulação de variáveis externas (devidamente parametrizadas)
- Facilita a legibilidade do código através da:
  - estruturação (subrotinas são agrupadas fora do programa principal)
  - enxugamento (através de diversas chamadas da mesma subrotina)

Tipos de subrotinas

Existem dois tipos de subrotinas:

- Procedimentos: não retornam nenhum valor. São usadas para realizar alguma operação que não gera dados.
- Funções: retornam valor. São utilizadas para realizar uma operação e retornam alguma resposta relativa à operação realizada.

Procedimentos

```
1 void nomedoprocedimento (lista de parâmetros) {
2     declaração de variáveis
3     comandos
4 }
```

- nomedoprocedimento: Identifica a ação a ser executada no procedimento (SEM ESPAçOS EM BRANCO!) Ex: imprimeMedia
- lista de parâmetros: Valores recebidos como parâmetro. Ex: (A, B, 20, 30)
- declaração de variáveis: Variáveis necessárias para a codificação do procedimento, além das passadas na lista de parâmetros.
- comandos: comandos que implementam o procedimento desejado.

Procedimentos

```c
#include <stdio.h>
void imprimeMaior (int X, int Y) {
     if (X > Y)
          printf("%d", X);
     else
          printf("%d", Y);
}
int main() {
     int X, Y;
     scanf("%d %d", &X, &Y);
     imprimeMaior(X, Y);
     return 0;
}
```

Variáveis locais

- Toda variável pertencente ao procedimento é chamada de variável local, pois ela só pode ser utilizada dentro do escopo do procedimento.
- Fazem parte das variáveis locais de um procedimento:
  - as variáveis declaradas no procedimento;
  - todos os parâmetros recebidos pelo procedimento.

Parâmetros de uma subrotina

- **Chamada por valor**: é passado uma cópia da variável para a subrotina, ou seja, é feito uma cópia do argumento para o parâmetro. Qualquer alteração feita no parâmetro não reflete em alteração no argumento.
- **Chamada por referência**: todas as alterações realizadas no parm̂etro, refletem em alterações no argumento, ou seja, ambas as variáveis apontam para o mesmo endereço de memória. Para isso, é necessário que seja passado o endereço do argumento e o parâmetro receba-o na forma de ponteiro.

Parâmetros de uma subrotina

```c
#include <stdio.h>
void imprimeMaior (int X, int Y, int *Z) {
     if (X > Y)
          *Z = X;
     else
          *Z = Y;
}
int main() {
     int A, B, C;
     scanf("%d %d", &A, &B);
     imprimeMaior(A, B, &C);
     printf("%d", C);
     return 0;
}
```

Funções

- é um tipo especial de procedimento.
- **Retorna** como resultado o valor calculado pela função, que deve ser do tipo básico definido.

```
1 tipo nomedafuncao (lista de parâmetros) {
2     declaração de variáveis
3     comandos
4     return valordoretorno;
5 }
```

Funções

```
1 tipo nomedafuncao (lista de parâmetros) {
2     declaração de variáveis
3     comandos
4     return valordoretorno;
5 }
```

- tipo: tipo do dado a ser retornado como resultado da execução da função.
- nomedafuncao: nome que identifique a ação a ser executada na função.
- lista de parâmetros: valores recebidos como parâmetro.
- declaração de variáveis e comandos: variáveis locais e sequência de comandos
- return valordoretorno: a função permite retornar um valor, resultado das ações nela programadas. Este valor deve ser do tipo declarado antes do nome da função.

Funções

```c
#include <stdio.h>
int soma (int X, int Y) {
    return (X+Y);
}
int main() {
    int A, B, C;
    scanf("%d %d", &A, &B);
    C = soma(A, B);
    printf("%d", C);
    return 0;
}
```

Exemplos

O que será impresso no programa abaixo?

```c
#include <stdio.h>
void calculo (int *p, int *q) {
    *p = *p * 10;
    *q = *q + 10;
}

int main() {
    int x = 2, y = 5;
    calculo(&x, &y);
    printf("%d %d", x, y);
    return 0;
}
```

Exemplos

O que será impresso no programa abaixo?

```c
#include <stdio.h>
int calculo (int p, int q) {
    p = p * 10;
    q = q + 10;
    return (p + q);
}

int main() {
    int x = 2, y = 5;
    printf("%d %d %d", x, y, calculo(x, y));
    return 0;
}
```

Exemplos

O que será impresso no programa abaixo?

```c
#include <stdio.h>
int cal (int p, int q, int *r) {
    p = p * 10;
    q = q + 10;
    *r = *r - 10;
    return (p);
}

int main() {
    int x = 2, y = 5, z = 3, r;
    r = cal(x, y, &z);
    printf("%d %d %d %d", x, y, z, r);
    return 0;
}
```

Exercícios

1. Faça um programa que apresente o seguinte menu para o usuário:
Escolha uma opção de cálculo para dois números:
1) Soma
2) Produto
3) Quociente
4) Sair
Opção:
O menu acima deve ser apresentado para o usuário enquanto ele não escolher a opção 4 (sair do programa). O usuário fornecerá 2 números se escolher as opções de cálculo 1, 2 ou 3. Para cada opção de cálculo deve existir (obrigatoriamente) uma função definida (soma, produto e quociente dos dois números fornecidos pelo usuário). O resultado do cálculo deve ser escrito na tela.

Exercícios

2. Faça uma função que receba a idade de uma pessoa em anos, meses e dias e retorna essa idade expressa em dias.
3. Faça um procedimento que receba por parâmetro o tempo de duração de um experimento expresso em segundos e imprima na tela esse mesmo tempo em horas, minutos e segundos.
4. Escreva uma programa que calcule e imprima o quadrado de um número. O cálculo deve ser feito por uma função. Repita a operação até que seja lido um número igual a zero.
5. Faça uma função que receba um valor N inteiro e positivo e que calcula o fatorial deste valor. Retorne o resultado.
6. Seja o jogo de cartas para duas pessoas descrito a seguir. São distribuídas 3 cartas para cada jogador. O vencedor do jogo é determinado por quem possui a maior carta. Os naipes não precisam ser considerados. Faça um algoritmo que leia as cartas de cada jogador e determine quem foi o vencedor da partida. Considere que pode acontecer empate. Use uma função para determinar a maior carta.

Na próxima aula...

Vetores