Modularização

express

Aquilo que escuto eu esqueço,
Aquilo que vejo eu lembro,
Aquilo que faço eu aprendo.

Confúcio

”

1.
Módulos

Uma palavrinha

Módulos

- É a base da **Programação Estruturada**: divisão de um problema em **partes**.
- A cada uma dessas partes é dado o nome de **módulo**.
- **Modularização** é uma **técnica** para desenvolver programas, por meio de **refinamentos sucessivos**.
- Redução de um problema a um conjunto de tarefas bem específicas, destinadas a solucioná-lo de maneira eficiente
- Vantagens: reuso, portabilidade, facilidade para manutenção, organização
- Podem ser procedimentos ou funções
- Em Java não tem distinção. São métodos.

Procedimentos

- São módulos ou sub-rotinas que são chamados a partir de um outro programa (normalmente o programa principal).
- Depois de serem executadas, o fluxo de execução volta para o programa chamador.
- Não retorna nenhum resultado para o programa que o chamou
- Exemplos:
  - Gravar dados em arquivos
  - Enviar dados para o dispositivo de saída
  - Ordenar que o computador desligue

Funções

- São módulos ou sub-rotinas que são chamados a partir de um outro programa (normalmente o programa principal)
- Depois de serem executadas, o fluxo de execução volta para o programa chamador.
- Sempre retorna um resultado para o programa que o chamou, devendo portanto ser “consumido” pelo chamador. Ou armazenando o resultado na memória, ou transferindo para o dispositivo de saída.
- Exemplos:
  - Mostrar um menu e retornar a opção
  - Solicitar uma entrada ao usuário
  - Efetuar uma operação matemática

Argumentos ou Parâmetros

- Podem ser utilizados tanto em procedimentos quanto em funções
- Servem para informar valores de dados que serão necessários para executar a tarefa solicitada
- Exemplo: printf(**“Olá Mundo!”**)

Passagem de parâmetros

- É o mecanismo de informar sobre quais valores o processamento definido na função deve ser realizado
- Os parâmetros são passados para um método de acordo com a sua posição
- A cada chamada de método, os parâmetros são copiados e estas cópias são utilizadas na execução do mesmo

2.
Sintaxe Geral

Sintaxe geral

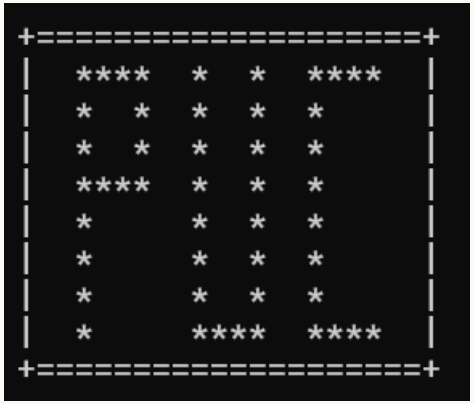
```
//sem parâmetros sem retorno
void NomeMetodo(){
    <corpo da função>
}
//com parâmetros sem retorno
void NomeMetodo(<tipo> arg1, .., <tipo> argN){
    <corpo da função>
}
//sem parâmetros com retorno
<tipo de retorno> NomeMetodo(){
    <corpo da função>
    return valorDeRetorno;
}
//com parâmetros com retorno
<tipo de retorno> NomeMetodo(<tipo> arg1, .., <tipo> argN){
    <corpo da função>
    return valorDeRetorno;
}
```

3.
Exercícios

Exemplos

sem Parâmetros sem Retorno

1. Crie um método com o nome **Cabecalho** que mostre o seguinte layout na tela:



```c
void Cabecalho(){
    printf("\n+======================+\n");
    printf("|   ****   *   *   ****   |\n");
    printf("|   *   *   *   *   *      |\n");
    printf("|   *   *   *   *   *      |\n");
    printf("|   ****   *   *   *      |\n");
    printf("|   *      *   *   *      |\n");
    printf("|   *      *   *   *      |\n");
    printf("|   *      *   *   *      |\n");
    printf("|   *      ****   ****   |\n");
    printf("+======================+\n\n");
}
```

sem Parâmetros com Retorno

2. Crie um método com o nome **SegundosDia** que retorne a quantidade de segundos que existem em um dia.

```
int SegundosDia(){
    int segundosDia;
    segundosDia = 24 * 60 * 60;
    return segundosDia;
}
```

com Parâmetros sem Retorno

3. Crie um método com o nome **Rodape** que receba uma mensagem motivacional e mostre o seguinte layout:

```
********** MENSAGEM MOTIVACIONAL **********
```

```
void Rodape(char mensagem[50]){
    printf("\n\n********** %s **********", mensagem);
}
```

com Parâmetros com Retorno

4. Crie um método com o nome **CalculaIdade** que receba o ano atual e o ano de nascimento e retorne a idade de uma pessoa

```
int CalculaIdade(int anoAtual, int anoNasc){
    int idade;
    idade = anoAtual - anoNasc;
    return idade;
}
```

Programa Principal

5. Crie um programa com o nome **Principal** que receba o nome, o ano de nascimento de uma pessoa e a mensagem motivacional e calcule a quantidade de segundos vividos por ela. Invoque todos os métodos criados nos exercícios anteriores. Esse deve ser o layout de saída:

```
Digite seu nome: lucilia
Digite seu ano de nascimento: 1969
Mensagem motivacional: continue a nadar

+====================+
|  ****  *  *  ****  |
|  *  *  *  *  *     |
|  *  *  *  *  *     |
|  ****  *  *  *     |
|  *     *  *  *     |
|  *     *  *  *     |
|  *     *  *  *     |
|  *     ****  ****  |
+====================+



lucilia,
VocÛ jß viveu 1734480000 segundos

********** continue a nadar **********
```

```c
int main() {
    char nome[10], msg[50];
    int anoAtual = 2024, anoNasc, idade, segVividos;

    printf("\n\nDigite seu nome: ");
    scanf("%s", nome);
    printf("Digite seu ano de nascimento: ");
    scanf("%d", &anoNasc);
    fflush(stdin); //limpa buffer do teclado -> setbuf(stdin, NULL)
    printf("Mensagem motivacional: ");
    scanf("%[^\n]", msg);

    idade = CalculaIdade(anoAtual, anoNasc);
    segVividos = idade * 365 * SegundosDia();
    Cabecalho();
    printf("\n\n%s,\nVocê já viveu %d segundos", nome, segVividos);
    Rodape(msg);
}
```

Obrigada!

Alguma Pergunta?

professora@lucilia.com.br

Créditos

- Template: SlidesCarnival
- Ilustrações: Köhler's Medizinal-Pflanzen in naturgetreuen at BHL
- Ascencio: “Fundamentos de programação”