Introdução à Lógica de Programação

Prof. Lucas Amparo Barbosa

Semestre letivo 2020.2

CONFORME O CONHECIMENTO CRESCE...

- Códigos mais complexos
- Aumento no número de linhas de código
- Muitas funcionalidades repetidas
- Como melhorar essa situação?

CONFORME O CONHECIMENTO CRESCE...

- Podemos organizar pedaços de códigos em estruturas que serão utilizadas de forma repetitiva
  - São as **funções** e os **procedimentos**

- Essas estruturas são executadas a partir de uma chamada no código
  - escreval é um procedimento do VisuALG
  - pow(base, expoente) é uma função do C++

- Qual a diferença entre elas?
  - Funções executam o código e retornam uma resposta
  - Procedimentos não retornam nada

MELHORANDO UM POUCO O CÓDIGO...

```
int main() {
  int n;
  cout << "Informe Quantas pessoas deseja cadastrar: ";
  cin >> n;

  pessoa cadastro[n];

  // Cadastra
  for (int i = 0; i < n; i++) {
    cin.ignore();
    cout << "Cadastro N: " << (i+1) << endl;
    cout << "Digite o nome: ";
    getline(cin, cadastro[i].nome);

    cout << "Digite a idade: ";
    cin >> cadastro[i].idade;

    cout << "Digite a altura: ";
    cin >> cadastro[i].altura;

    cout << "Digite o peso: ";
    cin >> cadastro[i].peso;
  }

  // Apresenta
  for (int i = 0; i < n; i++) {
    cout << "Cadastro N° " << (i+1) << endl;
    cout << "Nome: " << cadastro[i].nome << endl;
    cout << "Idade: " << cadastro[i].idade << endl;
    cout << "Altura: " << cadastro[i].altura << endl;
    cout << "Peso: " << cadastro[i].peso << endl;
  }
}
```

- É possível melhorar um pouco esse código?
- O FOR já melhora bastante a legibilidade do código minimizando as repetições
- Porém, é possível deixar ainda melhor

MELHORANDO UM POUCO O CÓDIGO...

```
void apresentaCadastro(pessoa p, int contador) {
    cout << "Cadastro N° " << contador << endl;
    cout << "Nome: " << p.nome << endl;
    cout << "Idade: " << p.idade << endl;
    cout << "Altura: " << p.altura << endl;
    cout << "Peso: " << p.peso << endl;
}
```

- Um procedimento para apresentar os dados
- Uma função para cadastrar os dados

```
pessoa executaCadastro() {
    pessoa p;
    cout << "Digite o nome: ";
    getline(cin, p.nome);

    cout << "Digite a idade: ";
    cin >> p.idade;

    cout << "Digite a altura: ";
    cin >> p.altura;

    cout << "Digite o peso: ";
    cin >> p.peso;

    return p;
}
```

```
int main() {
  int n;
  cout << "Informe Quantas pessoas deseja cadastrar: ";
  cin >> n;

  pessoa cadastro[n];

  for (int i = 0; i < n; i++) {
    cin.ignore();
    cout << "Cadastro N: " << (i+1) << endl;
    cadastro[i] = executaCadastro();
  }

  for (int i = 0; i < n; i++) {
    apresentaCadastro(cadastro[i], i+1);
  }
}
```

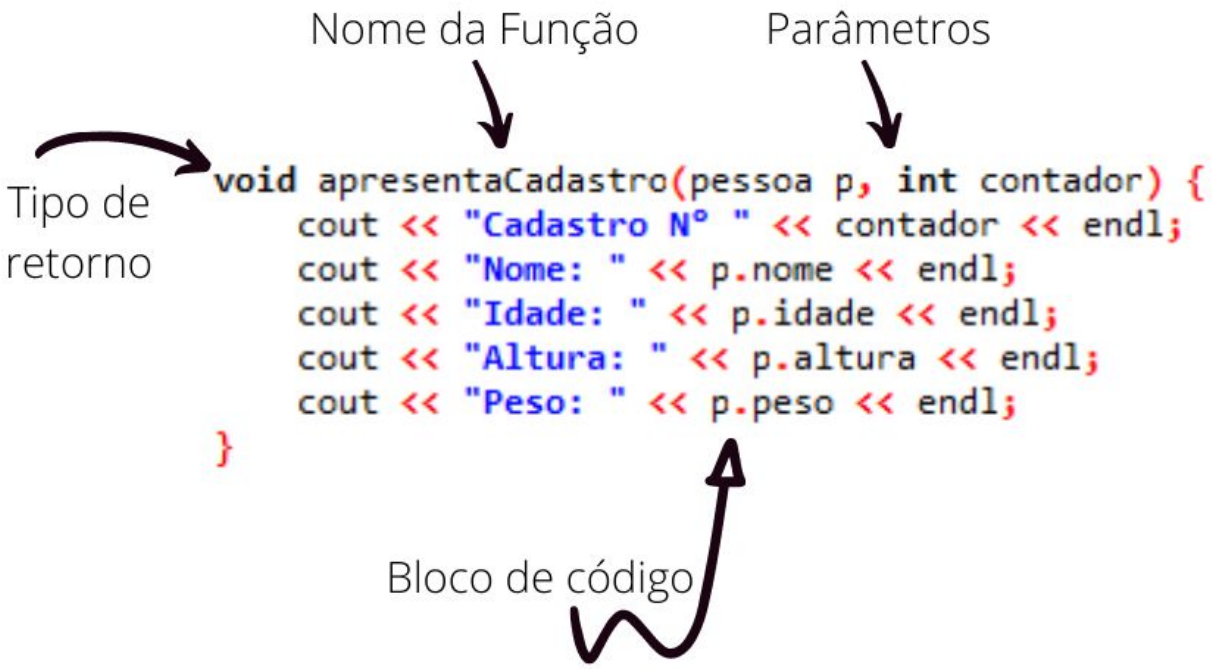
PORQUE UTILIZAR FUNÇÕES E PROCEDIMENTOS?

- Diminuir o tamanho dos blocos de código
- Auxilia na compreensão do mesmo
- Facilita a leitura do código
- Modularizar o sistema
- Evita repetição de código
- Auxilia na manutenção do código

COMO DECLARAR UMA FUNÇÃO/PROCEDIMENTO?

```
tipo_do_retorno nome_da_funcao(parametro) {
  //codigo que você deseja inserir
}
```

COMO DECLARAR UMA FUNÇÃO/PROCEDIMENTO?



COMO DECLARAR UMA FUNÇÃO/PROCEDIMENTO?

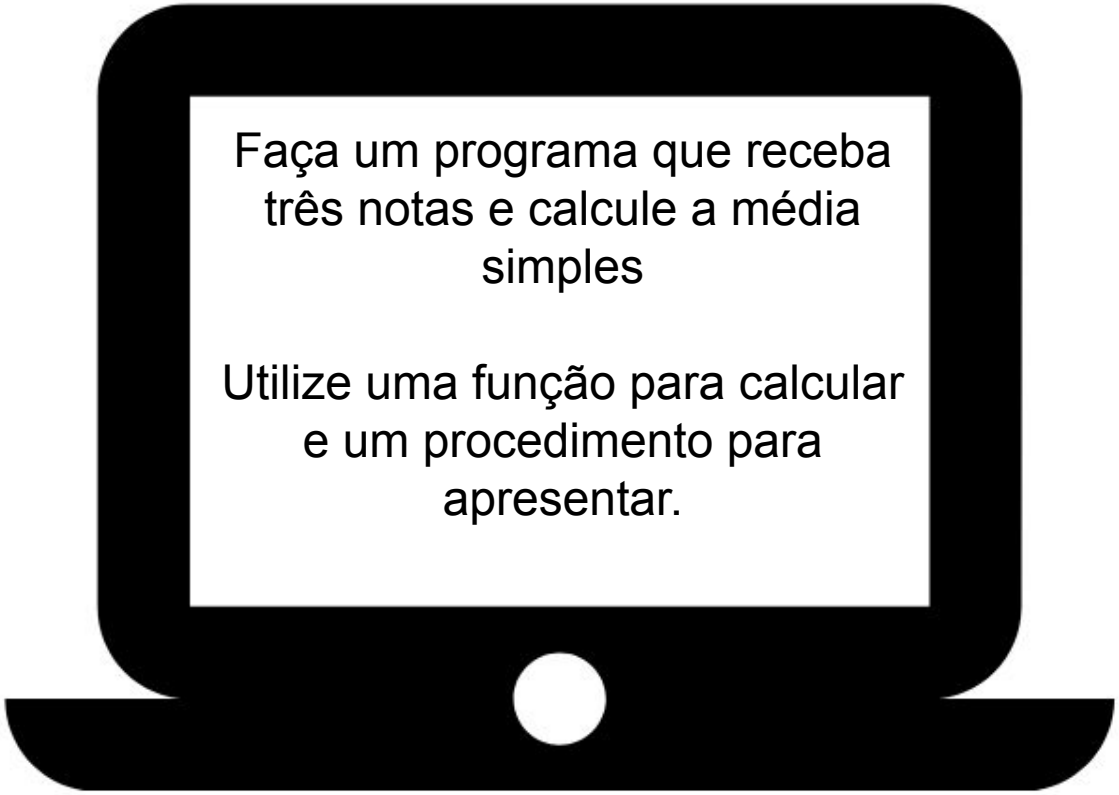
EXEMPLO 1: FUNÇÃO PARA SOMAR DOIS NÚMEROS

```
1   #include <iostream>
2   #include <string>
3
4   using namespace std;
5
6   int somar(int a, int b) {
7       return (a + b);
8   }
9
10  int main() {
11      int a, b, c;
12
13      cout << "Informe os valores a serem somados: " << endl;
14      cin >> a;
15      cin >> b;
16
17      c = somar(a, b);
18
19      cout << a << " + " << b << " = " << c << endl;
20  }
21
```

EXEMPLO 2: FUNÇÃO PARA EXECUTAR UM SOMATÓRIO

```
1   #include <iostream>
2   #include <string>
3
4   using namespace std;
5
6   int somatorio(int n) {
7       int resultado = 0;
8       for(int i = 1; i <= n; i++) {
9           resultado += i;
10      }
11
12      return resultado;
13  }
14
15  int main() {
16      int a, b;
17
18      cout << "Informe o limite do somatório: " << endl;
19      cin >> a;
20
21      b = somatorio(a);
22
23      cout << "Somatório de " << a << " = " << b << endl;
24  }
25
```

PRÁTICA 1: MÉDIA SIMPLES



PRÁTICA 2: MÉDIA PONDERADA

Faça um programa que receba N notas, seus respectivos pesos e, por fim, apresenta a média ponderada.

Utilize uma função para calcular e um procedimento para apresentar.

VARIÁVEIS LOCAIS E GLOBAIS

- Existe uma definição chamada “escopo” em programação
  - Vocês já encontraram erros com a palavra “scope”
  - É disso que estamos falando

- Pela definição, escopo é um “espaço” onde um determinado objeto “existe”
  - Em programação, os objetos são funções, procedimentos, variáveis, etc.

- Existem dois escopos básicos
  - Local: Só existe em determinado pedaço do código
  - Global: Existe em todo o código

VARIÁVEIS LOCAIS E GLOBAIS

```
for (int i = 0; i < 100; i++) {
    // Qualquer código
}
```

Nesse caso, i é uma variável local e só existe dentro do bloco do FOR

VARIÁVEIS LOCAIS E GLOBAIS

```
int main() {
    int a, b;

    cout << "Informe o limite do somatório: " << endl;
    cin >> a;

    b = somatorio(a);

    cout << "Somatório de " << a << " = " << b << endl;
}
```

As variáveis a e b são locais, só existem dentro da função main.

VARIÁVEIS LOCAIS E GLOBAIS

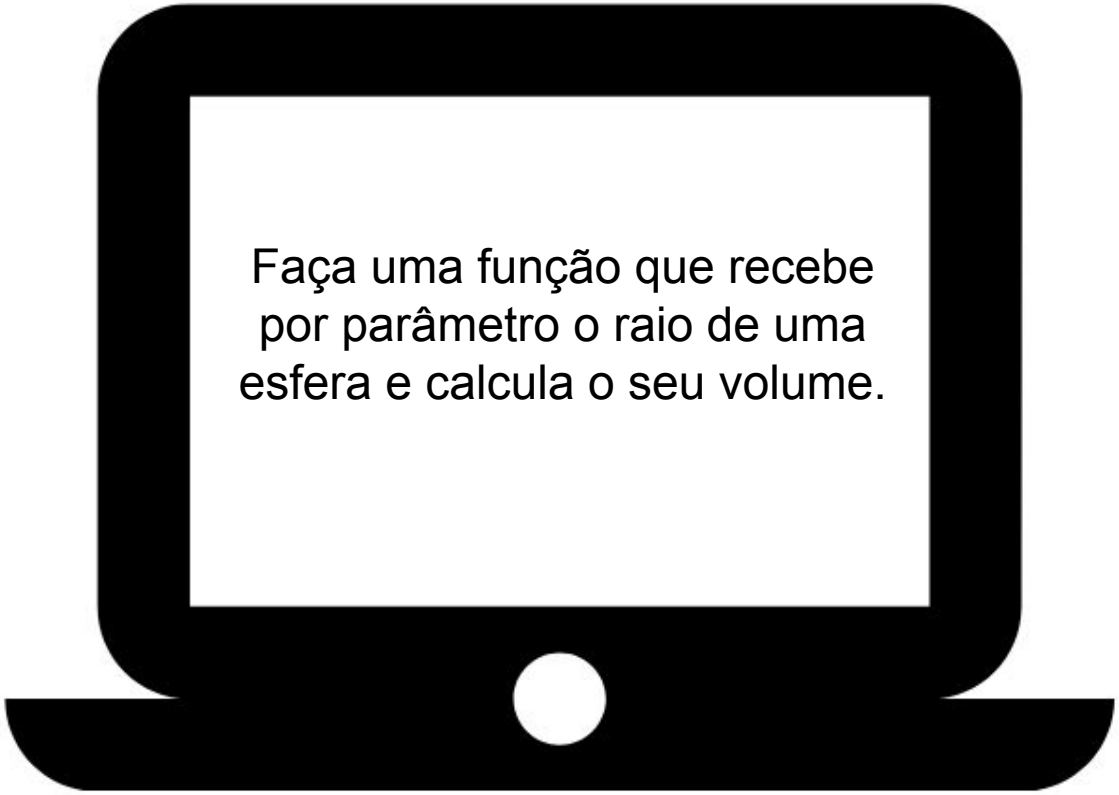
```
1   #include <iostream>
2   #include <string>
3
4   using namespace std;
5
6   string mensagem;
7
8   void apresentaMensagem() {
9       cout << mensagem << endl;
10  }
11
12  int main() {
13      cout << "Informe a mensagem: " << endl;
14      getline(cin, mensagem);
15
16      apresentaMensagem();
17  }
```

A variável mensagem é global.
Ela é manipulada na função main e na função apresentaMensagem sem nenhum tipo de passagem de parâmetro

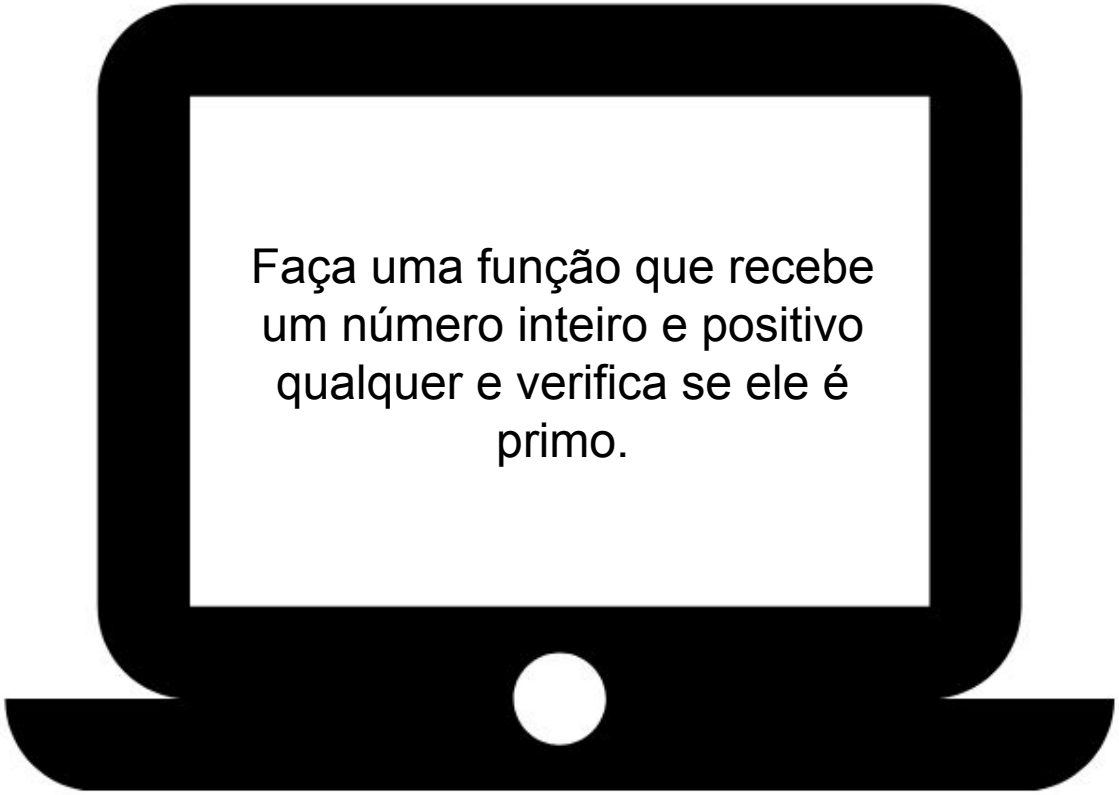
PRÁTICA 3: CONTAGEM DE MEMBROS

Faça um programa que receba N números inteiros positivos. Pare a inserção ao digitar 0. Após isso, pergunte um número inteiro positivo qualquer. Apresente quantas vezes esse aparece na inserção anterior.

PRÁTICA 4: VOLUME DA ESFERA



PRÁTICA 5: PRIMO



PRÁTICA 6: TRIÂNGULO

Faça uma função que três valores qualquer representando os lados de um triângulo e informa se ele é equilátero, isósceles ou escaleno.