

DIVIDIR PARA CONQUISTAR

Para resolver um problema o método mais fácil é dividi-lo em problemas menores:

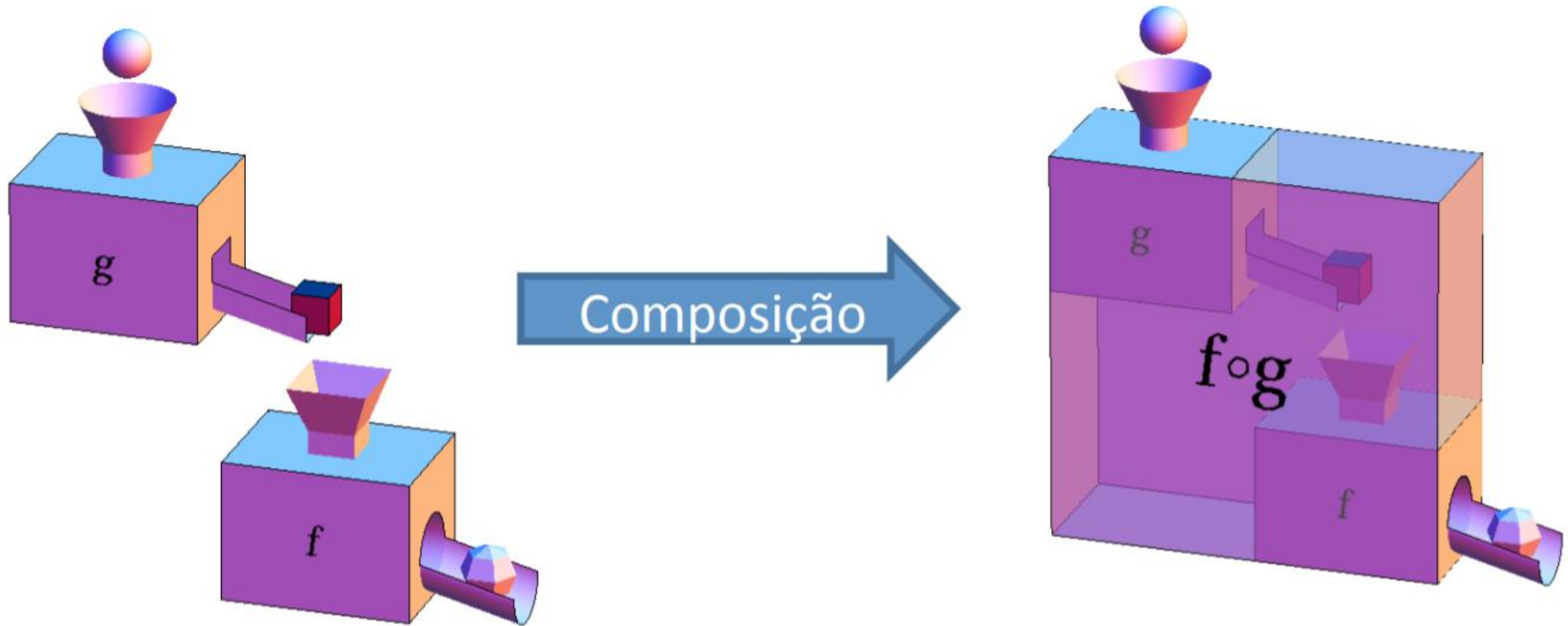
1. Dividir o problema em subproblemas;
2. Solução de todos os subproblemas;
3. Composição de todas as soluções dos subproblemas para resolver o problema principal.

DIVIDIR PARA CONQUISTAR

Quando estamos desenvolvendo um código para resolver algo, o processo mais simples e eficaz é quebrar o código-fonte em trechos menores (micro-módulos).

O nome desta metodologia é programação modular.

PROGRAMAÇÃO MODULAR



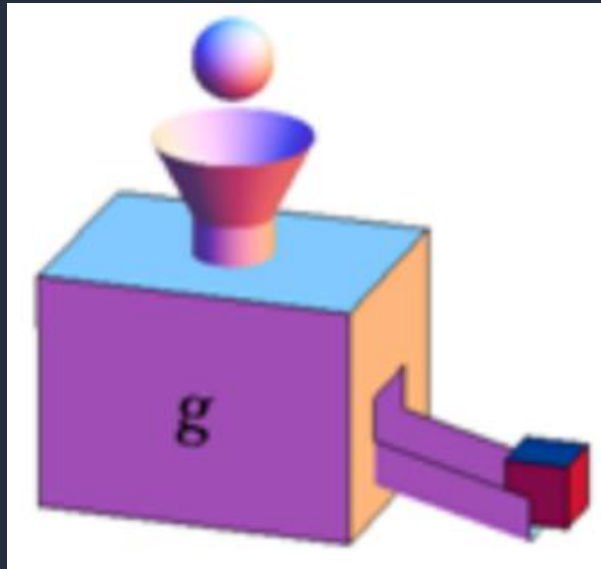
PROGRAMAÇÃO MODULAR

As principais vantagens dos módulos são:

- Podem ser desenvolvidos uma vez e reutilizado sempre que for necessário.
- Compostos para solucionar problemas complexos.
- Facilidade de manutenção:
 - É mais fácil identificar um erro no código-fonte de um módulo do que todo o código.
 - Facilidade em compreender todo o código-fonte.

FUNÇÕES

- Um módulo pode ser desenvolvido através de uma função (C) ou grupo com relações lógicas.
- É um conjunto de instruções/código-fonte utilizados para realizar um determinada tarefa



CRIAR FUNÇÕES

Parâmetros de Entrada: são os dados necessários para a implementação da tarefa.

- Vazio se não precisar de entrada de dados.

Tipos de retorno: são os dados obtidos da implementação deste módulo.

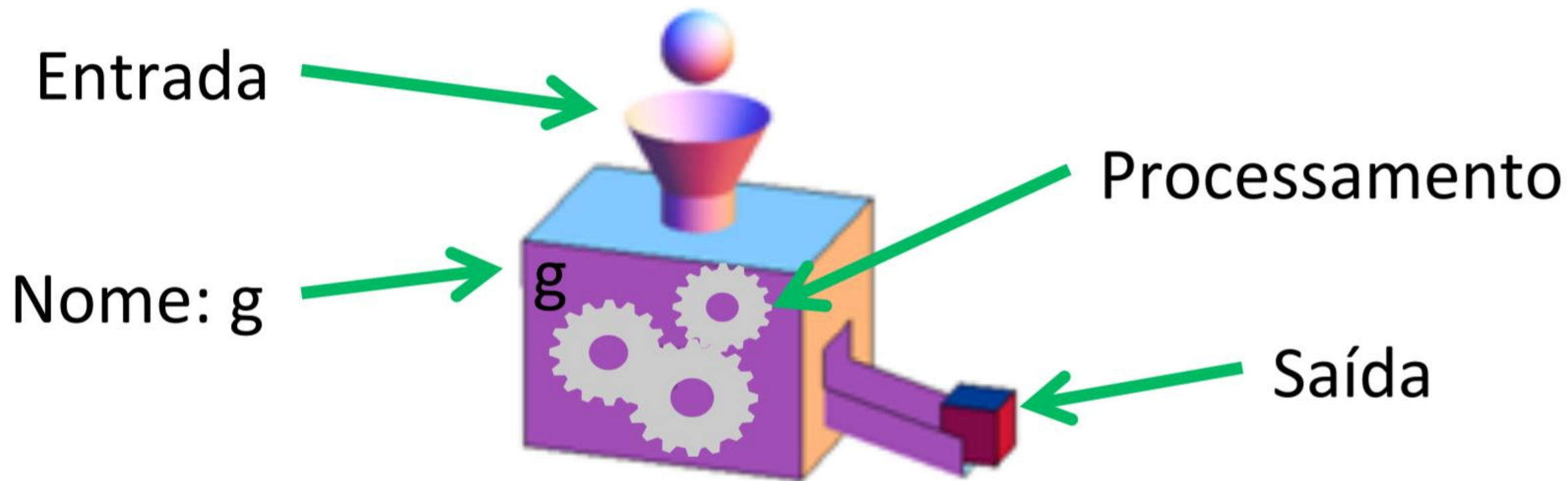
- Sem o return se não retornar nenhum resultado.

Processamento: é o interior da implementação na função/método, onde a tarefa é realizada.

Nome: indentificador da função, usa-se a mesma regra para criar uma variável.

Modificador: São palavras reservadas que definem o modo de acesso.

CRIAR FUNÇÕES



EXEMPLO FUNÇÃO EM LINGUAGEM C

```
float segundoGrau(float x, float a, float b, float c)
{
    float y = 0;

    y = a*x*x + b*x + c;

    return y;
}
```

EXEMPLO FUNÇÃO EM LINGUAGEM C

Tipo de dados da
saída (retorno): float

Entradas e seus tipos:

1º parâmetro: float x

2º parâmetro: float a

3º parâmetro: float b

4º parâmetro: float c

```
float segundoGrau(float x, float a, float b, float c) {  
    float y;  
  
    y = a*x*x + b*x + c; //y = ax^2 + bx + c  
  
    return y;  
}
```

Processamento
“corpo da função”

Nome da função: “segundoGrau”

Retornar para a saída o
resultado do processamento

EXERCÍCIO PRÁTICO 1

MENU OPÇÕES

Criar uma função em C para ler um número do utilizador e verificar as opções digitadas:

1 – Sair do programa

2 – Imprimir a mensagem "Chamou 1"

Se o utilizador não digitar as opções, mostra a mensagem "opção inválida" e executa novamente a função.

Criar uma função para imprimir a mensagem "Chamou 1".

EXERCÍCIO PRÁTICO 2

CALCULADORA

Criar um programa com as funções para somar, subtrair, multiplicar e dividir, ler um número do utilizador e verificar as opções para operação, ler 2 números e por fim imprimir o resultado.

EXERCÍCIOS



EXERCÍCIOS

1. Implemente um programa com uma função/método capaz de inverter um número de três dígitos inseridos pelo utilizador, por exemplo: entrada 765, saída 567).
2. Crie um programa com uma função que retorna o maior de dois números inseridos pelo utilizador, em seguida crie outra função que imprima o maior de 3 números reutilizando o função criada.
3. Implemente um programa utilizando funções para calcular o peso ideal da pessoa, o utilizador irá inserir o peso e a altura.
4. Crie um programa que utiliza uma função que recebe um vetor de inteiros e o total de elementos, retornando o maior elemento do vetor

