

Fundamentos de Programação

CP41F

Criação de bibliotecas, arquivos de cabeçalhos e variáveis globais.
Programação modular.

Aula 17

Prof. Daniel Cavalcanti Jeronimo

Universidade Tecnológica Federal do Paraná (UTFPR)
Engenharia de Computação – 1º Período
2016.1

- Arquivos de cabeçalhos.
- Criação de bibliotecas.
- Programação modular.
- Variáveis globais.

Arquivos de cabeçalhos

- Arquivos de cabeçalho (*header*)
 - Arquivos contendo **declarações** (nomes/identificadores e tamanhos)



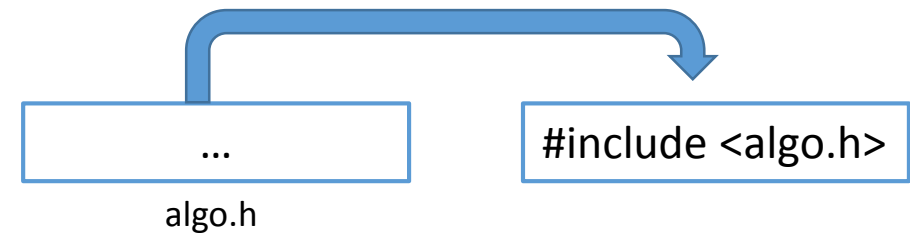
- Promovem encapsulação
- Auxiliam a reutilização de código

Arquivos de cabeçalhos

- Arquivos de cabeçalho (*header*)
 - A extensão **.h** é uma convenção, ajuda a identificar arquivos de cabeçalho
 - O arquivo de cabeçalho pode ter qualquer extensão e/ou nome, desde que não haja conflitos

Arquivos de cabeçalhos

- Arquivos de cabeçalho (*header*)
 - São incluídos em arquivos de código pela diretiva de pré-processamento **include**



- Exemplo: **#include <stdio.h>**
- A diretiva substitui sua linha pelo conteúdo do arquivo

Criação de bibliotecas

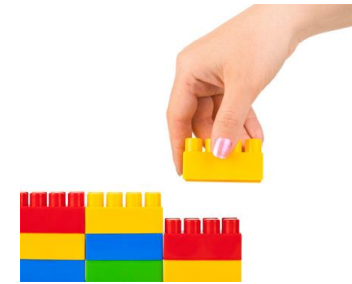
- Bibliotecas
 - Agrupam **funcionalidades** relacionadas a alguma tarefa
 - Podem ser estáticas ou dinâmicas
 - Exemplos:
 - **stdio.h** – biblioteca de entrada e saída
 - **string.h** – biblioteca de manipulação de strings

Criação de bibliotecas

- Bibliotecas
 - Podem ser definidas em um único arquivo de cabeçalho
 - Mais comum, a combinação de:
 - arquivo de cabeçalho (declarações)
 - arquivo de código (definições)
 - Bibliotecas muito grandes contém múltiplos arquivos de cabeçalhos e códigos

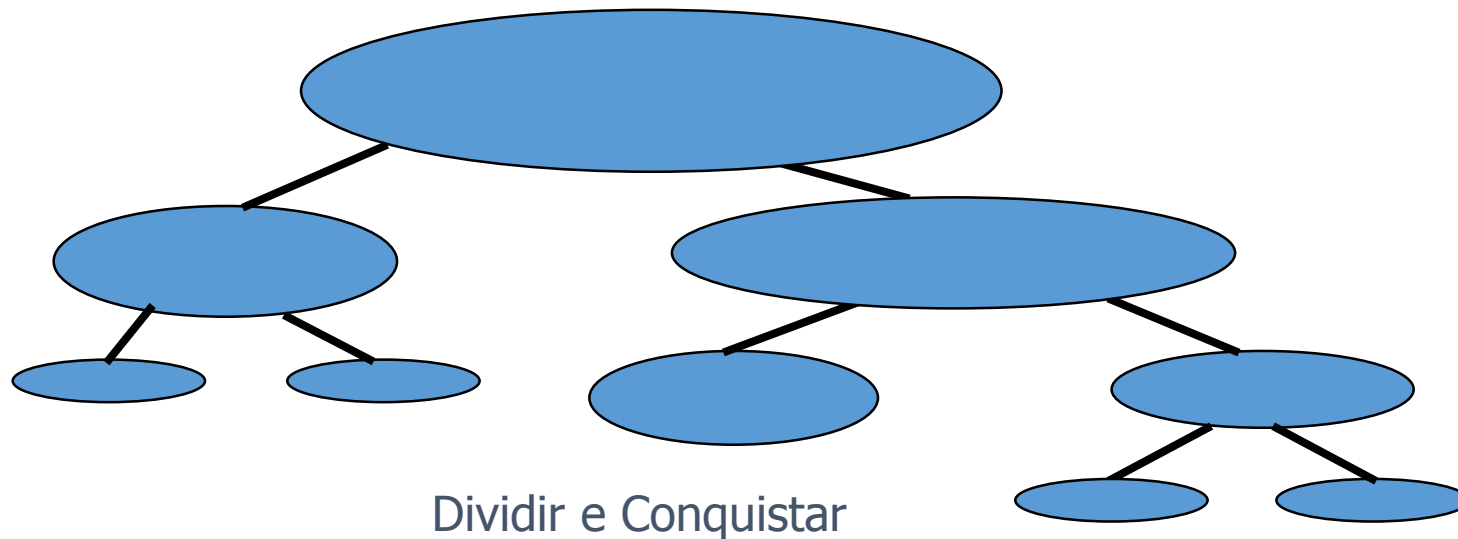
Programação modular

- Módulo
 - Construção abstrata – não tem definição precisa
 - Possui uma interface bem definida (entradas e saídas) e um propósito (executa uma tarefa)
 - Independente – contém em si mesmo toda a sua funcionalidade (auto-contido)
 - Intercambiável – pode ser trocado e reutilizado



Programação modular

- Divisão do código em módulos
- Cada módulo resolve um problema específico



Programação modular

- Vantagens

- Módulos podem ser escritos e **testados** separadamente

unit tests!



- Módulos podem ser reutilizados, reduzindo o tamanho do programa
- Grandes projetos podem ser desenvolvidos paralelamente

Programação modular

- Vantagens
 - Abstração:
 - O módulo esconde detalhes de solução da tarefa
 - Não é necessário saber como o módulo faz algo, apenas que ele faz algo
 - **Interface** e propósito bem definidos

Programação modular

- Princípio fundamental – Separação de Conceitos
 - Do inglês *Separation of Concerns (SoC)*
 - Cada seção do programa atende um aspecto de funcionalidade
 - Encapsulação de informação

Programação modular

- Revisando as ferramentas disponíveis:
 - Funções
 - Resolvem tarefas específicas
 - Bibliotecas
 - Agrupam funções relacionadas a alguma tarefa

Variáveis globais

- Tempo de vida – persistem ao longo do tempo de vida do programa
- Visibilidade – podem ser *extern*, visíveis a todas as unidades de compilação, ou *static*, visíveis apenas a uma unidade
- São declaradas e definidas no escopo global

Prática

- Como criar uma biblioteca para a APS?
- O que estou tentando resolver? (**propósito**)
- Quais as saídas e entradas esperadas? (**interface**)