

Tipos Abstratos de Dados

Linguagem de Programação I

Waldson Patrício



Introdução

- Até agora temos feito tudo de forma não modularizada:
 - O nosso programa encontra-se todo em um arquivo
 - Se quisermos reaproveitar nosso código, teremos que ter trabalho de modificar a implementação de alguma coisas
- Se a função `main` do programa de vocês for apagada e o resto do código for passado para outras pessoas, eles vão conseguir recriar o programa que vocês criaram?

Modularização

- Um programa pode ser dividido em vários arquivos fontes (.c, .cpp, .cxx)
- Cada módulo pode ser compilado separadamente, gerando um arquivo objeto
- Após a compilação de cada módulo, os arquivos objetos podem ser *linkados* para gerar um programa

Modularização

- Em geral um módulo agrupa **tipos** e **funções** relacionados
- Quando um módulo define um **tipo de dado novo** e o **conjunto de operações para manipular este tipo de dado**, dizemos que este módulo define um **tipo abstrato de dados** (TAD).

- Abstrato neste contexto significa que a implementação das operações não importa e deve ser **transparente** pra quem utiliza o tipo
- Então, um **TAD** é definido pela criação de um tipo e suas operações associadas e não pela forma que essas operações são implementadas.

- Se pensarmos no que fizemos nas aulas anteriores, qual(is) TADs poderíamos termos definido?

- Se pensarmos no que fizemos nas aulas anteriores, qual(is) TADs poderíamos termos definido?
 - Diário, Mensagem, Data

- Se pensarmos no que fizemos nas aulas anteriores, qual(is) TADs poderíamos termos definido?
- Quais operações deveríamos associar a cada um desses TADs?