Análise e Desenvolvimento de Algoritmos (2006/2007)

Rosália Rodrigues

Capítulo 0 : Introdução

Objectivos: Um curso intermédio de programação, com uma abordagem algorítmica

Conteúdo Programático:

0. Revisões / Conceitos Fundamentais
1. Verificação Formal
2. Recorrência
3. Análise de Complexidade
4. Estruturas de Dados
5. Tipos Abstractos de Dados

Página de rede da disciplina:

www.mat.ua.pt/rosalia/cadeiras/ADA/

Bibliografia:

J. Pavão Martins, *Introdução à Programação usando o Pascal*, McGraw-Hill, 1994.
(para "relembrar" os conceitos fundamentais)

Niklaus Wirth, *Algorithms + Data Structures = Programs*, Prentice-Hall, 1976.
(o texto "clássico" desta matéria)

Niklaus Wirth, *Algoritmos e Estruturas de Dados*, Prentice-Hall, Brasil, 1989.
(uma tradução)

Niklaus Wirth, *Algorithms & Data Structures*, Prentice-Hall, 1986
(versão do anterior, com os exemplos na linguagem Modula-2)

{uma das melhores séries de livros desta área}

P. Helman and R. Veroff, *Intermediate Problem Solving and Data Structures: WALLS AND MIRRORS*, Benjamin-Cummings, 1986
(o primeiro livro da série, em Pascal)

P. Helman and R. Veroff, *WALLS AND MIRRORS: Intermediate Problem Solving and Data Structures*, Benjamin-Cummings, 1988
(versão do anterior, em Modula-2)

P. Helman, R. Veroff and F.M. Carrano,
Intermediate Problem Solving and Data Structures: WALLS AND MIRRORS (2nd Edition),
Benjamin-Cummings, 1991
(versão actualizada, de novo em Pascal)

F.M. Carrano, P. Helman and R. Veroff,
Data Structures and Problem Solving with Turbo Pascal: WALLS AND MIRRORS,
Benjamin-Cummings, 1993
(versão em Turbo Pascal)

Janet J. Prichard and Frank M. Carrano,
Data Abstraction and Problem Solving with C++: WALLS AND MIRRORS (3rd Edition),
Addison-Wesley, 2001

Frank M. Carrano,
Data Abstraction and Problem Solving with C++: WALLS AND MIRRORS (4rd Edition),
Addison-Wesley, 2004
(versões em C++)

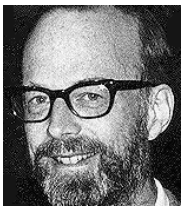
Frank M. Carrano and Janet J. Prichard,
Data Abstraction and Problem Solving with Java: WALLS AND MIRRORS (1st Edition),
Addison-Wesley, 2000

Frank M. Carrano and Janet J. Prichard,
Data Abstraction and Problem Solving with Java: WALLS AND MIRRORS (Updated Edition),
Addison-Wesley, 2004
(versões em Java)

Algumas Citações:

"It has often been said that a person does not really understand something until he teaches it to someone else. Actually a person does not really understand something until he can teach it to a computer, i.e., express it as an algorithm."

Donald E. Knuth



"I hold the opinion that the construction of computer programs is a mathematical activity like the solution of differential equations, that programs can be derived from their specifications through mathematical insight, calculation, and proof, using algebraic laws as simple and elegant as those of elementary arithmetic."

Sir Charles Antony Richard Hoare

"It is a mistake to consider the prime characteristic of high-level languages to be that they allow us to express programs merely in their shortest possible form and in terms of letters, words, and mathematical symbols instead of coded numbers.
Instead, the language is to provide a framework of abstractions and structures that are appropriately adapted to our mental habits, capabilities, and limitations."

Niklaus Wirth

"Programming is the most difficult part of applied mathematics."

"Computer Science is no more about computers than astronomy is about telescopes."

"Referring to what we do as "computer science" is like referring to surgery as "knife science"."

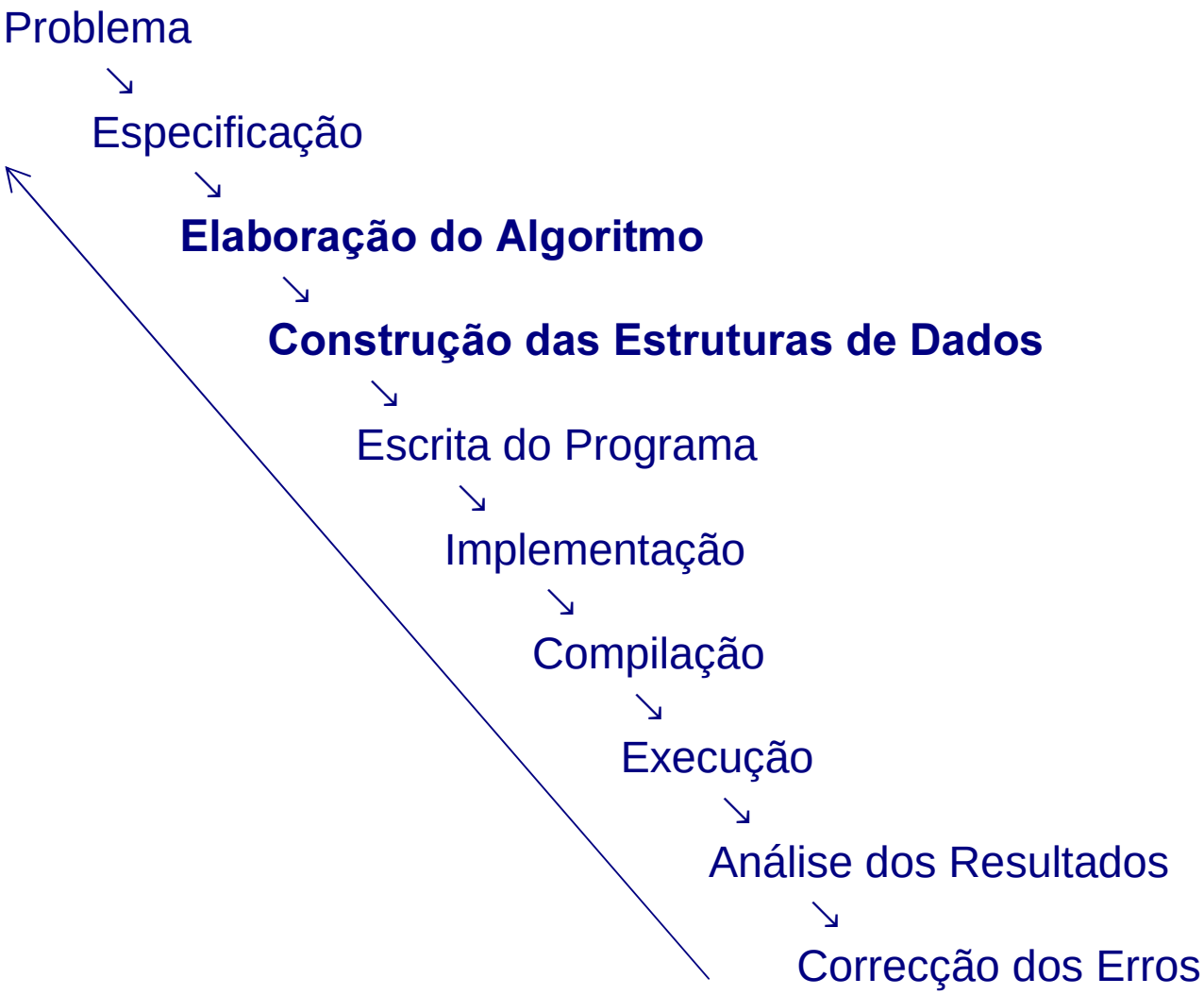
Edsger W. Dijkstra (1930-2002)

"The Analytical Engine weaves algebraic patterns, just as the Jacquard loom weaves flowers and leaves."

Lady **Ada** Augusta (1815-1852), Condessa de Lovelace, filha de Lord Byron, amiga de Charles Babbage, considerada a inventora da Programação de Computadores.

Programar ≡ Resolver Problemas

{ Programa ≡ Solução de um Problema }



Programa = Algoritmos + Estruturas de Dados

Instrumentos para a Resolução de Problemas:

- **Modularidade dos Algoritmos**
  (Programação Estruturada)
- **Modularidade das Estruturas de Dados**
  (Tipos Abstractos de Dados)
- **Recorrência**

O que é uma Boa Solução? O que é um Bom Programa?
(Metodologia da Programação)

- **Clareza:** O programa deve reflectir claramente a estrutura do algoritmo. Deve ser fácil de **ler**, corrigir, ampliar ou modificar, mesmo por outro programador.
- **Correcção:** O programa deve cumprir exactamente as especificações.
- **Eficiência:** O programa deve tentar minimizar, tanto o seu **tempo** de execução, como o **espaço** de memória utilizado.

Que Linguagem de Programação?