

# BCC 701

# Programação de Computadores I

Professor: Anderson Almeida Ferreira

[anderson.decom@gmail.com](mailto:anderson.decom@gmail.com)

<http://www.decom.ufop.br/anderson>

Sala: COM 10

DECOM - UFOP

# Programa da disciplina

- Ementa:
  - Conceitos básicos de organização de computadores, sistemas operacionais e ambientes de programação. Conceito de algoritmo. Conceitos básicos de programação: valores e expressões de tipos primitivos, variáveis, comando de atribuição, comandos de controle de fluxo, entrada e saída padrão, procedimentos e funções, tipos de dados compostos.

# Programa da disciplina

- Objetivos
  - Introduzir noções básicas de organização de computadores, representação de dados e programação
  - Apresentar ao aluno alguns princípios básicos da construção de algoritmos e de sua implementação em um ambiente de programação
  - Tornar o aluno fluente no uso de uma ferramenta computacional, o *Scilab*, de vasta aplicação nas ciências e engenharias

# Programa da disciplina

- Programa
  1. Organização de Computadores
    - 1.1. Arquitetura básica de um sistema de computação
    - 1.2. Sistema operacional
    - 1.3. Compilação e interpretação de programas
    - 1.4. Representação de dados
      - Sistema binário
      - Codificação de caracteres
  2. Conceitos e representação de algoritmos

# Programa da disciplina

3. Conceitos básicos de programação
  - 3.1. Valores, tipos e expressões
  - 3.2. Variáveis e comando de atribuição
  - 3.3. Comandos de entrada e saída
  - 3.4. Comandos de controle de fluxo
  - 3.5. Procedimentos e funções
  - 3.6. Recursividade
  - 3.7. Estruturas de dados homogêneas
  - 3.8. Estruturas de dados heterogêneas

# Programa da disciplina

- **BIBLIOGRAFIA**

- Farrer, H e outros. Algoritmos Estruturados - 3 Ed - Livros Técnicos e Científicos Editora S. A. - 1999.
- Ascencio, A. F . G.; Campos, E. A. V. Fundamentos da Programação de Computadores. Longman do Brasil, 2007.
- Guimarães, A. M. E.; Lages, N. A. C. - Algoritmos e Estruturas de Dados - Livros Técnicos e Científicos Editora S. A. -1985.
- Carvalho, O. Introdução à Organização e à Programação de Computadores, 2011.
- <http://www.scilab.org>

# Programa da disciplina

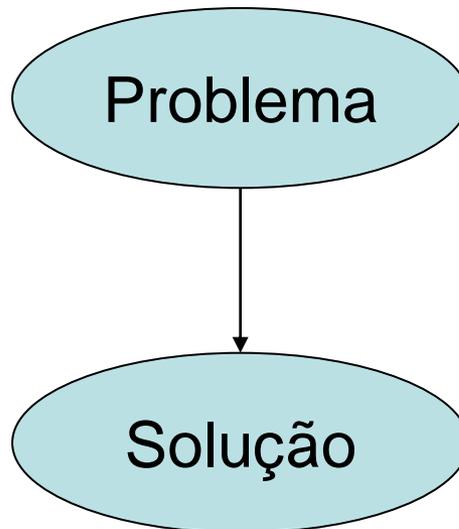
- Avaliação:
  - 2 provas (30% e 30%)
  - 3 trabalhos práticos (30%)
  - Listas (10%)

# Por que estudar programação?

- Todos nós programamos.
- Ao programar construímos algoritmos.
- Ao programar utilizamos lógica.

# Algoritmo

- Seqüência de passos finitos com o objetivo de solucionar problemas.
- Algoritmo não é a solução do problema.



# Algoritmo

- Todos nós, no dia-a-dia, nos deparamos com vários problemas...
  - Estamos a todo instante concebendo algoritmos...
  - Estamos a todo instante programando...
- Exemplo:
  - 1- Retirar o telefone do gancho
  - 2- Esperar o sinal
  - 3- Discar o número
  - 4- Falar ao telefone
  - 5- Colocar o telefone no gancho

# Exemplo

- Algoritmo para trocar lâmpada
  - Se (lâmpada estiver fora do alcance)  
pegar a escada;
  - Pegar lâmpada;
  - Se (lâmpada estiver quente)  
pegar pano;
  - Tirar lâmpada queimada;
  - Colocar lâmpada nova.

# Exemplo

- Algoritmo para fazer uma prova
  - Ler a prova;
  - Pegar caneta;
  - Enquanto ((houver questão em branco) e (houver tempo)) faça
    - Se (souber a questão)  
Resolva-a;
    - Senão  
Pule para a próxima;
  - Entregar a prova.

# Algoritmo

- Algoritmo para levar um leão, uma cabra e um pedaço de grama de um lado para outro do rio, usando um bote para tal tarefa. Sabe-se que nunca um leão pode ficar sozinho com uma cabra e nem a cabra sozinha com a grama.

# Algoritmo

- Algoritmo para levar um leão, uma cabra e um pedaço de grama de um lado para outro do rio, usando um bote para tal tarefa. Sabe-se que nunca um leão pode ficar sozinho com uma cabra e nem a cabra sozinha com a grama.
  - 1- Levar a grama e o leão;
  - 2- Voltar com o leão;
  - 3- Deixar o leão;
  - 4- Levar a cabra;
  - 5- Deixar a cabra;
  - 6- Voltar com a grama;
  - 7- Levar o leão e a grama;
- Forneça outro algoritmo.

# Algoritmo

- Forneça um algoritmo para o problema das *Torres de Hanoi*.
- A proposição do problema é a seguinte: inicialmente têm-se 3 hastes, A, B e C, e a haste A tem três anéis de diâmetros distintos, em ordem decrescente. O objetivo é transferir os três anéis da haste A para a B, usando C se necessário. As regras de movimentação são:
  - Deve-se mover um único anel por vez;
  - Um anel de diâmetro maior nunca pode repousar sobre algum outro de diâmetro menor.

# Modelagem de problemas

- É responsável pela facilidade ou dificuldade da resolução de problemas.
- Na matemática ou na engenharia, por exemplo, o uso da linguagem matemática é fundamental, principalmente pela eliminação de duplos sentidos.
- O mesmo ocorre na computação, com o emprego de linguagens de descrição de algoritmos e de linguagens de programação.

# Modelagem de problemas

- Exemplo:
  - Compraram-se 30 canetas iguais, que foram pagas com uma nota de R\$ 100,00, obtendo-se R\$ 67,00 como troco. Quanto custou cada caneta?

# Modelagem de problemas

- Uma possível resposta:
  - Se eu tinha R\$ 100,00 e recebi como troco R\$ 67,00, o custo do total das canetas é a diferença entre os R\$ 100,00 que eu tinha e os R\$ 67,00 do troco. Ora, isto vale R\$ 33,00; portanto, esse valor foi o total pago pelas canetas. Para saber quanto custa cada caneta, basta dividir R\$ 33,00 por 30, resultando no preço de cada caneta. Assim, cada caneta custou o equivalente a R\$ 1,10.

# Modelagem de problemas

- Matematicamente:
  - Seja  $x$  o custo de cada caneta, então  $\text{quantogastei} = 30x$ .
  - Como  $\text{quantogastei} + \text{troco} = R\$ 100,00$ , tem-se:
    - $30x + 67 = 100$
    - $30x = 100 - 67$
    - $30x = 33$
    - $x = 33/30$
    - $x = 1.1$

# Modelagem de problemas

- Algoritmo
  - Pegar os valores 30, 67 e 100.
  - Subtrair 67 de 100 e dividir o resultado por 30.
  - Mostrar o resultado final.

# Modelagem de problemas

- Algoritmo geral
  - Ler os valores de  $N$ ,  $Y$  e  $Z$ .
  - Subtrair  $Y$  de  $Z$  e dividir o resultado por  $N$ .
  - Mostrar o resultado.

# Modelagem de problemas

- Algoritmo detalhado
  - Ler os valores de  $N$ ,  $Y$  e  $Z$ .
  - **Se**  $Z > Y$  e  $N > 0$  e  $Y \geq 0$  e  $Z > 0$  **Então**
    - Subtrair  $Y$  de  $Z$  e dividir o resultado por  $N$ .
    - Mostrar o resultado final.
  - **Senão**
    - Exibir a mensagem: “Erro: os valores são inconsistentes”.
  - **Fim se**

# Como se portar em um curso de programação

- O grande problema apresentado pelos estudantes não está relacionado as linguagens de programação ou a descrição de algoritmos, mas sim a dificuldade de abstrair e descrever as soluções de problemas contando apenas com poucas e simples estruturas.
- **É um erro decorar as soluções em computação.**
- Deve-se procurar o **entendimento** de como foi obtida um solução, armazená-la na memória e então utilizá-la adaptando-a a outras situações.

# Como se portar em um curso de programação

- Não existe uma “fórmula mágica” para resolver problemas. Algumas dicas:
  - Ao deparar-se com um problema novo, tente entendê-lo.
    - O que se deve descobrir ou calcular?
    - Quais são os dados disponíveis?
    - Quais as condições necessárias e suficientes para resolver o problema?
    - Faça um esboço informal de como ligar os dados e com as condições.
    - Se possível, modelar o problema de forma matemática.

## – Crie um plano com a solução

- Consulte a memória e verifique se você já resolveu algum problema similar.
- Verifique se é necessário introduzir algum elemento novo no problema.
- Se o problema for complicado, tente quebrá-lo em partes menores e solucionar essas partes.
- É possível enxergar o problema de outra forma, de modo que seu entendimento se torne mais simples?

## – Formalize a solução

- Crie um algoritmo informal com passos que resolvam o problema.
- Verifique se cada passo desse algoritmo está correto.
- Escreva um algoritmo detalhado

## – Examine os resultados

- Teste o algoritmo com diversos dados de entrada e verifique o resultado.
- Se o algoritmo não gerou resultado algum, o problema está na sua sintaxe e nos comandos utilizados. Volte e tente encontrar o erro.
- Se o algoritmo gerou resultados, estes estão corretos? Analise sua consistência.
- Se não estão corretos, alguma condição, operação ou ordem das operações está incorreta. Volte e tente encontrar o erro.

## – Otimização da solução

- É possível melhorar o algoritmo?
- É possível diminuir o número de passos ou dados?
- É possível conseguir uma solução ótima?

# Linguagem e o Ambiente Scilab

- Desenvolvido desde 1990 por pesquisadores do INRIA e da École Nationale des Ponts et Chaussées (França)
- Muito semelhante ao MatLab – e gratuito!
  - <http://www.scilab.org>
- É um interpretador
- A linguagem e o sistema têm o mesmo nome, Scilab

# Scilab

- Uma linguagem de programação, como as linguagens naturais, une riqueza de expressão a detalhes sintáticos
- Seu aprendizado exige uma postura paciente, pois envolve no início uma taxa inicial de memorização
- Também como nas linguagens naturais, a fluência vem com o uso