

BCC701 – Programação de Computadores I
Universidade Federal de Ouro Preto
Departamento de Ciência da Computação

www.decom.ufop.br/bcc701
2012/01



Semana 02:

Introdução ao Scilab.

Comandos de entrada e saída de dados.

Material Didático Unificado.

Agenda

- Introdução;
- Comandos de entrada e saída de dados;
- Introdução ao uso do Fluxograma;
- Utilizando o ambiente SciNotes;
- Exercícios.

Introdução;

Comandos de entrada e saída de dados;

Introdução ao uso do Fluxograma;

Utilizando o ambiente SciNotes;

Exercícios.

INTRODUÇÃO

A linguagem Fortran

- Em 1954, a linguagem de alto nível Fortran foi proposta por um grupo da IBM.
- O primeiro **compilador** (ou seja, um programa que traduz programas escritos em linguagem de alto nível para instruções de máquina) foi naturalmente escrito em Assembler.
- A máquina era um IBM 704: um computador com 15K de memória.

Linguagens de programação

- Existem várias linguagens de programação que descendem do Fortran; por exemplo:
 - 1959 – Cobol;
 - 1964 – Basic;
 - 1970 – Pascal;
 - 1971 – C;
 - 1983 – C++;
 - 1991 – Python;
 - 1995 – Java;
 - 1995 – PHP.

Matlab

- Foi criado no fim dos anos 70 por *Cleve Moler* e lançado comercialmente em 1984 pela empresa *MathWorks*.
- É voltado para engenheiros e cientistas.
- Possui grande facilidade para o tratamento de matrizes (MatLab = *Matrix Laboratory*).
- É um ***interpretador***, ou seja, um programa que executa programas; ao contrário de um compilador, não traduz um programa para instruções de máquina.

Scilab

- Foi criado em 1990 por pesquisadores do INRIA e da École Nationale des Ponts et Chaussées (França), sendo gratuito e bastante semelhante ao MatLab.
 - <http://www.scilab.org>
- Consiste também em um interpretador.
- A linguagem e o sistema possuem o mesmo nome: Scilab.
- Será apresentada a versão 5.3.3 do Scilab.

A linguagem Scilab

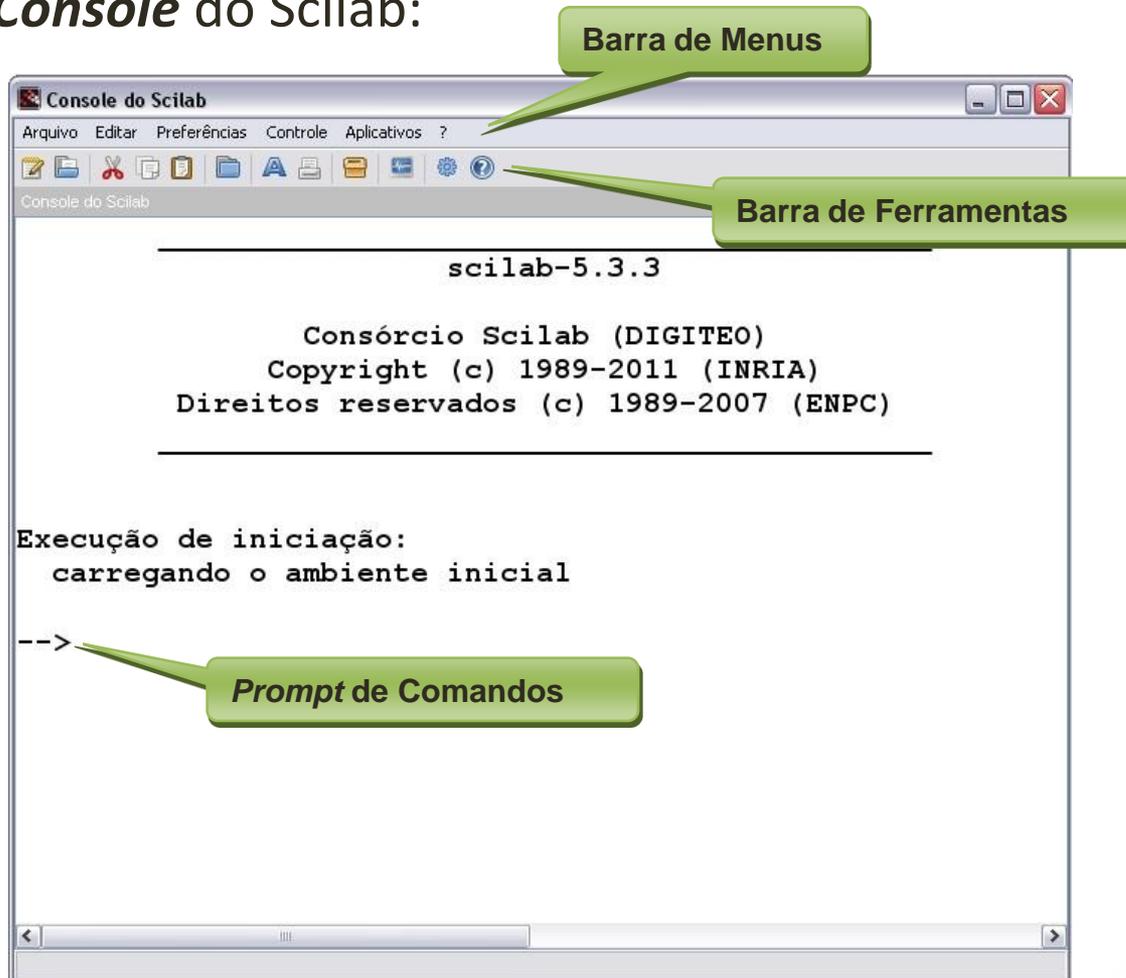
- Como qualquer linguagem natural, a linguagem Scilab:
 - Une riqueza de expressão a detalhes sintáticos;
 - Exige uma postura paciente em seu aprendizado, pois envolve uma taxa inicial de memorização;
 - A fluência vem com a prática.

O ambiente Scilab

- Interpreta comandos e programas através de uma console para a interação com o usuário;
- Oferece um editor para a construção de programas (SciNotes);
- Emite mensagens de erros relativos à obediência da sintaxe da linguagem e a problemas na execução de um programa (como divisão por zero).

O ambiente Scilab

- Janela **Console** do Scilab:



Introdução;

Comandos de entrada e saída de dados;

Introdução ao uso do Fluxograma;

Utilizando o ambiente SciNotes;

Exercícios.

COMANDOS DE ENTRADA E SAÍDA DE DADOS

Variáveis

- Variáveis correspondem a nomes para espaços de memória que são gerenciados pelo Scilab;
- O programador não precisa ter qualquer ideia de como tal gerência é realizada;

Variáveis

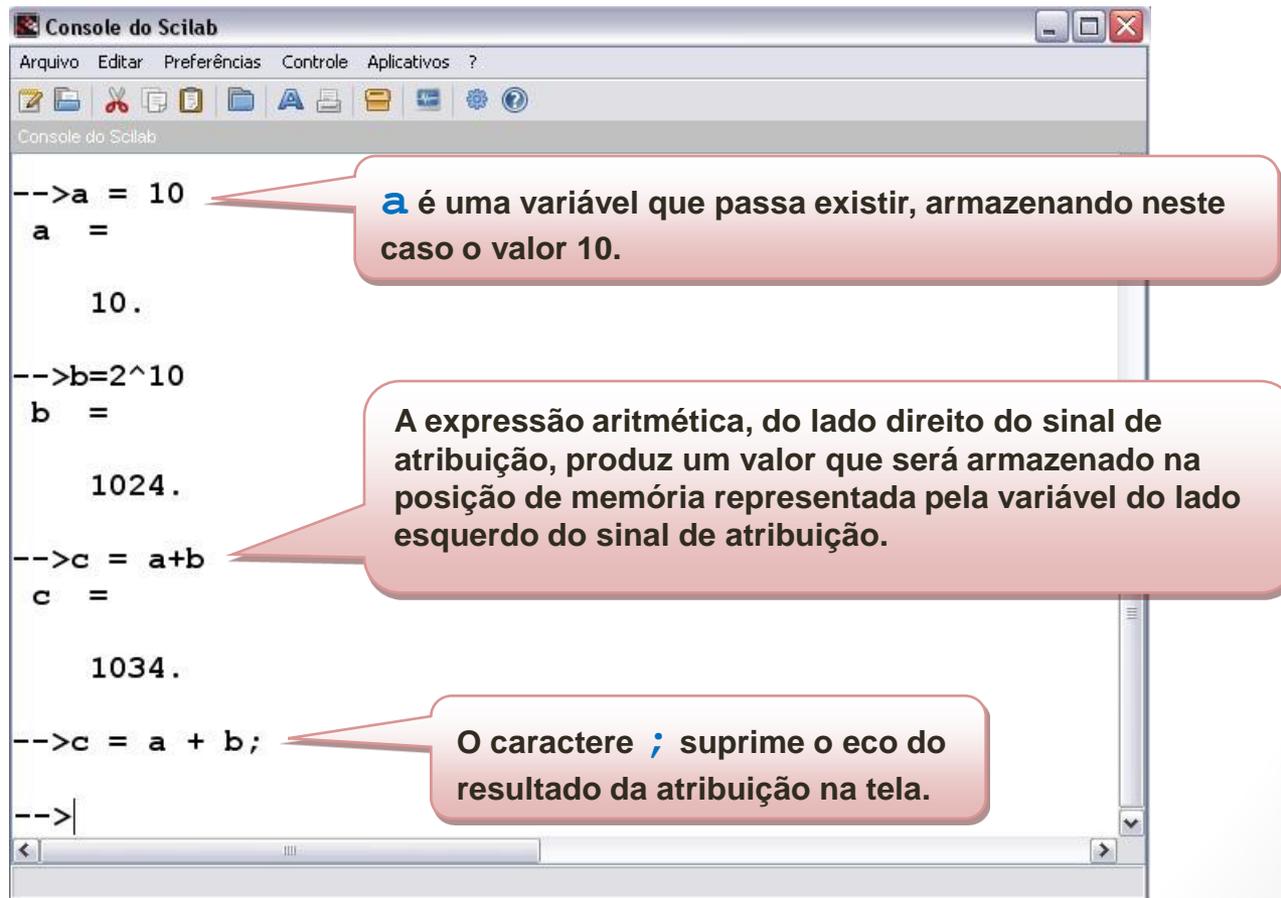
- Os nomes das variáveis são escolhidos pelo programador, respeitando as seguintes regras:
 1. O **primeiro caractere** do nome deve ser uma letra ou qualquer caractere dentre '%', '_', '#', '!', '\$' e '?';
 2. Os **outros caracteres** podem ser letras ou dígitos ou qualquer caractere dentre '_', '#', '!', '\$' e '?';
 3. Caracteres acentuados não são permitidos;
 4. Nomes de variáveis são sensíveis a maiúsculas e minúsculas. Por exemplo, variável Alpha é diferente das variáveis ALPHA, alpha e ALPhA.

Variáveis

- A escolha de nomes significativos para as variáveis ajuda ao programador entender o que o programa faz e a prevenir erros;
- **Nomes válidos:**
 - a , A , Jose , total_de_alunos , #funcionarios.
- **Nomes inválidos:**
 - 1Aluno (o primeiro caractere é um algarismo)
 - total de alunos (tem espaços)
 - José (é acentuado)

Variáveis

- Definindo variáveis:



The screenshot shows the Scilab Console window with the following text and callouts:

```
-->a = 10
a =
    10.

-->b=2^10
b =
    1024.

-->c = a+b
c =
    1034.

-->c = a + b;

-->|
```

a é uma variável que passa existir, armazenando neste caso o valor 10.

A expressão aritmética, do lado direito do sinal de atribuição, produz um valor que será armazenado na posição de memória representada pela variável do lado esquerdo do sinal de atribuição.

O caractere **;** suprime o eco do resultado da atribuição na tela.

Comando de atribuição

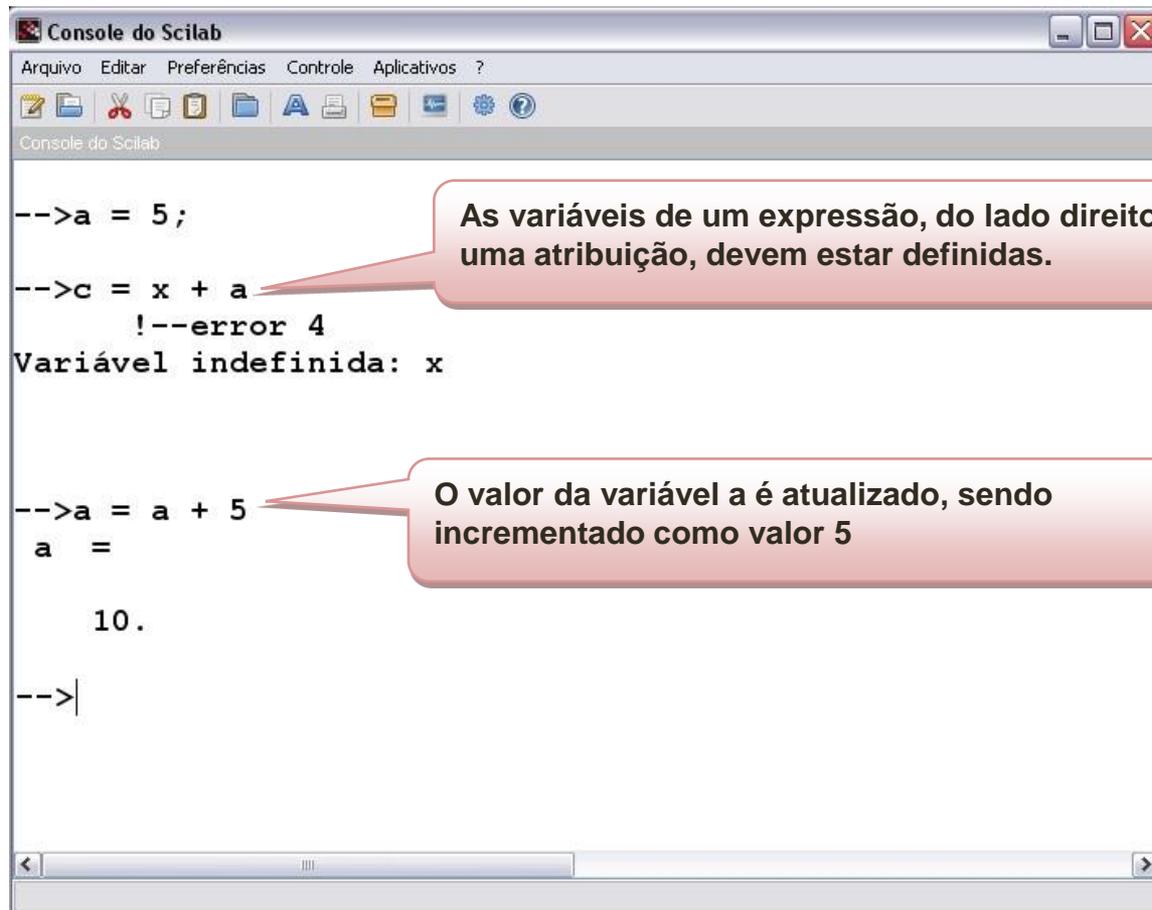
- Sintaxe:

<variável alvo> = <expressão>

- A *<variável alvo>*, se não existia, passa a existir;
 - Se existia, o valor anterior é perdido;
- Na execução do comando, a *<expressão>* é calculada e o resultado é atribuído à *<variável alvo>*.

Comando de atribuição

- Exemplos de atribuição:



```
Console do Scilab
Arquivo  Editar  Preferências  Controle  Aplicativos  ?
[Icons]
Console do Scilab

-->a = 5;

-->c = x + a
      !--error 4
Variável indefinida: x

-->a = a + 5
a =
    10.

-->
```

As variáveis de um expressão, do lado direito de uma atribuição, devem estar definidas.

O valor da variável a é atualizado, sendo incrementado como valor 5

Operadores aritméticos

- A linguagem SciLab possui os **operadores aritméticos**:

| Operador Aritmético | Denotação em SciLab | Exemplo | Resultado |
|--------------------------------|---------------------|-----------|-----------|
| Soma | + | $7 + 5$ | 12 |
| Subtração | - | $10 - 9$ | 1 |
| Multiplicação | * | $22 * 10$ | 220 |
| Divisão | / | $50 / 2$ | 25 |
| Menos Unário | - | -26 | -26 |
| Exponenciação (potenciação) | ^ | 8^2 | 64 |

Funções elementares

- São exemplos de **funções implementadas** no SciLab:

| Função | Denotação em SciLab | Exemplo | Resultado |
|--------------------------|---------------------|--------------|-----------|
| Resto da Divisão Inteira | modulo | modulo(8, 3) | 2 |
| Raiz Quadrada | sqrt | sqrt(32) | 5.6568542 |
| Valor Absoluto | abs | abs(-8) | 8 |
| Coseno | cos | cos(30) | 0.1542514 |
| Tangente | tan | tan(7.3456) | 1.7945721 |
| Seno | sin | sin(%pi) | 1.225D-16 |

Notação Scilab (e Fortran, e C, e Java, e ...) para:
2.418 x 1024

- OBS: Nas funções trigonométricas os ângulos devem ser usados em radianos.

Valores pré-definidos

- O SciLab possui alguns **valores pré-definidos**, alguns exemplos:

| Denotação em Scilab | Valor |
|---------------------|---|
| %pi | O número π . |
| %inf | Representa infinito ∞ . |
| %i | $\sqrt{-1}$ |
| %e | A base do logaritmo natural. |
| %t ou %T | Representa o valor booleano verdadeiro. |
| %f ou %F | Representa o valor booleano falso. |

- Como o Scilab é sensível a maiúsculas e minúsculas, não será possível usar %PI, %Pi, %Inf, ou qualquer variação desta natureza, ao menos que seja definido na linguagem, como para os valores verdadeiro e falso.

Precedência de operadores

- A precedência de operadores indica qual operador deverá ser executado primeiro.
- Assim, na expressão aritmética $2 + 3 * 6$, a subexpressão $3 * 6$ é executada primeiro;
 - Portanto, tem-se como resultado para a expressão o valor 20.

Precedência de operadores

- O caso da expressão $2^3 * 4$, o valor resultante será:
 - $2^{3*4} = 2^{12} = 4096$,
 - ou o valor será $2^3 * 4 = 8 * 4 = 32$?
- Para respondermos esta pergunta, além do conhecimento da prioridade dos operadores envolvidos, devemos saber também qual são as suas associatividades.

Precedência de operadores

- A tabela abaixo define a precedência e a associatividade para alguns operadores:

| Prioridade | Operação | Associatividade |
|----------------|------------|-----------------------------|
| 1 ^a | \wedge | Da direita para a esquerda. |
| 2 ^a | $*$ $/$ | Da esquerda para a direita. |
| 3 ^a | $+$ $-$ | Da esquerda para a direita. |

- Exemplos:
 - $2+10/5$ → $10/5$ é avaliada primeiro;
 - $A+B/C+D$ → B/C é avaliada primeiro;
 - $R*3+B^3/2+1$ → B^3 é avaliada primeiro.

Precedência de operadores

- **Associatividade** é a regra usada quando os operadores têm a mesma prioridade;
- Por exemplo, para as operações de **adição** e **subtração** (que possuem mesma prioridade) a regra de associatividade diz que a operação mais a esquerda é avaliada primeiro:
 - $A-B+C+D \rightarrow A-B$ é avaliada primeiro, pois está mais à esquerda;
- O mesmo vale para **multiplicação** e **divisão**;
- Mas, para **potenciação**, a regra da associatividade diz que a operação mais a direita deve ser avaliada primeiro:
 - $A^B^C^D \rightarrow C^D$ é avaliada primeiro, pois está mais à direita.

Precedência de operadores

- A ordem de prioridade pode ser alterada pelo uso do parênteses:
 - $(A+4)/3$ → $A+4$ é avaliada primeiro;
 - $(A-B)/(C+D)$ → $A-B$ é avaliada primeiro, depois a soma e por último a divisão;
 - $R*3+B^(3/2)+1$ → $3/2$ é avaliada primeiro.

Entrada de dados

- O comando de atribuição é uma forma que o programador possui para armazenar valores numéricos, dentre outros, na memória do computador;
- Outra possibilidade que dispõe o programador, é a utilização do comando de leitura de dados pelo teclado, **input**;
- Este comando permite o armazenamento de valores diferentes para uma mesma variável, a cada execução do programa;
- A seguir, a sintaxe geral do comando **input**.

Entrada de dados

- Sintaxe geral do comando **input**:

<variável alvo> = input(<frase>)

- Onde:
 - **<variável alvo>** é uma variável que representará uma posição da memória que armazenará o valor digitado;
 - **<frase>** é uma *string* que informa ao usuário qual o dado que ele deve digitar nesta interação. A *string* deve estar entre aspas duplas.
- Suponha que o usuário deseje armazenar o valor **50**, referente à quantidade de alunos em uma sala de aula, na variável **Qtd_Alunos**. Isso pode ser realizado pela instrução:
 - **Qtd_Alunos = input("DIGITE A QUANTIDADE DE ALUNOS").**

Saída de dados

- Após um dado ser armazenado em uma variável, seja por atribuição ou por leitura, o mesmo pode ser exibido na tela do computador através do comando **printf**, o qual tem a seguinte sintaxe geral:

printf(<frase>, <lista de expressões>)

- Onde:
 - **<frase>** é a sentença que se quer imprimir na tela, e que pode estar entremeada por códigos de formato como **%g**;
 - **%g** é um código de formato geral para expressões com valores numéricos (veremos em seguida expressões com outros tipos de valores);
 - Existem vários outros códigos de formato como **%d**, **%f** ou **%s**, que exploraremos em exercícios e em outros exemplos neste texto;
 - **<lista de expressões>** é uma lista de expressões separadas por vírgulas, que são calculadas no momento da execução do comando;
 - As expressões na lista são mapeadas uma a uma nos códigos de formato, na mesma sequência em que aparecem na **<frase>**, e a sentença impressa é obtida pela substituição do valor da expressão na posição marcada pelo código de formato.

Saída de dados

- Por exemplo:
 - Sejam os valores 30 e 60 armazenados nas variáveis X e Y, respectivamente;
 - Para exibir estes valores na tela de vídeo, pode-se usar a instrução:

```
printf("PRIMEIRO VALOR: %g - SEGUNDO VALOR: %g", X, Y)
```

Introdução;
Comandos de entrada e saída de dados;
Introdução ao uso do Fluxograma;
Utilizando o ambiente SciNotes;
Exercícios.

INTRODUÇÃO AO USO DO FLUXOGRAMA

Fluxograma

- Com relação à linguagem SciLab, um programa de computador é uma **sequencia de instruções**, ou comandos, **executados sequencialmente**;
- A execução do programa inicia-se em uma primeira instrução, passando a seguir para a segunda instrução, a seguir para a terceira, e assim sucessivamente, até que terminem todas as instruções desse programa;
- Este **fluxo de execução** pode ser representado por um diagrama chamado **fluxograma**. Para um programa que possui 5 instruções genéricas, o fluxograma é ilustrado por:

Fluxograma

- Para um programa que possui 5 instruções genéricas, o fluxograma é ilustrado por:



Fluxograma

- **Exemplo:** Seja a equação do segundo grau $ax^2 + bx + c = 0$;
- Sua solução pode ser obtida através dos seguintes passos:
 1. Atribuir um valor para **a**;
 2. Atribuir um valor para **b**;
 3. Atribuir um valor para **c**;
 4. Calcular o valor de **delta**, onde $\text{delta} = b^2 - 4 * a * c$;
 5. Calcular o valor de x_1 , onde $x_1 = (-b + \text{sqrt}(\text{delta})) / (2 * a)$;
 6. Calcular o valor de x_2 , onde $x_2 = (-b - \text{sqrt}(\text{delta})) / (2 * a)$.

Fluxograma

- **Exemplo:** Para a equação $2x^2 - 4x + 2 = 0$, tem-se a seguinte execução no *console* da SciLab:

```
--> a = 2;
```

```
--> b = -4;
```

```
--> c = 2;
```

```
--> delta = (b * b) - 4*a*c
```

```
delta =
```

```
0.
```

```
-->x1 = ( -b + sqrt(delta) ) / (2*a)
```

```
x1 =
```

```
1.
```

```
-->x2 = ( -b - sqrt(delta) ) / (2*a)
```

```
x2 =
```

```
1.
```

```
-->
```

Introdução;
Comandos de entrada e saída de dados;
Introdução ao uso do Fluxograma;
Utilizando o ambiente SciNotes;
Exercícios.

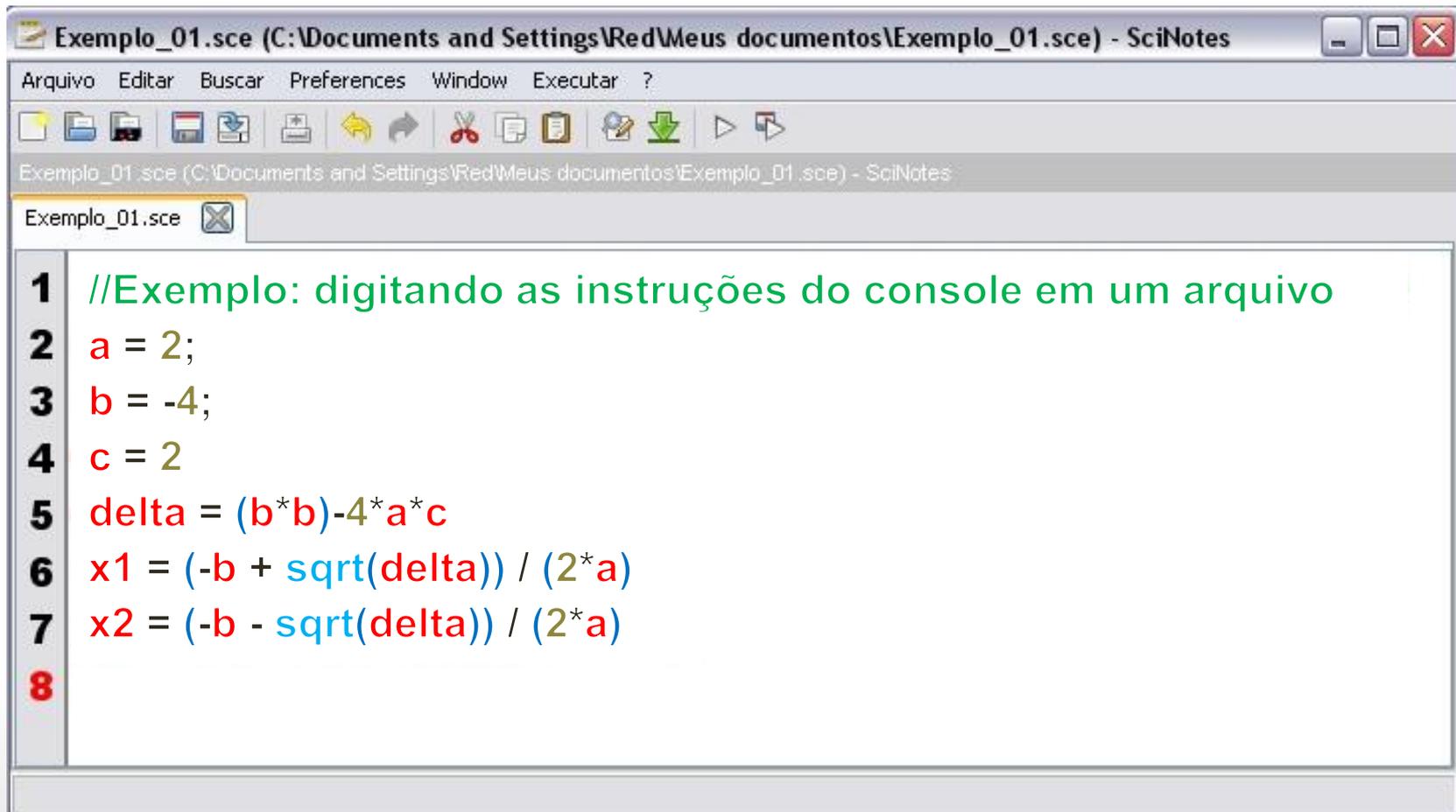
UTILIZANDO O AMBIENTE SCINOTES

SciNotes

- O exemplo da equação do segundo grau, realizado no **console**, poderia ser editado em um arquivo utilizando-se o **SciNotes**;
- Dessa forma, um arquivo seria armazenado em memória secundária para posterior uso;
- A seguir, a tela de edição do **SciNotes**:
 - Para abrir a tela de edição do SciNotes, acione a opção de menu da tela de **console** “*Aplicativos -> SciNotes*”.

SciNotes

- Tela de edição do **SciNotes**:



The screenshot shows the SciNotes application window titled "Exemplo_01.sce (C:\Documents and Settings\Red\Meus documentos\Exemplo_01.sce) - SciNotes". The menu bar includes "Arquivo", "Editar", "Buscar", "Preferences", "Window", and "Executar ?". The toolbar contains icons for file operations and execution. The main editor area shows the following code:

```
1 //Exemplo: digitando as instruções do console em um arquivo
2 a = 2;
3 b = -4;
4 c = 2
5 delta = (b*b)-4*a*c
6 x1 = (-b + sqrt(delta)) / (2*a)
7 x2 = (-b - sqrt(delta)) / (2*a)
8
```

SciNotes

- Exemplo utilizando input e printf:
a = input("Defina um valor para a: ");
b = input("Defina um valor para b: ");
c = input("Defina um valor para c: ");

delta = (b*b)-4*a*c;
x1 = (-b + sqrt(delta)) / (2*a);
x2 = (-b - sqrt(delta)) / (2*a);

printf("A raiz x1 é %g.\n", x1);
printf("A raiz x2 é %g.", x2);

Introdução;
Comandos de entrada e saída de dados;
Introdução ao uso do Fluxograma;
Utilizando o ambiente SciNotes;

Exercícios.

EXERCÍCIOS

Exercícios

- Codifique os programas a seguir na linguagem Scilab. Utilize comentários e mensagens textuais para o usuário.
1. Codifique um programa que leia dois valores. O programa calcula a soma desses valores, armazenando-a em uma variável. A seguir o programa imprime o resultado da soma.
 2. Modifique o programa anterior, onde o resultado de (1) será o numerador de uma divisão. O denominador será um novo valor lido pelo teclado. O programa imprime o resultado final da divisão.

Exercícios

3. Crie um programa que imprima a hipotenusa de um triângulo retângulo de acordo com a leitura de seus catetos.
4. Crie um programa que leia do teclado um valor de temperatura em graus **Celsius** ($^{\circ}\text{C}$), calcule e imprima essa temperatura em graus **Fahrenheit** ($^{\circ}\text{F}$) e em graus **Kelvin** ($^{\circ}\text{K}$).

OBS.: $^{\circ}\text{F} = ^{\circ}\text{C} \times 1.8 + 32$
 $^{\circ}\text{K} = ^{\circ}\text{C} + 273.15$

Próxima aula prática: resolução de exercícios com o uso do SciLab e SciNotes.

Próxima aula teórica: Comandos de desvio de fluxo; Operadores relacionais; Fluxogramas.

FIM!

DÚVIDAS?