

Métodos/Procedimentos/Funções

- Aula passada vimos como funciona, agora vamos às utilidades:
 - Eliminar redundância
 - Facilitar depuração
 - Encapsular
 - agrupar métodos em módulos por semelhança ou “afinidade”
 - Modularizar
 - permitir que módulos equivalentes sejam substituídos.

Exemplo: Concessão de crédito para aquisição de casa

- Um estudo estatístico encomendado por um banco mostra que a chance de um empréstimo dar o maior lucro possível depende apenas de:
 - valor do **crédito**
 - **renda** familiar
 - número de **pessoas** na família.
- Um programa foi criado para auxiliar o gerente das agências na decisão de conceder ou não o empréstimo, mas o programador escreveu tudo no main...

Exemplo e mais!

- CasaZero
- CasalInicial
- CasaMetodo
- TestaCasa/Casa
-
- API: <http://docs.oracle.com/javase/6/docs/api/>
- Math:
<http://docs.oracle.com/javase/6/docs/api/java/lang/Math.html>
- TestaCasa2

Tipos abstratos de dados (TAD)

- Pode se fazer tudo com tipos primitivos, embora seja difícil organizar a informação - por exemplo: como armazenar uma tabela usando tipos primitivos??
- Ganha-se em conforto e flexibilidade quando se permite ao programador definir seus próprios tipos de dados.
- Historicamente, Arrays e Strings são os primeiros **tipos abstratos de dados**.
- Os objetos de Java são uma extensão do conceito de ADT

A idéia é simples...

Reservar um bloco de memória e armazenar nele
o que quiser...

Array - o exemplo mais simples

- Um array é uma “lista estruturada de variáveis de tipo igual” o conceito de array existe em praticamente todas as linguagens de programação.
- Em geral o que dá a estrutura são índices: inteiros que informam qual posição na lista queremos acessar.
- O bloco de memória é dividido em “n” partes. Em Java, se o array se chama A, cada parte é acessada acrescentando, entre colchetes, o índice, p. ex. A[0], A[100], A[i]
- Como é que se faz essa “mágica”??

arrays em programas Java

- Inicialização direta (ou on-line):
 - `int[] A={200, 300, -400, 25};`
 - aloca um bloco de memória para 4 inteiros e armazena os valores em cada posição.
- Alocação dinâmica:
 - `int[] A;`
 - `A=new int[25];`
 - aloca um bloco de memória para 25 inteiros, não inicializa as posições.
- `A[i]` é um inteiro e pode (deve) ser usado como tal.

E na memória, como é??

- Em especial, o que está armazenado na variável A?
 - todos os valores simultaneamente?
 - o primeiro valor?
 - o último valor?
 - NDA

NDA

- Em A está armazenada a **REFERÊNCIA** para o bloco de memória que efetivamente contém os valores!!
- **Referência** pode ser interpretado como ENDEREÇO DE MEMÓRIA (lembra do HIPO??)

Mapa da memória

- `int[] A; // cria a referência para o bloco de memória`
- `A=new int[25]; // aloca (reserva) o bloco`

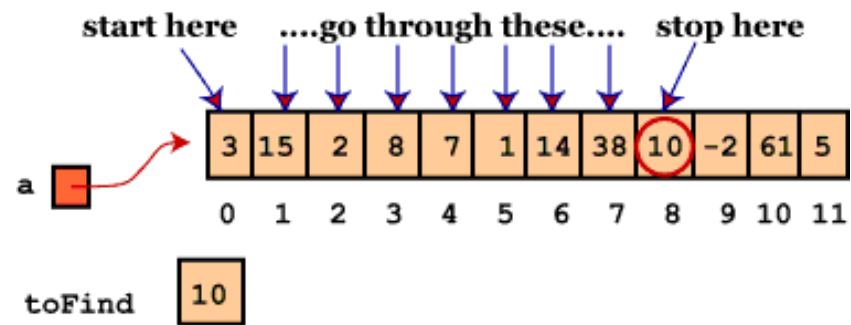
e se fizer:

- `int[] A={23, 54, 56};`
- `A=new int[25];`
-
- O que acontece com o 23, 54 e 56???
-
- PERRRRRDEU PREIBÓI!!!
-
- (Fazer mapa da memória)

Qual a vantagem de usar arrays?

- Não precisar criar no código tantos nomes de variáveis quanto necessário (D_0 , D_1 , D_2 ...)
- Escrever métodos que retornam mais de um valor.
- Tratar funções **multidimensionais**: $f(x)$ onde x pertence a \mathbb{R}^3
- Representar sistemas lineares: João e Maria tem ao todo 5 balas, Maria tem 3 balas mais que João. O que João faz?
 - Chora, pq tem só uma bala e Maria tem 4...
- Representar imagens, grafos, ...

1 dimensão

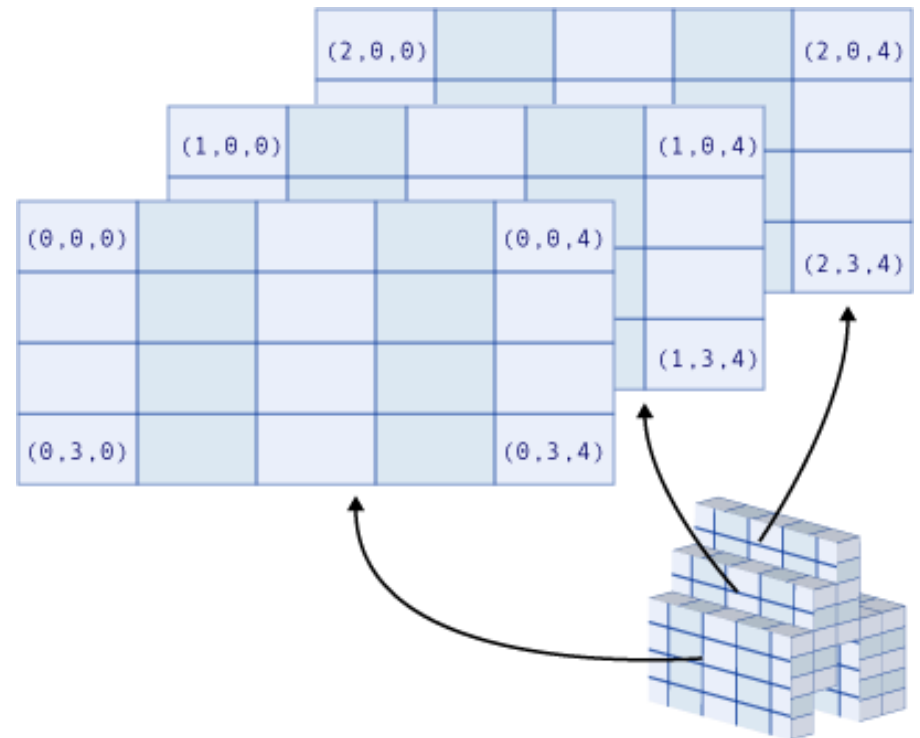
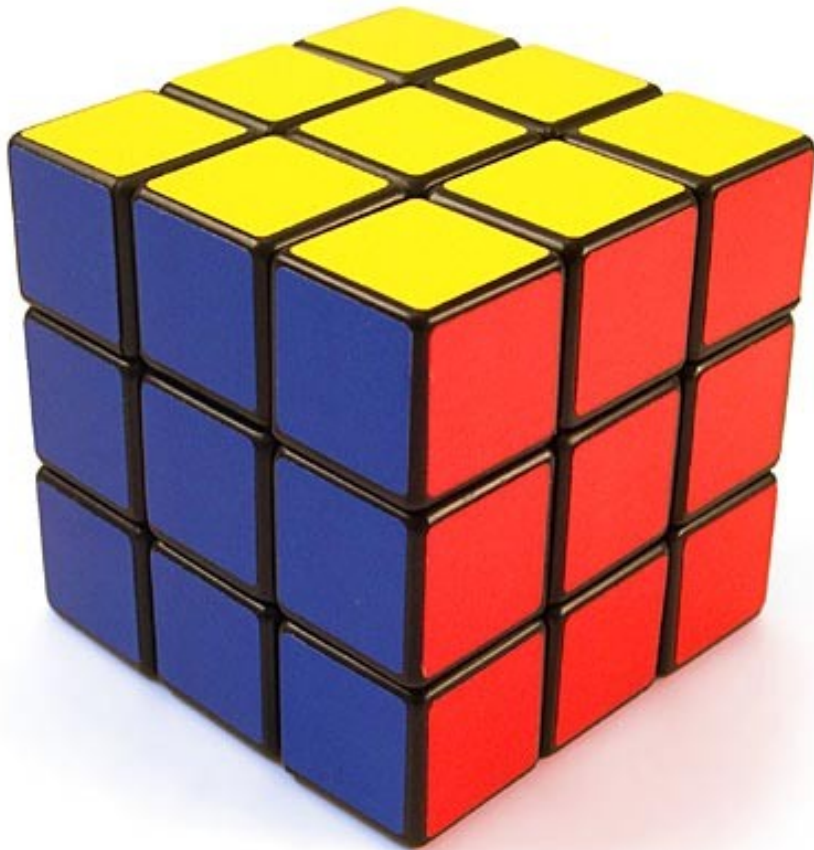


2 dimensões

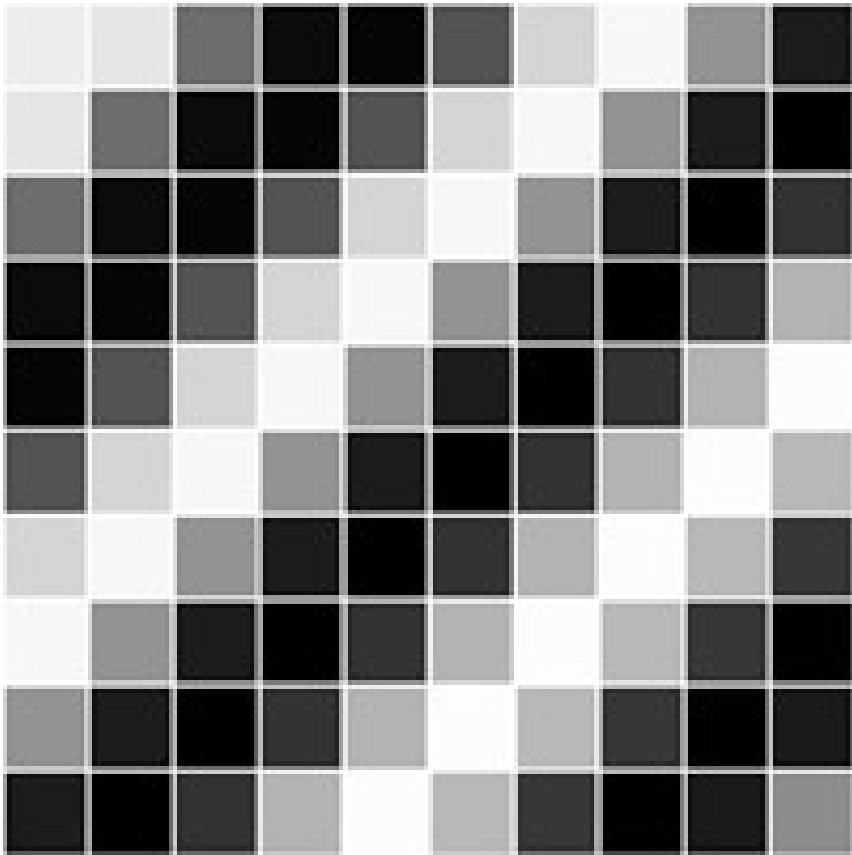
A	0	1	2	3
0	8	2	6	5
1	6	3	1	0
2	8	7	9	6

	0	1	2	3
0	[0][0]	[0][1]	[0][2]	[0][3]
1	[1][0]	[1][1]	[1][2]	[1][3]
2	[2][0]	[2][1]	[2][2]	[2][3]
3	[3][0]	[3][1]	[3][2]	[3][3]
4	[4][0]	[4][1]	[4][2]	[4][3]

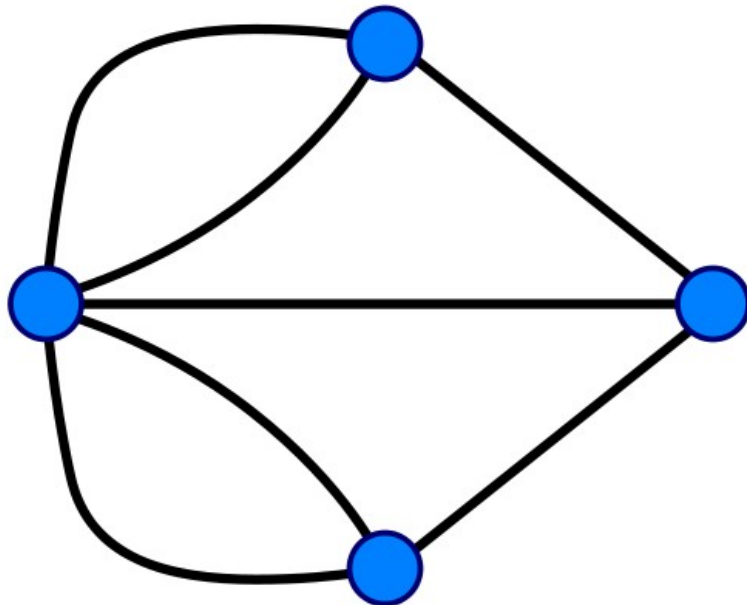
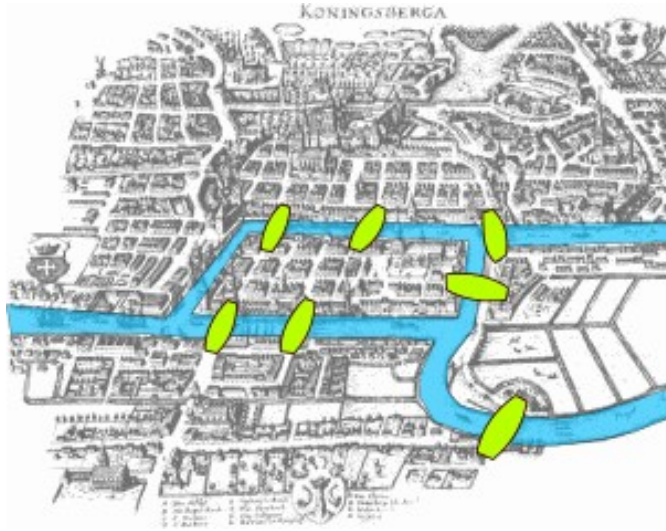
3 dimensões



Imagens



Grafos



Euler 1736

Königsberg

É possível partir de algum ponto da cidade, passar exatamente uma vez por cada ponte e voltar ao mesmo ponto?

Matriz de adjacência

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

