

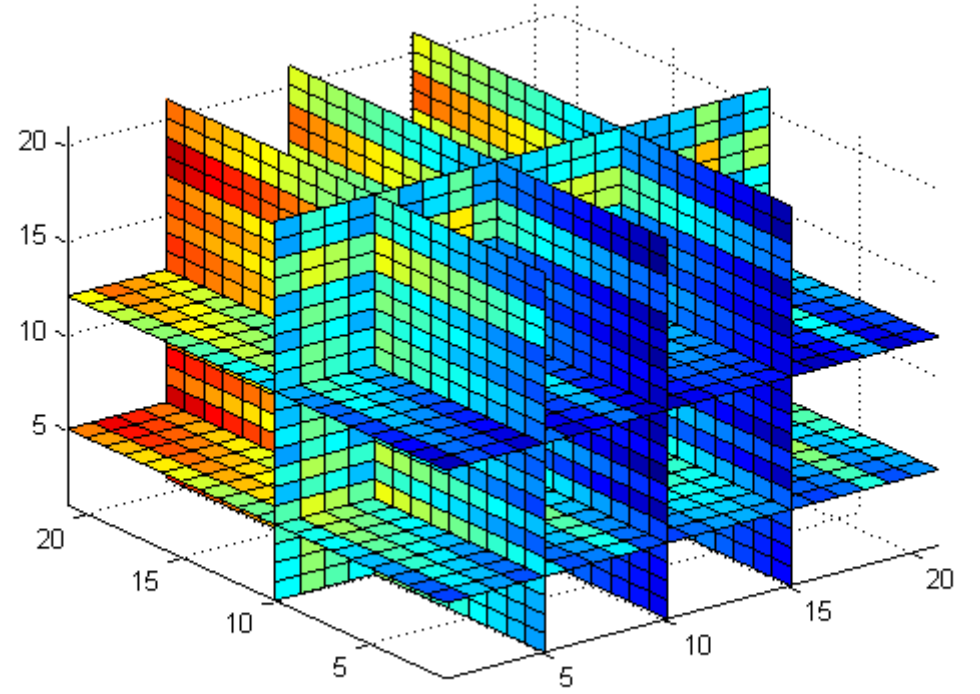
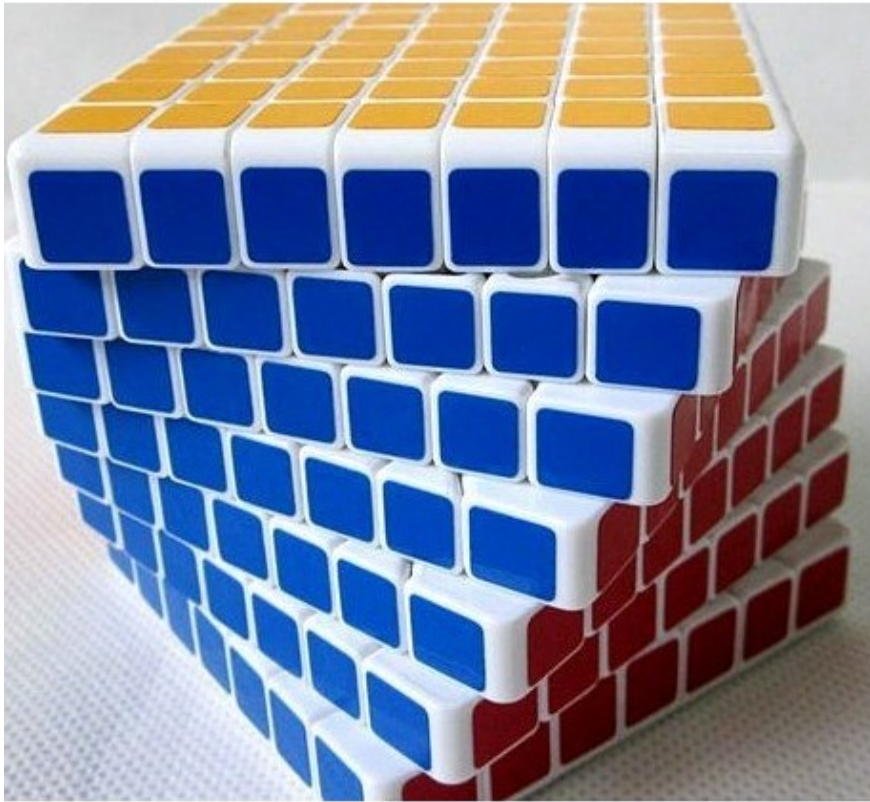


Processamento da Informação

Vetores – Parte 1 (Arrays / Arranjos)

Prof. Jesús P. Mena-Chalco
CMCC/UFABC

Q2/2018



Armazenar 10 inteiros em um programa...

- Usando variáveis:

```
int a0 = 6;  
int a1 = 30;  
int a2 = 82;  
int a3 = 0;  
int a4 = 100;  
int a5 = 8;  
int a6 = 14;  
int a7 = 83;  
int a8 = 11;  
int a9 = 20;
```

- Usando um vetor:

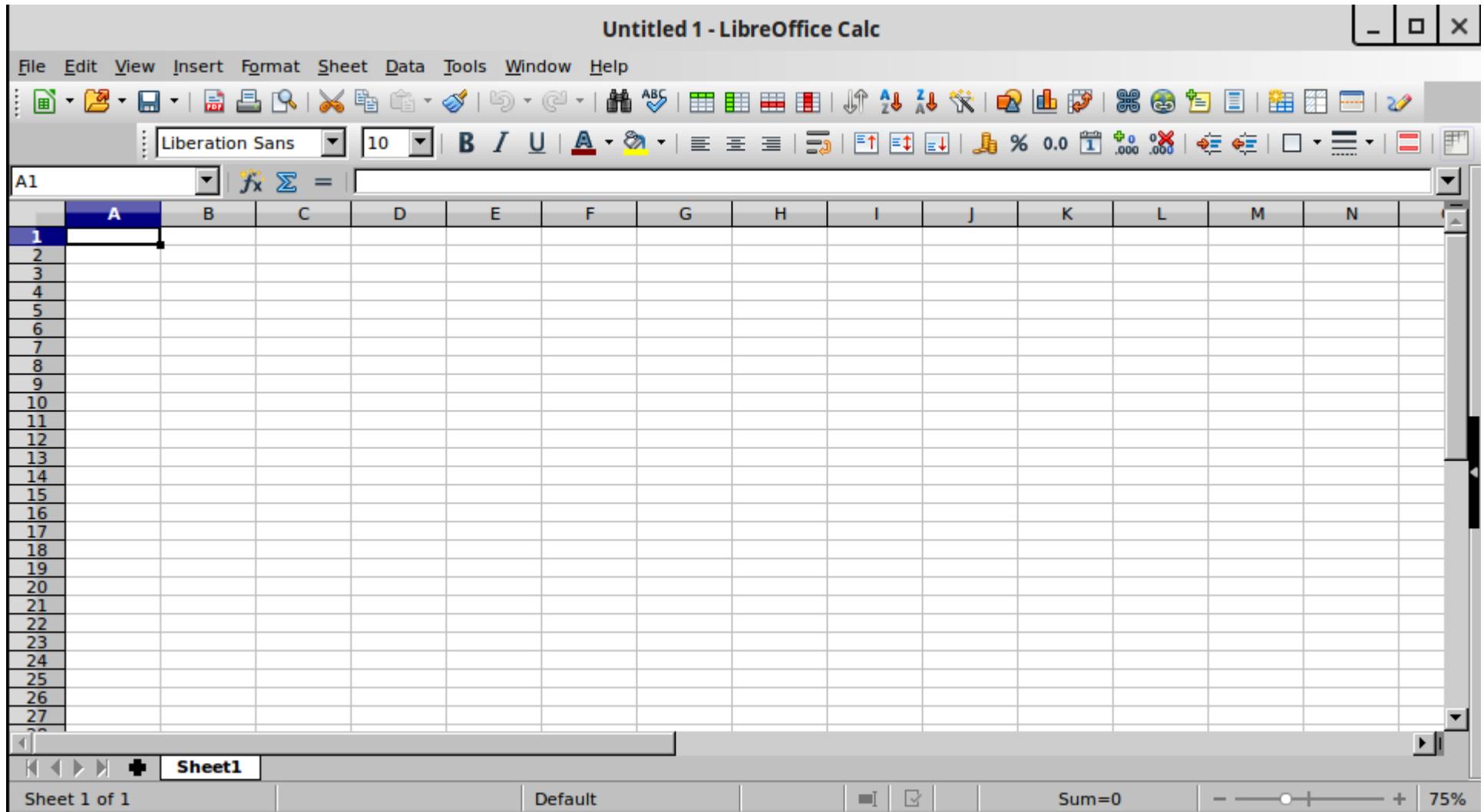
```
int a[] = {6, 30, 82, 0, 100, 8, 14, 83, 11, 20};
```

Para acessar a um elemento, use um índice.

a[0] → 6

a[4] → 100

Vetores, são realmente úteis?



Vetores, são realmente úteis?

Permitem organizar e armazenar **grandes quantidades** de dados.

Por exemplo:

- Todos os RAs de todos os alunos da UFABC
- 300 números associados a experimentos.
- 1 milhão de caracteres de um livro.
- 4 bilhões de nucleotídeos de um sequência de DNA.
- ...

Vetor

Um **vetor** (ou array) é o nome a uma matriz unidimensional.

X X X X X X X X X X X X
1 row of 12

X X X X X X
X X X X X X
2 rows of 6

X X X X
X X X X
X X X X
3 rows of 4

X X X
X X X
X X X
X X X
4 rows of 3

X X
X X
X X
X X
X X
X X
6 rows of 2

X
X
X
X
X
X
X
X
X
X
X
X
12 rows of 1

Vetor

Os elementos de um vetor são armazenados/**alocados de forma consecutiva** na memória.

Os elementos são **acessados por seu índice** dentro do array.

	0	1	2	3	4	...
X:=	10	4	-95	37	2910	

Memória (hardware)



RAM



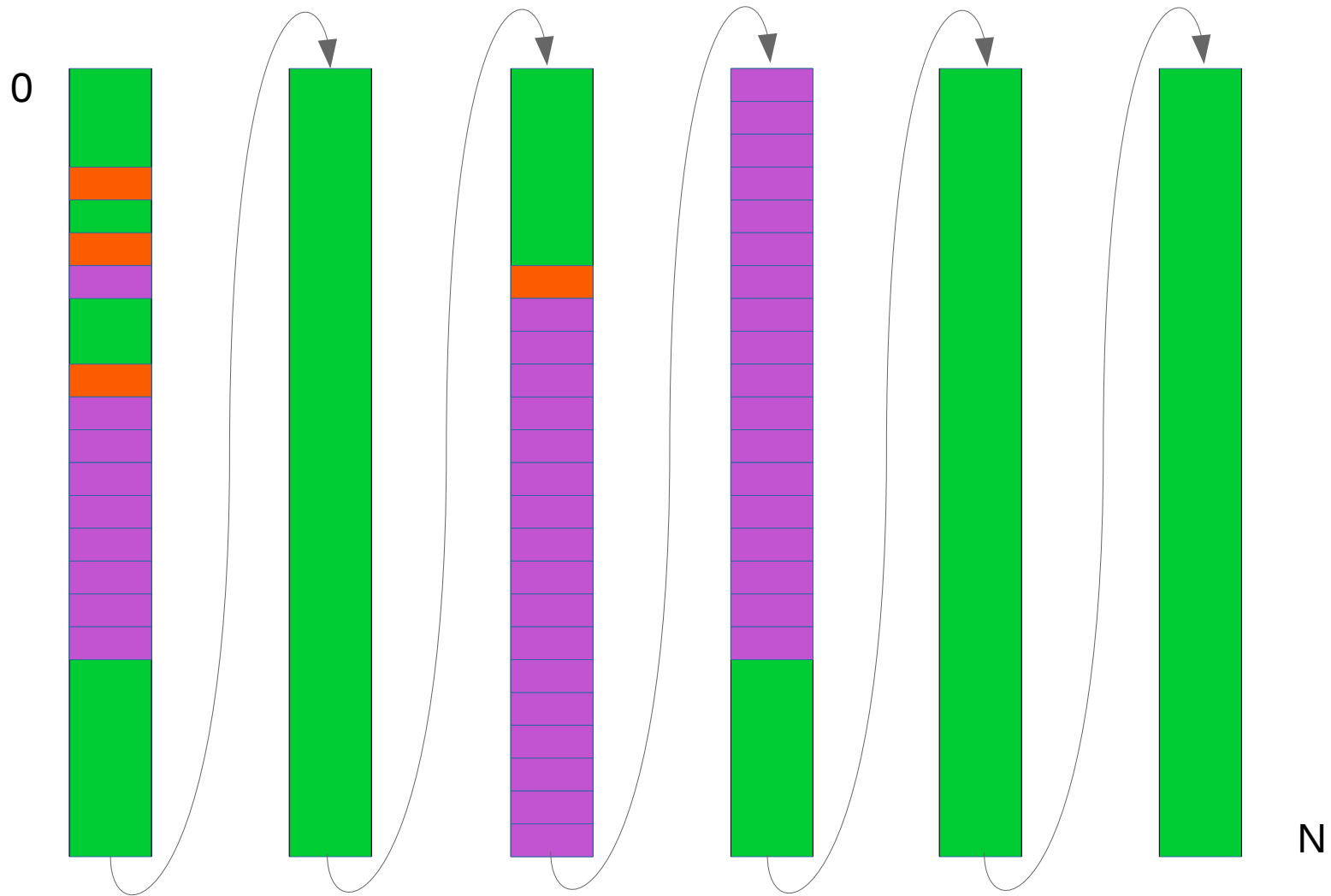
HDD



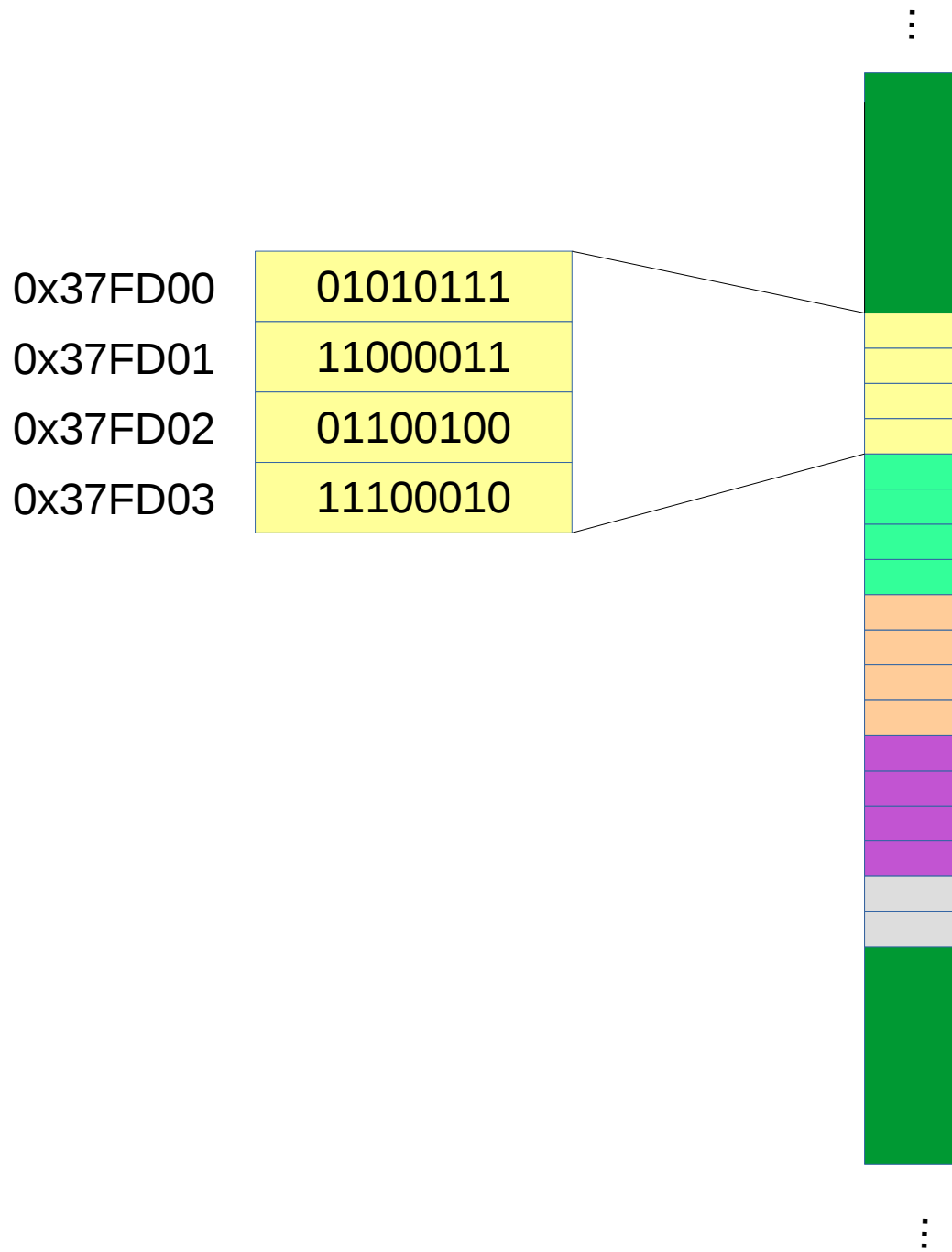
SSD



Memória (hardware)



Geralmente o endereço do 1ro byte é o endereço do objeto.



4 bytes

Category	Types	Size (bits)
Integer	byte	8
	char	16
	short	16
	int	32
Floating-point	long	64
	float	32
Other	double	64
	boolean	1
	void	--

Vetor em Java

Declaração de uma variável que representa um vetor de 13 inteiros

```
int a;  
a = 45;  
  
int array[];  
array = new int[13];
```

vetor :=

0	1	2	3	4	5	6	7	8	9	10	11	12
10	4	-95	37	2910	-6	10	4	-95	37	2910	3	0
1	2	3	4	5	6	7	8	9	10	11	12	13

Índice / Deslocamento



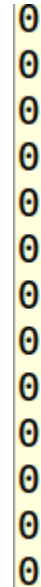
Posição



O array contém 13 Elementos

Exemplo 01: Definição

```
int i;  
  
int array[];  
array = new int[13];  
  
for(i=0; i<13; i=i+1) {  
    System.out.println( array[i] );  
}
```



0
0
0
0
0
0
0
0
0
0
0
0
0

Exemplo 02: Atribuição

```
int i;  
  
int array[];  
array = new int[13];  
  
array[4] = 5678;  
array[12] = 1111;  
  
for(i=0; i<13; i=i+1) {  
    System.out.println( array[i] );  
}
```

```
0  
0  
0  
0  
5678  
0  
0  
0  
0  
0  
0  
0  
0  
1111
```

Exemplo 03: Tamanho

```
int i;  
  
int array[];  
array = new int[13];  
  
array[4] = 5678;  
array[12] = 1111;  
  
for(i=0; i<array.length/2; i=i+1) {  
    System.out.println( array[i] );  
}
```

```
0  
0  
0  
0  
5678  
0
```

Exemplo 04: Inicialização

```
int i;  
int a[] = {10,12,14,16,18};  
  
for(i=0; i<a.length/2; i=i+1) {  
    System.out.println( a[i] );  
}
```

```
10  
12
```

Em concreto

Um vetor é uma **coleção de valores**.

Três importantes características:

- Os vetores representam um grupo de dados relacionados.
- Todos os dados devem ter o mesmo tipo.
- O tamanho do vetor é definido na sua criação/definição.

Terminologia : Palavra reservada **new**

```
int v[] = new int[10];
```



Usada para criar um novo **objeto**.
Em Java, um vetor é um objeto.

Não trataremos de detalhes de Programação Orientada a Objetos nessa disciplina.

Apenas saiba que com **new** podemos criar um **objeto** que **represente uma coleção de elementos**.

Criando vetores de diferentes tipos de dados

```
int a[] = new int[100];  
double b[] = new double[200];  
float c[] = new float[1000];
```

Category	Types	Size (bits)
Integer	byte	8
	char	16
	short	16
	int	32
	long	64
Floating-point	float	32
	double	64
Other	boolean	1
	void	--



Exercícios

Exercício 1: Soma dos elementos

Crie um método que permita somar todos os elementos de um vetor de inteiros.

Assinatura:

```
static int somarElementos ( int v[ ] )
```

Exemplo:

v = [1, 3, 5]

Resultado: 9

v = [10, 20, 30, 40]

Resultado: 100

Exercício 1: Soma dos elementos

```
static int somarElementos(int v[]) {  
    int i;  
    int soma = 0;  
  
    for (i=0; i<v.length; i=i+1) {  
        .....  
        soma = soma + v[i];  
    }  
  
    return soma;  
}
```

Exercício 1: Soma dos elementos

```
static int somarElementos(int v[]) {  
    int i;  
    int soma = 0;  
  
    for (i=0; i<v.length; i=i+1) {  
        soma = soma + v[i];  
    }  
  
    return soma;  
}
```

```
public static void main(String []args) {  
    int vetor[ ] = {10, 20, 30, -70};  
    System.out.println( somarElementos(vetor) );  
}
```

Exercício 2: Busca de um elemento

Crie um método que permita procurar um elemento em um vetor.

- Se existir o elemento devolva seu índice no vetor.
- Caso contrário devolva -1

Assinatura:

```
static int buscaElemento (int v[], int x)
```

Exemplos:

```
v = [1,2,3]  
x = 3  
Resposta: 2
```

```
v = [1,2,3]  
x = 30  
Resposta: -1
```

Exercício 2: Busca de um elemento

```
static int buscaElemento (int v[], int x) {  
    int i;  
  
    for (i=0; i<v.length; i=i+1) {  
        if (x==v[i])  
            return i;  
    }  
  
    return -1;  
}
```

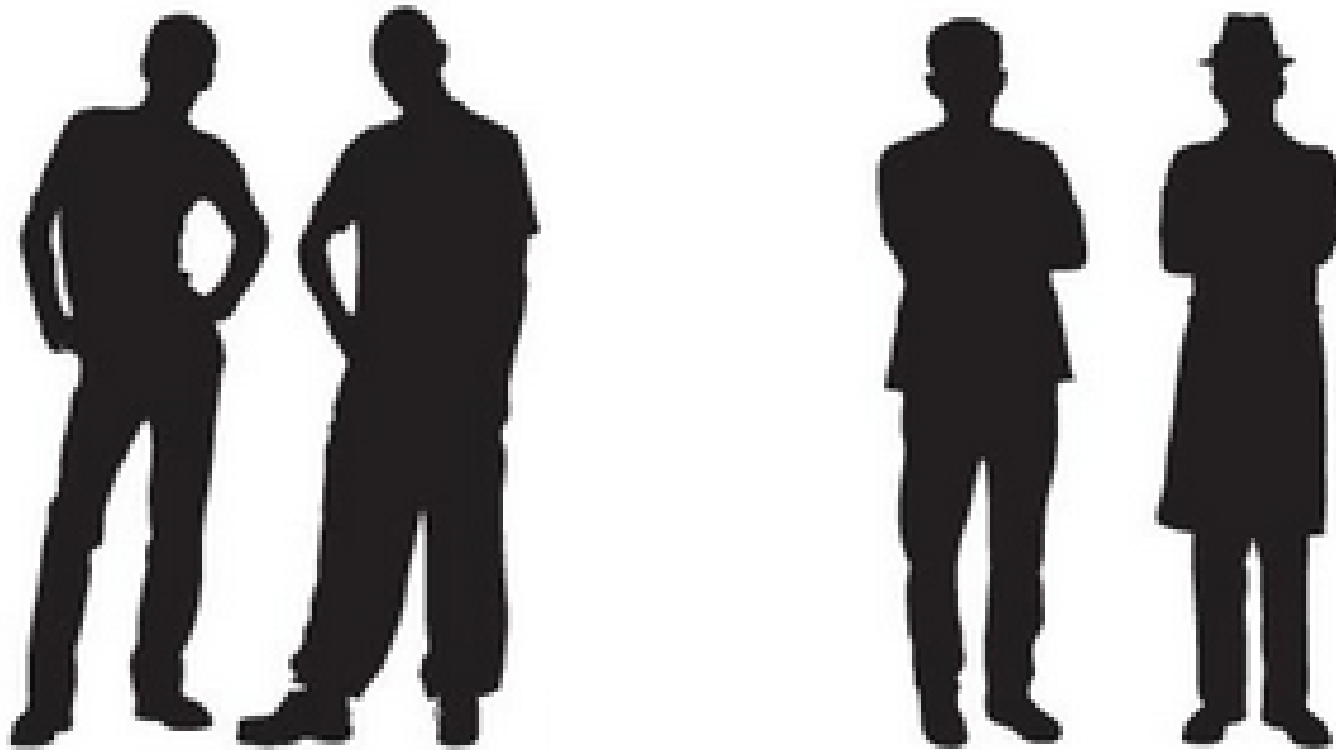

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur vel est augue. Donec aliquam laoreet ipsum, ac sagittis odio aliquam quis. In eleifend est tortor, dapibus posuere odio consectetur id. Donec varius eget est eu luctus. Phasellus scelerisque, nunc laoreet feugiat sagittis, risus neque condimentum nisi, eget feugiat turpis purus quis urna. Mauris tempor eros in turpis tincidunt, bibendum euismod est pretium. Praesent ligula dui, fermentum a porttitor vitae, pulvinar ut odio. Suspendisse vitae pretium dolor, at sodales mauris.

Sed consequat purus nec bibendum suscipit. Donec ultricies euismod enim, quis interdum mi ornare et. Ut facilisis elit vitae elementum fringilla. Pellentesque sed orci iaculis, imperdiet elit et, mollis enim.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur vel est augue. Donec aliquam laoreet ipsum, ac sagittis odio aliquam quis. In eleifend est tortor, dapibus posuere odio consectetur id. Donec varius eget est eu luctus. Phasellus scelerisque, nunc laoreet feugiat sagittis, risus neque condimentum nisi, eget feugiat turpis purus quis urna. Mauris tempor eros in turpis tincidunt, bibendum euismod est pretium. Praesent ligula dui, fermentum a porttitor vitae, pulvinar ut odio. Suspendisse vitae pretium dolor, at sodales mauris.

Sed consequat purus nec bibendum suscipit. Donec ultricies euismod enim, quis interdum mi ornare et. Ut facilisis elit vitae elementum fringilla. Pellentesque sed orci iaculis, imperdiet elit et, mollis enim.

Programação elegante?

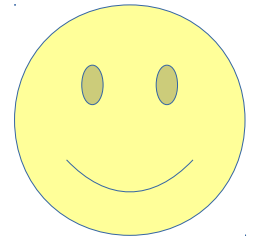


Exercício 2: Busca de um elemento

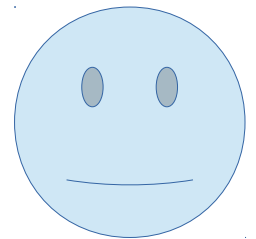
```
static int buscaElemento (int v[], int x) {  
    int i=v.length-1;  
  
    while (i>=0 && x!=v[i]) {  
        i = i-1;  
    }  
  
    return i;  
}
```

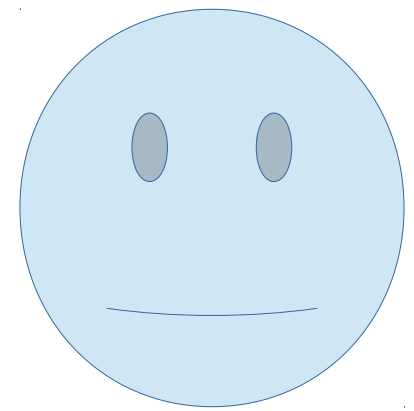
Programação elegante

```
static int buscaElemento (int v[], int x) {  
    int i=v.length-1;  
  
    while (i>=0 && x!=v[i]) {  
        i = i-1;  
    }  
  
    return i;  
}
```



```
static int buscaElemento (int v[], int x)  
{int i=v.length-1;  
while (i>=0 && x!=v[i]) {  
    i = i-1;  
}  
return i;}
```





```
import java.util.Scanner;

public class DivReal {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int a,b;
        float divisao;
        a= sc.nextInt();
        b= sc.nextInt();
        divisao= a/ (float) b;
        System.out.printf( "%.2f\n", divisao);
    }
}
```

Exercício 2: Busca de um elemento

```
public class Exercicio02
{
    static int buscaElemento (int v[], int x) {
        int i=v.length-1;

        while (i>=0 && x!=v[i]) {
            i = i-1;
        }

        return i;
    }

    public static void main(String[] args) {
        int v[] = {10, 20, 30, -70};
        System.out.println( buscaElemento(v, 30) );
        System.out.println( buscaElemento(v, 15) );
    }
}
```

2
-1

Exercício 3: Maior elemento

Crie um método que permita devolver o maior elemento contido em um vetor.

Assinatura:

```
static int maiorElemento (int v[])
```

Exemplos:

```
v = [1,2,3]  
Resposta: 3
```

Exercício 3: Maior elemento

```
static int maiorElemento(int v[]) {  
    int i;  
    int maior = v[0];  
  
    for (i=1; i<v.length; i=i+1) {  
        if (maior<v[i])  
            maior = v[i];  
    }  
  
    return maior;  
}
```


Exercício 3: Maior elemento

```
static int maiorElemento(int v[]) {
    int i;
    int maior = v[0];

    for (i=1; i<v.length; i=i+1) {
        if (maior<v[i])
            maior = v[i];
    }

    return maior;
}
```

```
public static void main(String []args) {
    int i;
    int vetor[ ] = {10, 20, 30, -70};

    System.out.println( maiorElemento(vetor) );
}
```

30