

**INSTITUTO  
FEDERAL**  
Santa Catarina

## **Arrays em Java**

### **Pensamento Computacional e Algoritmos**

Análise e Desenvolvimento de Sistemas

Prof. Adriano Lima

[adriano.lima@ifsc.edu.br](mailto:adriano.lima@ifsc.edu.br)



## cores

#111027

#277756

#16ABCD

#FFF4EC

#C74E23

## fontes

Fira Sans Extra Condensed

Ubuntu

Roboto Mono



# Problema

calcular média

pseudo-código



# Problema

calcular média

pseudo-código

$$m\acute{e}dia = \frac{n\acute{u}mero_1 + n\acute{u}mero_2}{2}$$

# Problema

calcular média

$$média = \frac{número_1 + número_2}{2}$$

pseudo-código

entrada: *número1*, *número2*

saída: *média*

Início

1. *média*  $\leftarrow$  *número1* + *número2*

2. *média*  $\leftarrow$  *média* / 2

3. Retorna *média*

Fim

# Problema

calcular média

$$média = \frac{número_1 + ... + número_n}{n}$$

pseudo-código

entrada: *número1*, *número2*

saída: *média*

Início

1. *média*  $\leftarrow$  *número1* + *número2*

2. *média*  $\leftarrow$  *média* / 2

3. Retorna *média*

Fim

# Problema

calcular média

$$média = \frac{número_1 + ... + número_n}{n}$$

pseudo-código

entrada: *coleção*

saída: *média*

Início

1. *soma, contador*  $\leftarrow$  0
2. Para cada *número* da *coleção*:
3.     *soma*  $\leftarrow$  *soma* + *número*
4.     *contador*  $\leftarrow$  *contador* + 1
5. *média*  $\leftarrow$  *soma* / *contador*
6. Retorna *média*

Fim



# Coleções

um nome para se referir ao todo



# Coleções

um nome para se referir ao todo



**classe**



# Coleções

um nome para se referir ao todo



**biblioteca**



**classe**



# Coleções

um nome para se referir ao todo



**biblioteca**



**classe**



**alcateia**

# Coleções

números

notas = {6, 7.5, 9, 8}



# Coleções

números

notas = {6, 7.5, 9, 8}

textos

alunos = {Carlos, Ana, Joaquim, Eduarda}



# Coleções

**números**

notas = {6, 7.5, 9, 8}

**textos**

alunos = {Carlos, Ana, Joaquim, Eduarda}

**símbolos**

operações = {+, -, \*, /}

# Coleções

**números**

notas = {6, 7.5, 9, 8}

**textos**

alunos = {Carlos, Ana, Joaquim, Eduarda}

**símbolos**

operações = {+, -, \*, /}

**valores lógicos**

booleanos = {verdadeiro, falso}

# Coleções

**números**

notas = {6, 7.5, 9, 8}

**textos**

alunos = {Carlos, Ana, Joaquim, Eduarda}

**símbolos**

operações = {+, -, \*, /}

**valores lógicos**

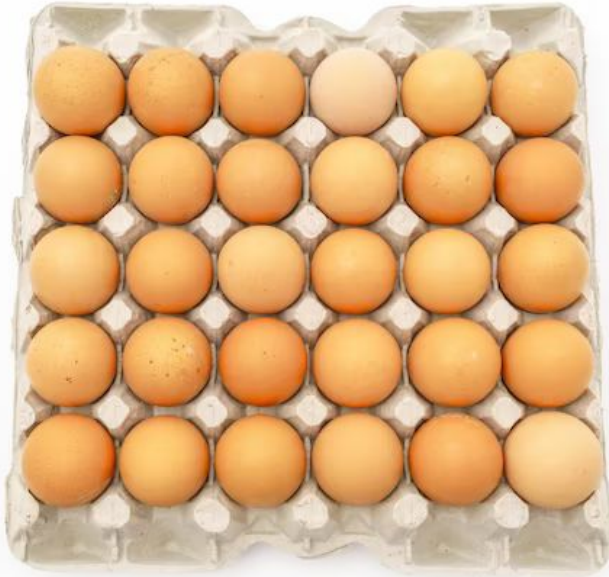
booleanos = {verdadeiro, falso}

**coleções**

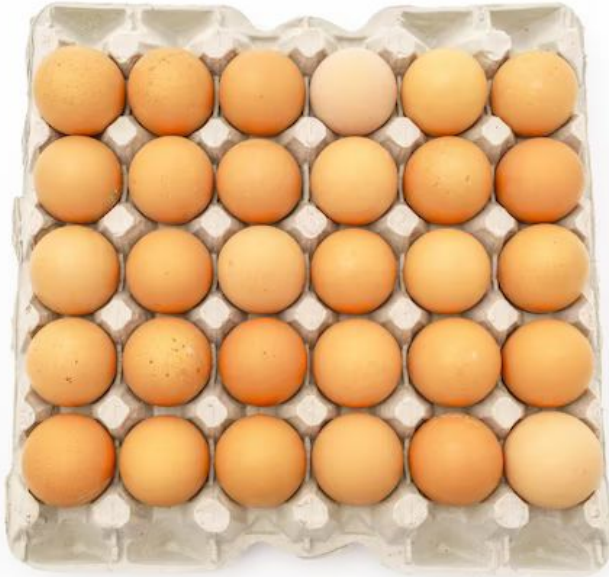
duplas = {{Tom, Jerry}, {Batman, Robin}, {Zé Colmeia, Catatau}}



# Arrays em Java



# Arrays em Java



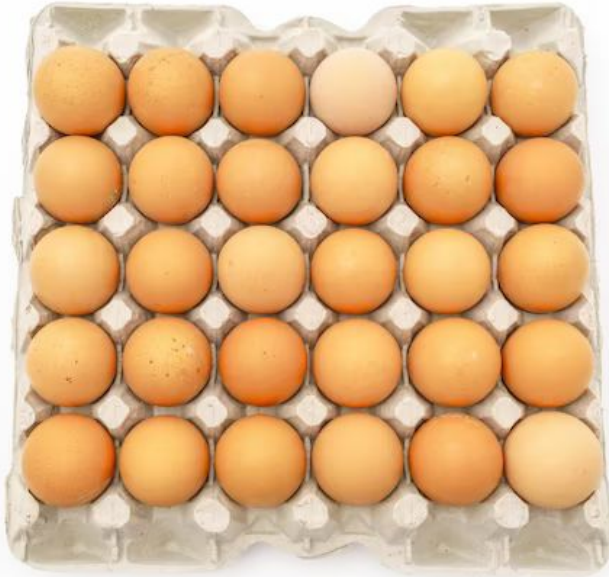
- grupo de valores que podem ser atribuídos a uma única variável

# Arrays em Java



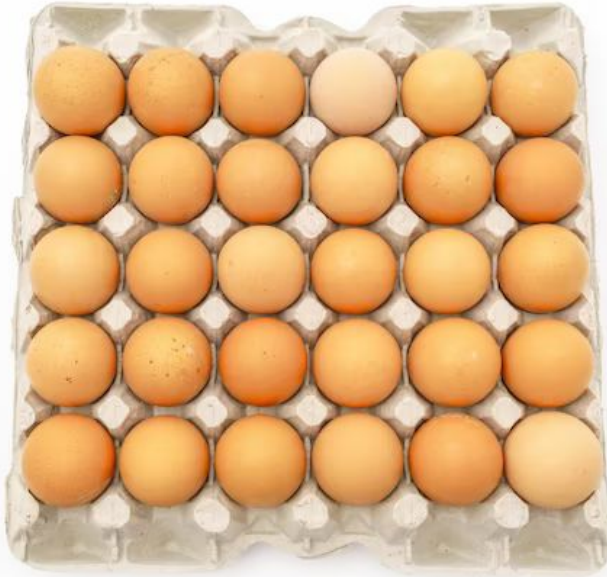
- grupo de valores que podem ser atribuídos a uma única variável
- do inglês: "matriz"

# Arrays em Java



- grupo de valores que podem ser atribuídos a uma única variável
- do inglês: "matriz"
- os valores em um array são chamados **elementos**

# Arrays em Java



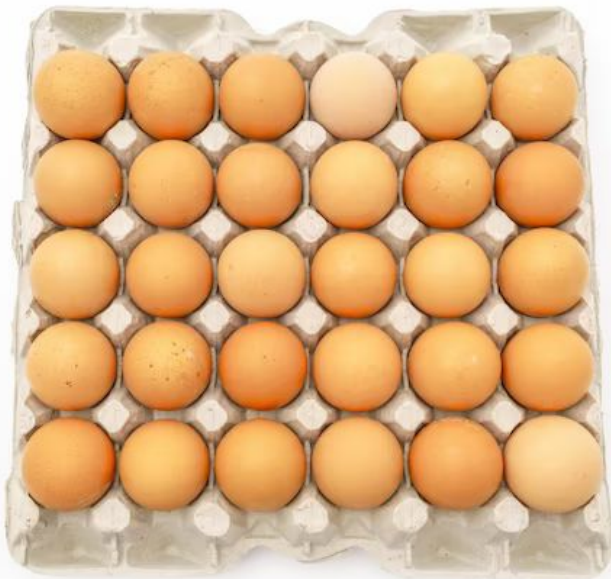
- grupo de valores que podem ser atribuídos a uma única variável
- do inglês: "matriz"
- os valores em um array são chamados **elementos**
  - os valores de um mesmo array devem ser do **mesmo tipo**

# Arrays em Java



- grupo de valores que podem ser atribuídos a uma única variável
- do inglês: "matriz"
- os valores em um array são chamados **elementos**
  - os valores de um mesmo array devem ser do **mesmo tipo**
- **ordenado** a posição dos elementos no array é fixa

# Arrays em Java



- grupo de valores que podem ser atribuídos a uma única variável
- do inglês: "matriz"
- os valores em um array são chamados **elementos**
  - os valores de um mesmo array devem ser do **mesmo tipo**
- **ordenado** a posição dos elementos no array é fixa
- a posição numérica dos elementos é chamada de **índice** (começa em 0)

# Arrays em Java

0x82	0x8A	0x92	0x9A	0xA2	0xAA	0xB2	0xBA

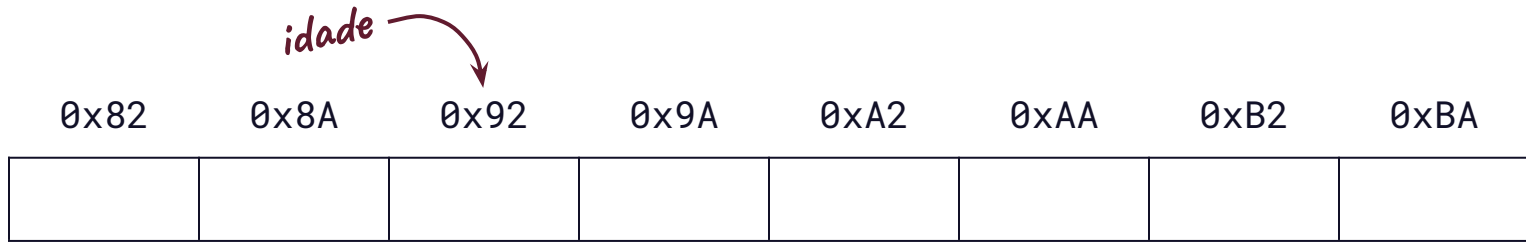


# Arrays em Java

0x82	0x8A	0x92	0x9A	0xA2	0xAA	0xB2	0xBA

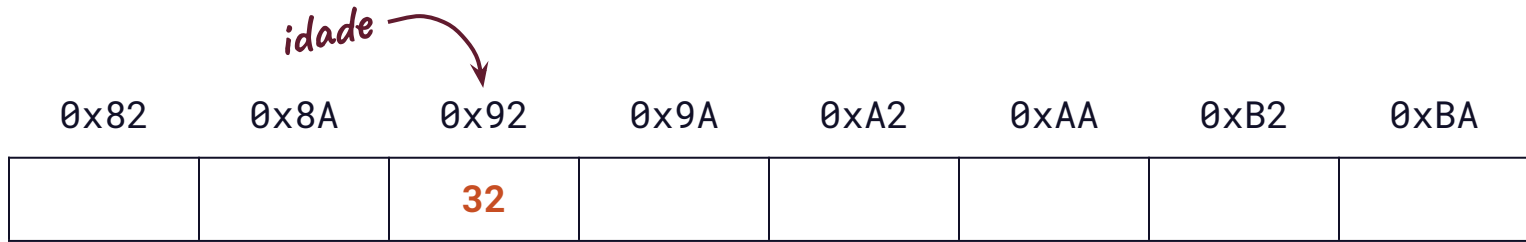
```
int idade = 32;
```

# Arrays em Java



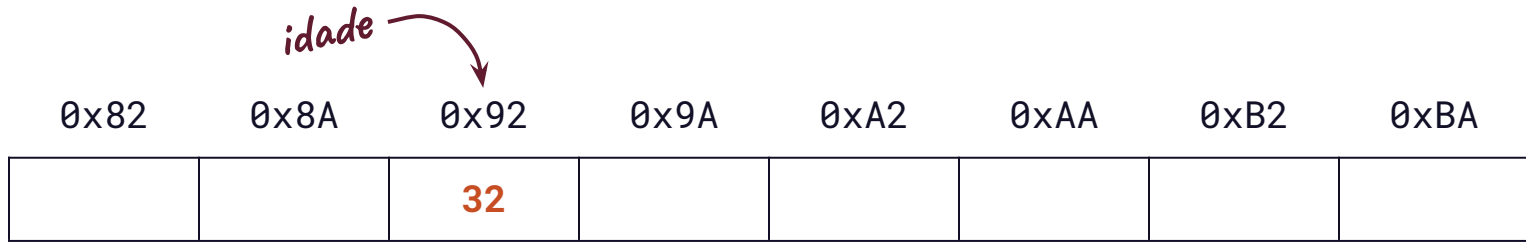
```
int idade = 32;
```

# Arrays em Java



```
int idade = 32;
```

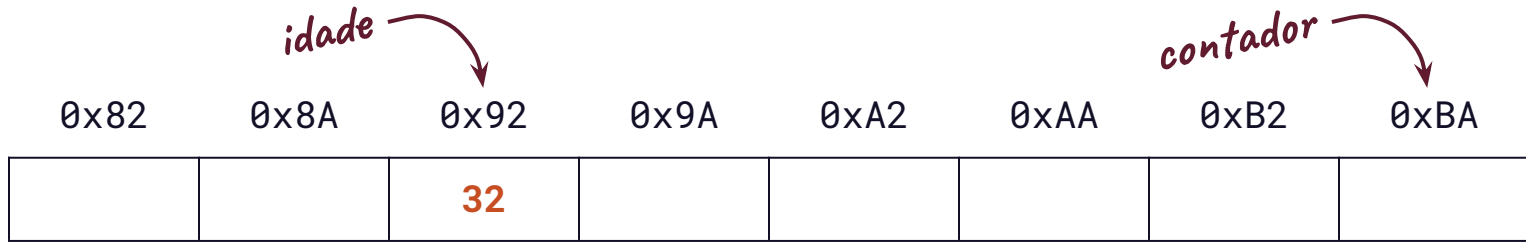
# Arrays em Java



```
int idade = 32;
```

```
int contador = 0;
```

# Arrays em Java



```
int idade = 32;
```

```
int contador = 0;
```

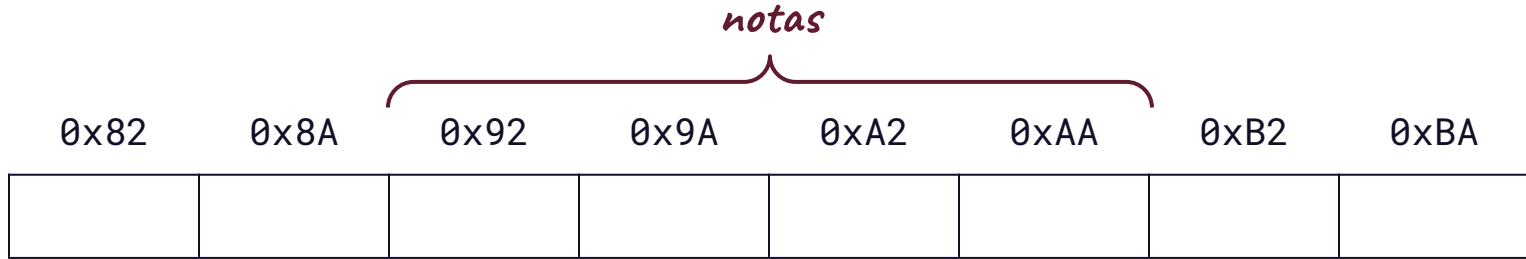
# Arrays em Java



```
int idade = 32;
```

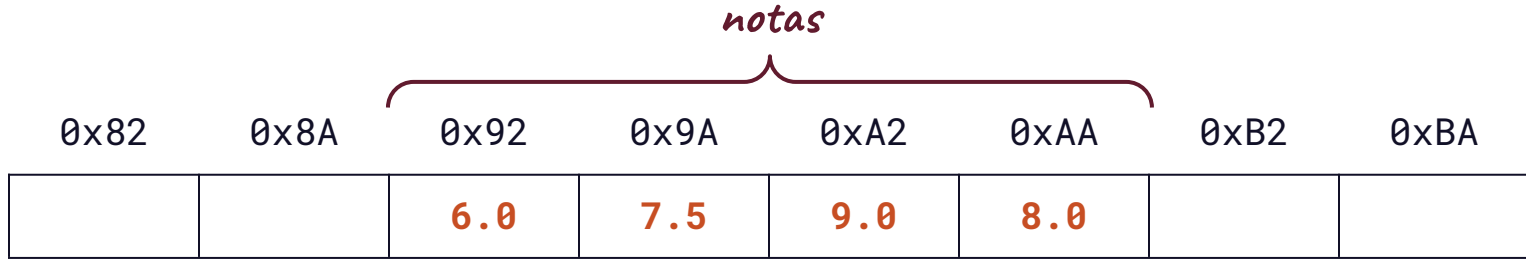
```
int contador = 0;
```

# Arrays em Java



**notas = {6, 7.5, 9, 8}**

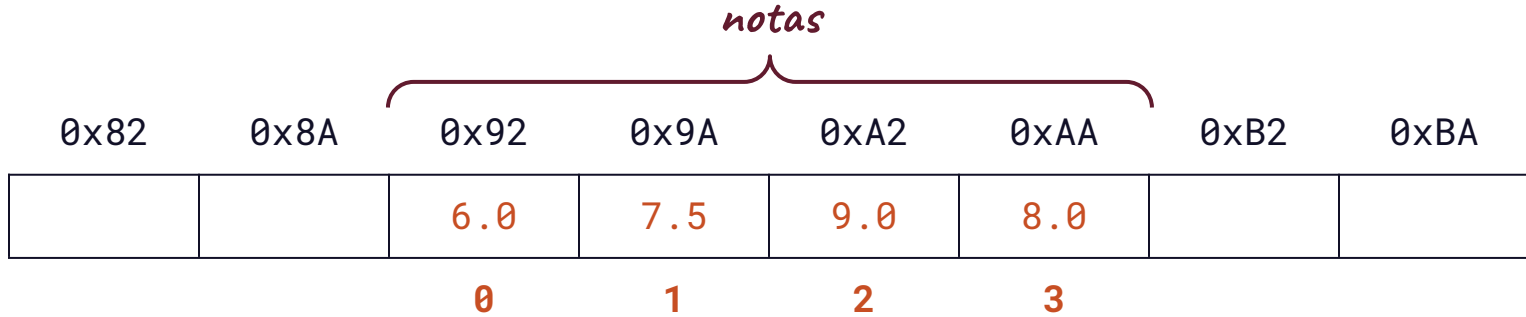
# Arrays em Java



**notas = {6, 7.5, 9, 8}**

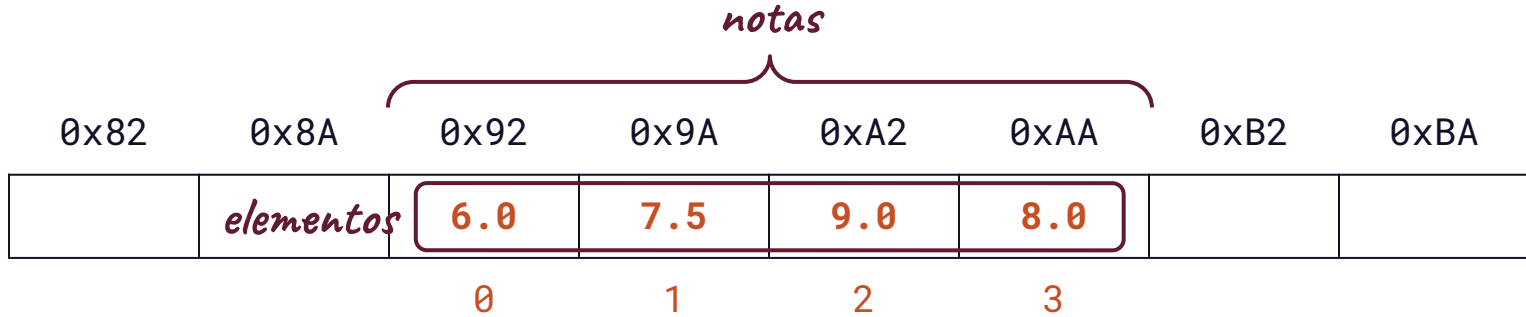


# Arrays em Java



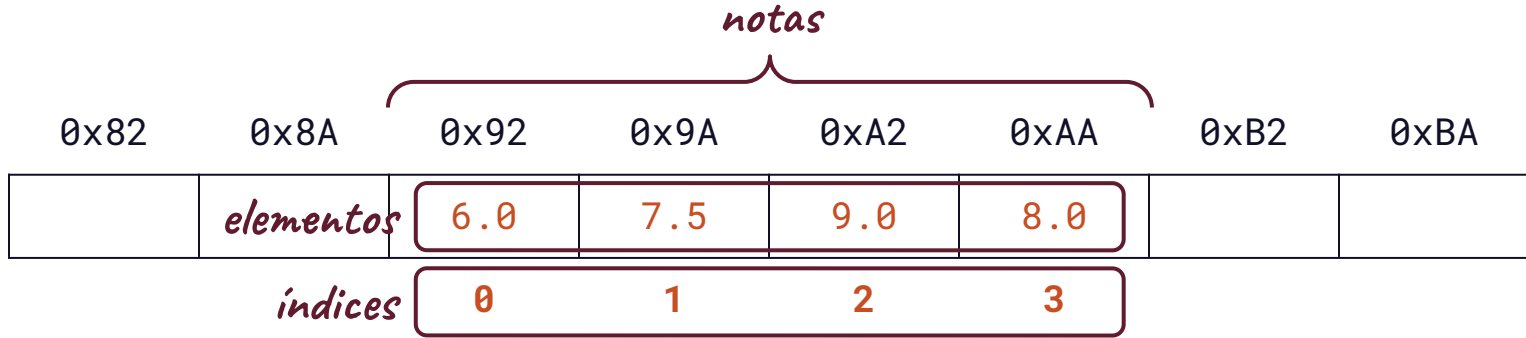
**notas = {6, 7.5, 9, 8}**

# Arrays em Java



**notas = {6, 7.5, 9, 8}**

# Arrays em Java



**notas = {6, 7.5, 9, 8}**

# Definição de Arrays em Java

declaração

exemplos

- sem inicialização



# Definição de Arrays em Java

## declaração

- **sem inicialização**

```
Tipo[] nome;
```

## exemplos



# Definição de Arrays em Java

## declaração

- **sem inicialização**

Tipo[] nome;

## exemplos

```
// Tipo[] nome;  
int[] numeros;  
char[] letras;
```



# Definição de Arrays em Java

## declaração

### ■ sem inicialização

```
Tipo[] nome;
```

```
Tipo nome[];
```

## exemplos

```
// Tipo[] nome;
```

```
int[] numeros;
```

```
char[] letras;
```



# Definição de Arrays em Java

## declaração

### ■ sem inicialização

```
Tipo[] nome;
```

```
Tipo nome[];
```

## exemplos

```
// Tipo[] nome;
```

```
int[] numeros;
```

```
char[] letras;
```

```
// Tipo nome[];
```

```
double medias[];
```

```
String nomes[];
```





# Definição de Arrays em Java

## declaração

- sem inicialização

```
Tipo[] nome;
```

```
Tipo nome[];
```

- com inicialização

## exemplos

```
// Tipo[] nome;
```

```
int[] numeros;
```

```
char[] letras;
```

```
// Tipo nome[];
```

```
double medias[];
```

```
String nomes[];
```



# Definição de Arrays em Java

## declaração

- sem inicialização

```
Tipo[] nome;
```

```
Tipo nome[];
```

- com inicialização

```
Tipo[] nome = {valores};
```

## exemplos

```
// Tipo[] nome;
```

```
int[] numeros;
```

```
char[] letras;
```

```
// Tipo nome[];
```

```
double medias[];
```

```
String nomes[];
```

# Definição de Arrays em Java

## declaração

- sem inicialização

```
Tipo[] nome;
```

```
Tipo nome[];
```

- com inicialização

```
Tipo[] nome = {valores};
```

- valores separados por vírgulas

## exemplos

```
// Tipo[] nome;
```

```
int[] numeros;
```

```
char[] letras;
```

```
// Tipo nome[];
```

```
double medias[];
```

```
String nomes[];
```

# Definição de Arrays em Java

## declaração

- sem inicialização

```
Tipo[] nome;
```

```
Tipo nome[];
```

- com inicialização

```
Tipo[] nome = {valores};
```

- valores separados por vírgulas
- valores entre chaves {}

## exemplos

```
// Tipo[] nome;
```

```
int[] numeros;
```

```
char[] letras;
```

```
// Tipo nome[];
```

```
double medias[];
```

```
String nomes[];
```

# Definição de Arrays em Java

## declaração

### ■ sem inicialização

```
Tipo[] nome;
```

```
Tipo nome[];
```

### ■ com inicialização

```
Tipo[] nome = {valores};
```

- valores separados por vírgulas
- valores entre chaves {}
- valores do mesmo tipo

## exemplos

```
// Tipo[] nome;
```

```
int[] numeros;
```

```
char[] letras;
```

```
// Tipo nome[];
```

```
double medias[];
```

```
String nomes[];
```

# Definição de Arrays em Java

## declaração

### ■ sem inicialização

```
Tipo[] nome;
```

```
Tipo nome[];
```

### ■ com inicialização

```
Tipo[] nome = {valores};
```

- valores separados por vírgulas
- valores entre chaves { }
- valores do mesmo tipo

## exemplos

```
// Tipo[] nome;
```

```
int[] numeros;
```

```
char[] letras;
```

```
// Tipo nome[];
```

```
double medias[];
```

```
String nomes[];
```

```
// Tipo[] nome = {valores};
```

```
int[] idades = {18,17,29,38};
```

```
String[] alunos = {"Carlos", "Ana"};
```

# Definição de Arrays em Java

## declaração

- sem inicialização

```
Tipo[] nome;
```

```
Tipo nome[];
```

- com inicialização

```
Tipo[] nome = {valores};
```

```
Tipo[] nome = new Tipo[tamanho];
```

## exemplos

```
// Tipo[] nome;
```

```
int[] numeros;
```

```
char[] letras;
```

```
// Tipo nome[];
```

```
double medias[];
```

```
String nomes[];
```

```
// Tipo[] nome = {valores};
```

```
int[] idades = {18,17,29,38};
```

```
String[] alunos = {"Carlos", "Ana"};
```



# Definição de Arrays em Java

## declaração

### ■ sem inicialização

```
Tipo[] nome;
```

```
Tipo nome[];
```

### ■ com inicialização

```
Tipo[] nome = {valores};
```

```
Tipo[] nome = new Tipo[tamanho];
```

- o tamanho deve ser inteiro

## exemplos

```
// Tipo[] nome;
```

```
int[] numeros;
```

```
char[] letras;
```

```
// Tipo nome[];
```

```
double medias[];
```

```
String nomes[];
```

```
// Tipo[] nome = {valores};
```

```
int[] idades = {18,17,29,38};
```

```
String[] alunos = {"Carlos", "Ana"};
```



# Definição de Arrays em Java

## declaração

- sem inicialização

```
Tipo[] nome;
```

```
Tipo nome[];
```

- com inicialização

```
Tipo[] nome = {valores};
```

```
Tipo[] nome = new Tipo[tamanho];
```

- o tamanho deve ser inteiro
- o tamanho define a capacidade do array

## exemplos

```
// Tipo[] nome;
```

```
int[] numeros;
```

```
char[] letras;
```

```
// Tipo nome[];
```

```
double medias[];
```

```
String nomes[];
```

```
// Tipo[] nome = {valores};
```

```
int[] idades = {18,17,29,38};
```

```
String[] alunos = {"Carlos", "Ana"};
```

# Definição de Arrays em Java

## declaração

- sem inicialização

```
Tipo[] nome;
```

```
Tipo nome[];
```

- com inicialização

```
Tipo[] nome = {valores};
```

```
Tipo[] nome = new Tipo[tamanho];
```

- o tamanho deve ser inteiro
- o tamanho define a capacidade do array
- uma vez definido, o tamanho não muda

## exemplos

```
// Tipo[] nome;
```

```
int[] numeros;
```

```
char[] letras;
```

```
// Tipo nome[];
```

```
double medias[];
```

```
String nomes[];
```

```
// Tipo[] nome = {valores};
```

```
int[] idades = {18,17,29,38};
```

```
String[] alunos = {"Carlos", "Ana"};
```



# Definição de Arrays em Java

## declaração

- sem inicialização

```
Tipo[] nome;
```

```
Tipo nome[];
```

- com inicialização

```
Tipo[] nome = {valores};
```

```
Tipo[] nome = new Tipo[tamanho];
```

- o tamanho deve ser inteiro
- o tamanho define a capacidade do array
- uma vez definido, o tamanho não muda

## exemplos

```
// Tipo[] nome;
```

```
int[] numeros;
```

```
char[] letras;
```

```
// Tipo nome[];
```

```
double medias[];
```

```
String nomes[];
```

```
// Tipo[] nome = {valores};
```

```
int[] idades = {18,17,29,38};
```

```
String[] alunos = {"Carlos", "Ana"};
```

```
// Tipo[] nome = new Tipo[tamanho];
```

```
double[] temperaturas = new double[5];
```

```
char[] simbolos = new char[11];
```

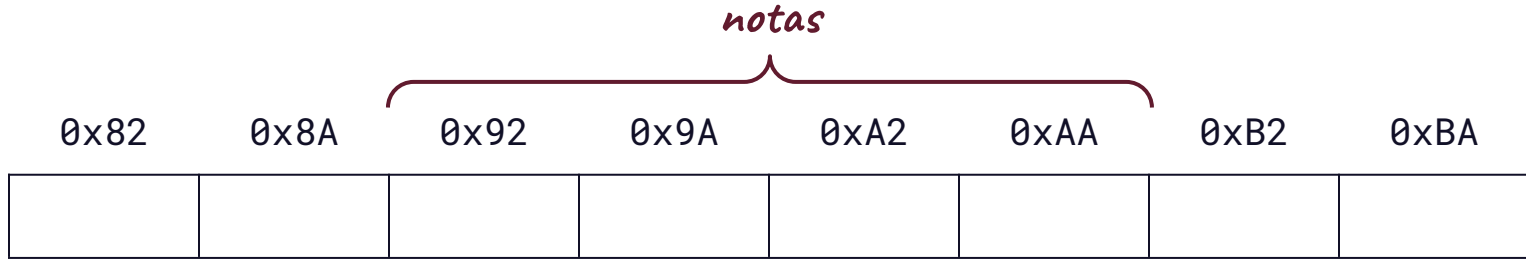


# Arrays em Java

0x82	0x8A	0x92	0x9A	0xA2	0xAA	0xB2	0xBA

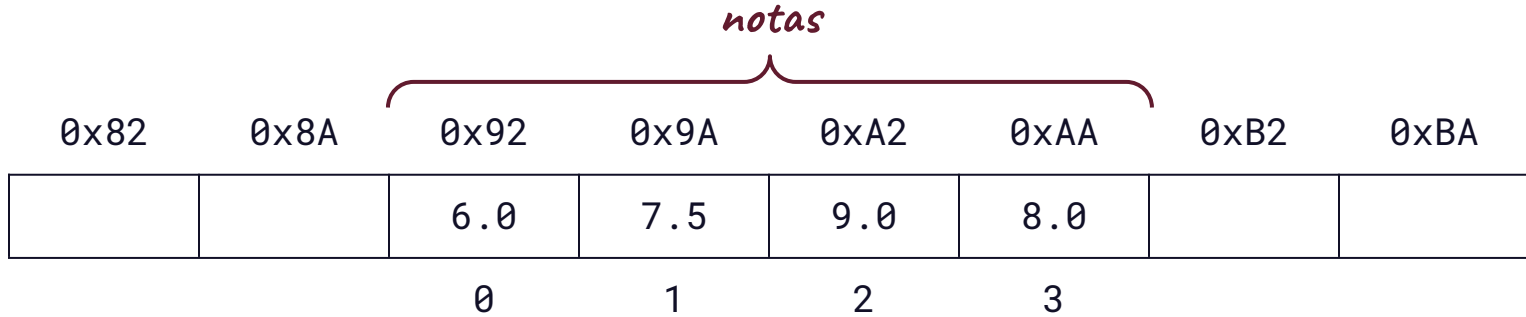
```
double[] notas = {6, 7.5, 9, 8};
```

# Arrays em Java



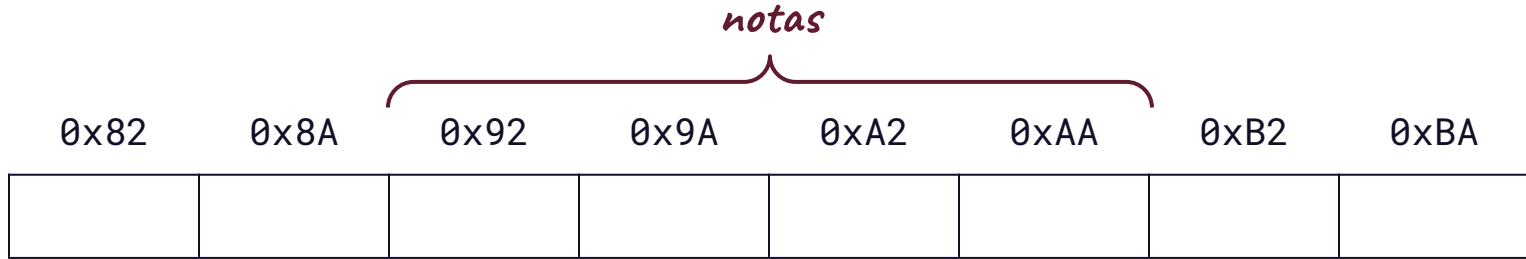
```
double[] notas = {6, 7.5, 9, 8};
```

# Arrays em Java



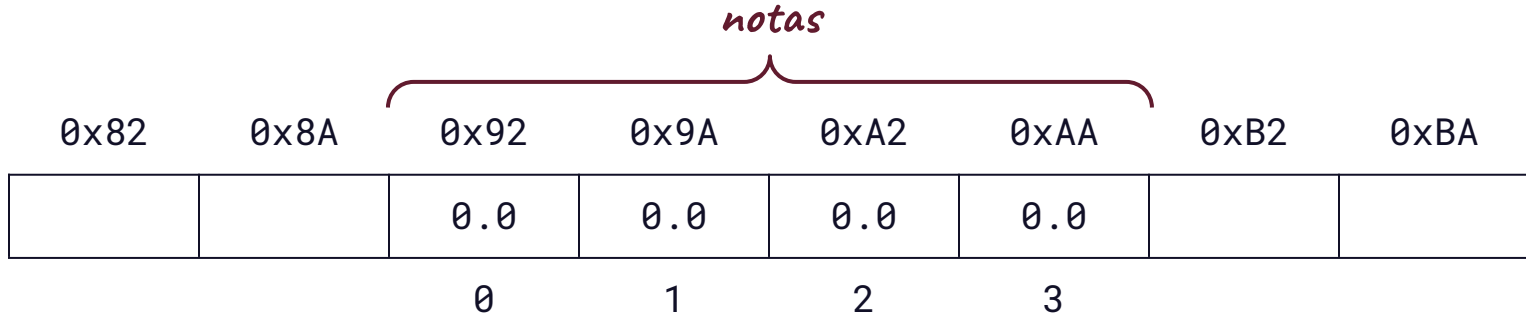
```
double[] notas = {6, 7.5, 9, 8};
```

# Arrays em Java



```
double[] notas = new double[4];
```

# Arrays em Java



```
double[] notas = new double[4];
```



# Acessando os Elementos do Array

*notas*

0x82	0x8A	0x92	0x9A	0xA2	0xAA	0xB2	0xBA
		6.0	7.5	9.0	8.0		
		0	1	2	3		

# Acessando os Elementos do Array

*notas*

0x82	0x8A	0x92	0x9A	0xA2	0xAA	0xB2	0xBA
		6.0	7.5	9.0	8.0		
		0	1	2	3		

**nome[índice]**

# Acessando os Elementos do Array

*notas*

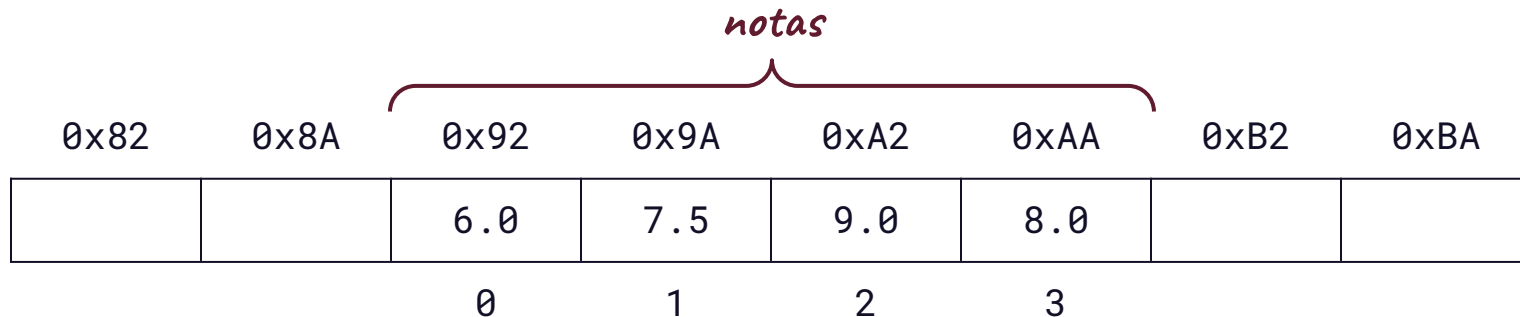
0x82	0x8A	0x92	0x9A	0xA2	0xAA	0xB2	0xBA
		6.0	7.5	9.0	8.0		
		0	1	2	3		

`nome[índice]`

**lendo um elemento**



# Acessando os Elementos do Array

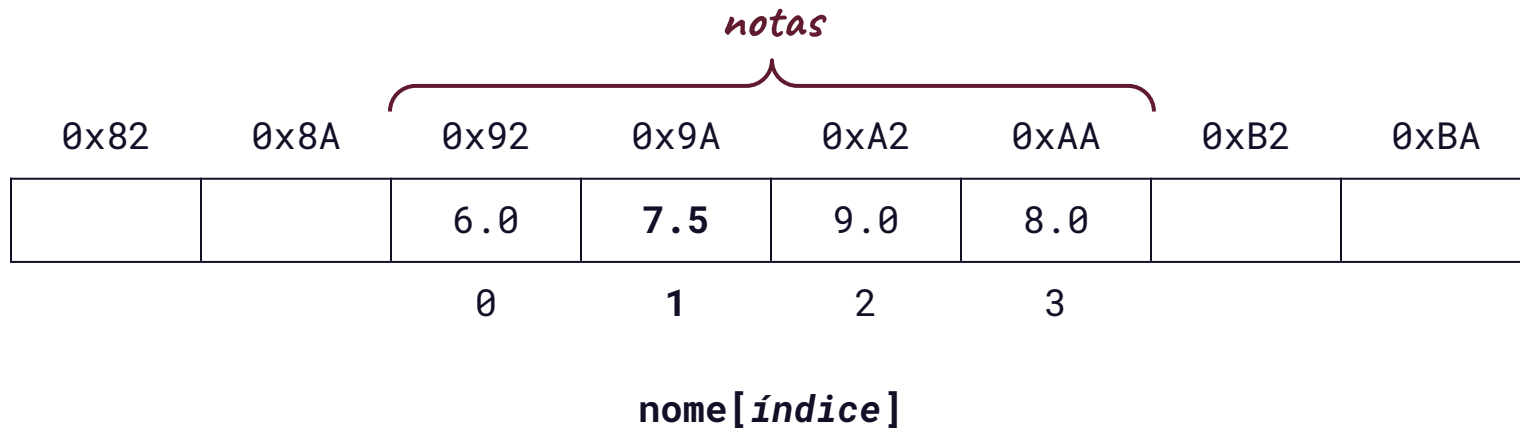


`nome[índice]`

## lendo um elemento

```
System.out.println(notas[1]);
```

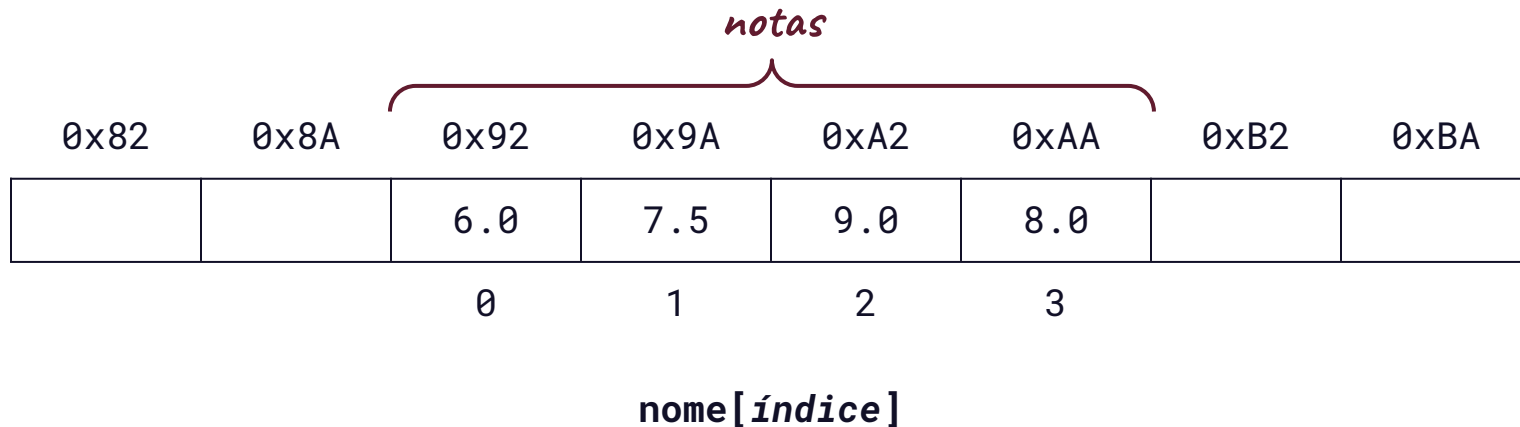
# Acessando os Elementos do Array



## lendo um elemento

```
System.out.println(notas[1]); // 7.5
```

# Acessando os Elementos do Array



## lendo um elemento

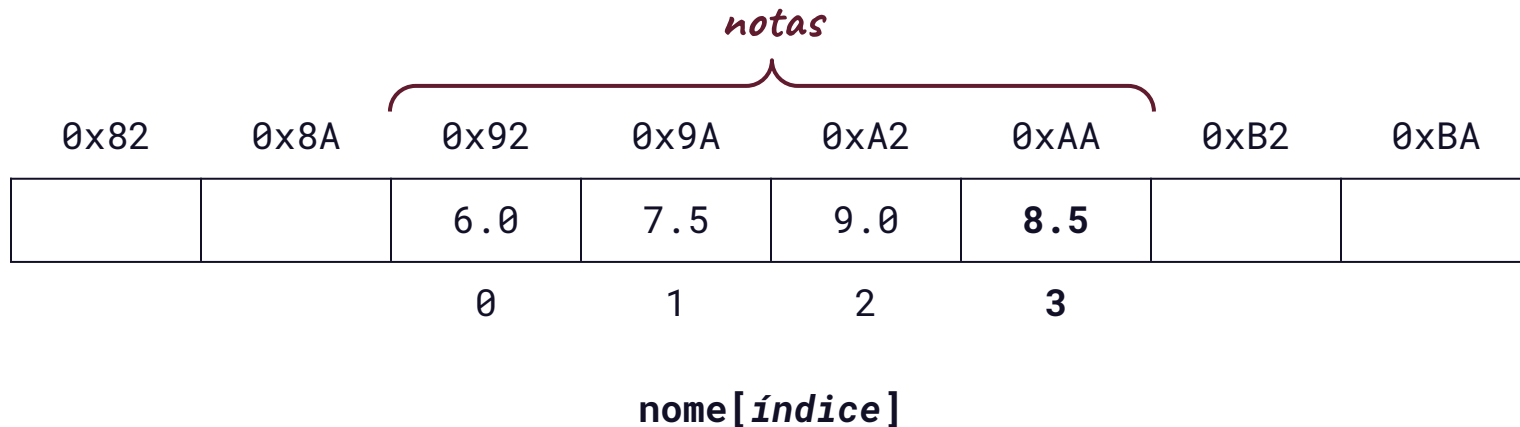
```
System.out.println(notas[1]); // 7.5
```

## escrevendo um elemento

```
notas[3] = 8.5;
```



# Acessando os Elementos do Array



## lendo um elemento

```
System.out.println(notas[1]); // 7.5
```

## escrevendo um elemento

```
notas[3] = 8.5;
```



# Programando a Solução

calcular média

$$média = \frac{número_1 + ... + número_n}{n}$$

pseudo-código

entrada: *coleção*

saída: *média*

Início

1. *soma, contador*  $\leftarrow$  0
2. Para cada *número* da *coleção*:
3.     *soma*  $\leftarrow$  *soma* + *número*
4.     *contador*  $\leftarrow$  *contador* + 1
5. *média*  $\leftarrow$  *soma* / *contador*
6. Retorna *média*

Fim





# Programando a Solução

código Java

pseudo-código

entrada: *coleção*

saída: *média*

Início

1. *soma, contador*  $\leftarrow$  0
2. Para cada *número* da *coleção*:
3.     *soma*  $\leftarrow$  *soma* + *número*
4.     *contador*  $\leftarrow$  *contador* + 1
5. *média*  $\leftarrow$  *soma* / *contador*
6. Retorna *média*

Fim



# Programando a Solução

## código Java

```
double[] notas = {6, 7.5, 9, 8};  
double media = 0;
```

```
media = notas[0] + notas[1];  
media = media + notas[2];  
media = media + notas[3];
```

```
media = media/4;
```

```
System.out.println(media);
```

## pseudo-código

entrada: *coleção*  
saída: *média*

Início

1. *soma, contador*  $\leftarrow$  0
2. Para cada *número* da *coleção*:
3.     *soma*  $\leftarrow$  *soma* + *número*
4.     *contador*  $\leftarrow$  *contador* + 1
5. *média*  $\leftarrow$  *soma* / *contador*
6. Retorna *média*

Fim



# Acessando os Elementos do Array

*notas*

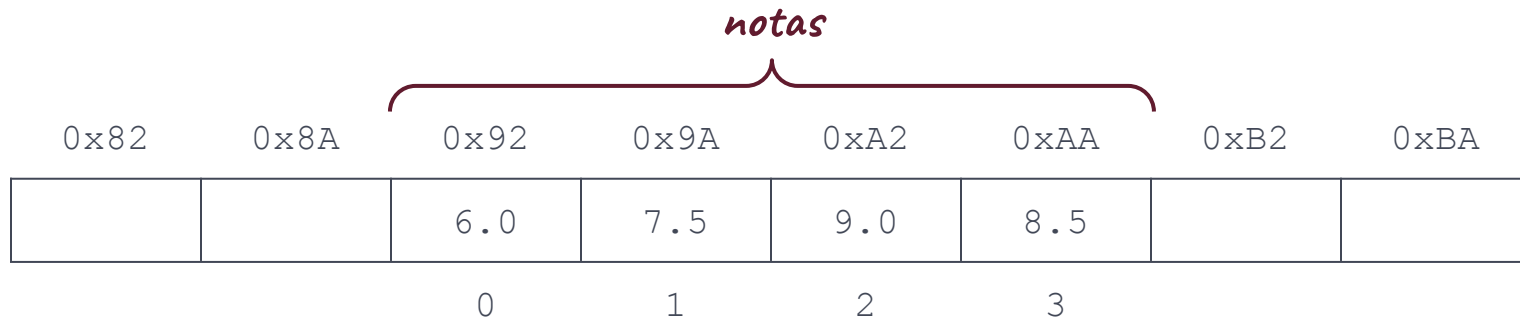
0x82	0x8A	0x92	0x9A	0xA2	0xAA	0xB2	0xBA
		6.0	7.5	9.0	8.5		
		0	1	2	3		

**nome[índice]**

**lendo todos os elementos**



# Acessando os Elementos do Array

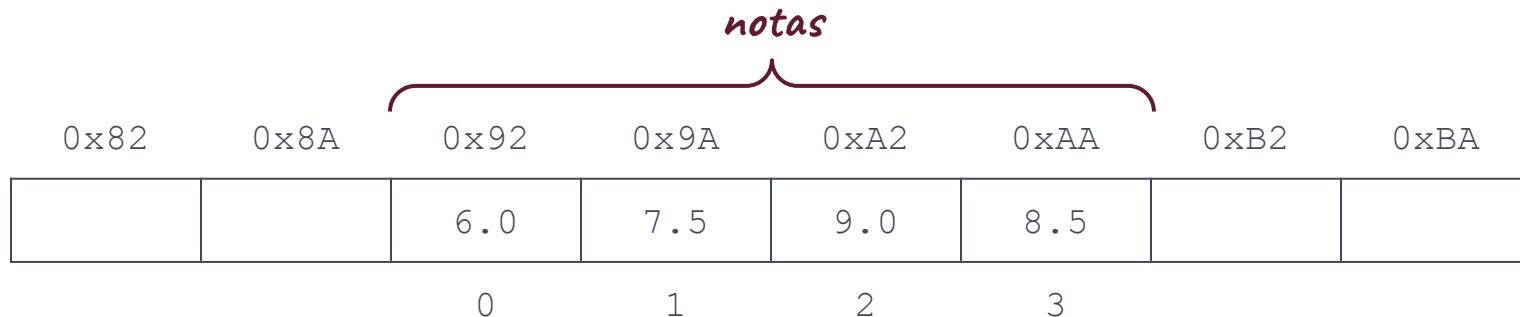


**nome[índice]**

## lendo todos os elementos

```
for(int i = 0; i < 4; i++) {  
    System.out.println(notas[i]);  
}
```

# Acessando os Elementos do Array



*início*

nome[índice]

**lendo todos os elementos**

```
for(int i = 0; i < 4; i++) {  
    System.out.println(notas[i]);  
}
```

# Acessando os Elementos do Array

*notas*

0x82	0x8A	0x92	0x9A	0xA2	0xAA	0xB2	0xBA
		6.0	7.5	9.0	8.5		
		0	1	2	3		

*início* *fim*

**nome[índice]**

**lendo todos os elementos**

```
for(int i = 0; i < 4; i++) {  
    System.out.println(notas[i]);  
}
```

# Programando a Solução

código Java

pseudo-código

entrada: *coleção*

saída: *média*

Início

1.  $soma, contador \leftarrow 0$
2. Para cada *número* da *coleção*:
3.      $soma \leftarrow soma + número$
4.      $contador \leftarrow contador + 1$
5.  $média \leftarrow soma / contador$
6. Retorna *média*

Fim



# Programando a Solução

## código Java

```
double[] notas = {6, 7.5, 9, 8};  
double media = 0;
```

```
for(int i = 0; i < 4; i++) {  
    media = media + notas[i];  
}
```

```
media = media/4;
```

```
System.out.println(media);
```

## pseudo-código

entrada: *coleção*

saída: *média*

Início

1. *soma, contador*  $\leftarrow$  0
2. Para cada *número* da *coleção*:
3.     *soma*  $\leftarrow$  *soma* + *número*
4.     *contador*  $\leftarrow$  *contador* + 1
5. *média*  $\leftarrow$  *soma* / *contador*
6. Retorna *média*

Fim





# Comprimento de Arrays

**length**

# Comprimento de Arrays

**length**

- do inglês: "comprimento"



# Comprimento de Arrays

## length

- do inglês: "comprimento"
- **todo array tem a propriedade length**



# Comprimento de Arrays

## length

- do inglês: "comprimento"
- todo array tem a propriedade `length`
- **`length` retorna um número inteiro**



# Comprimento de Arrays

## length

- do inglês: "comprimento"
- todo array tem a propriedade `length`
- `length` retorna um número inteiro
- **`length` é sempre maior do que qualquer índice**



# Comprimento de Arrays

## length

- do inglês: "comprimento"
- todo array tem a propriedade length
- length retorna um número inteiro
- length é sempre maior do que qualquer índice

## exemplos

```
double[] notas = {6, 7.5, 9, 8};  
System.out.println(notas.length); // 4
```



# Comprimento de Arrays

## length

- do inglês: "comprimento"
- todo array tem a propriedade length
- length retorna um número inteiro
- length é sempre maior do que qualquer índice

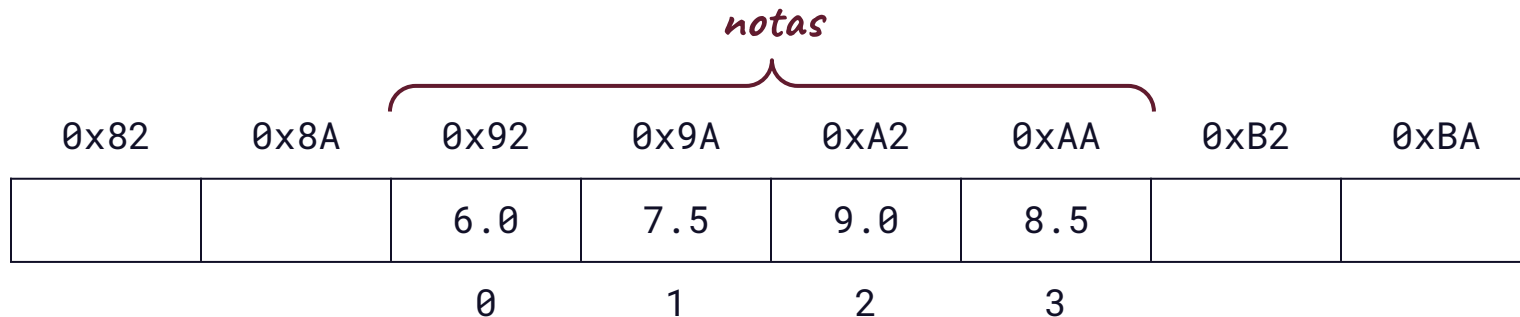
## exemplos

```
double[] notas = {6, 7.5, 9, 8};  
System.out.println(notas.length); // 4
```

```
int[] vazio = new int[1000];  
System.out.println(vazio.length); // 1000
```



# Acessando os Elementos do Array



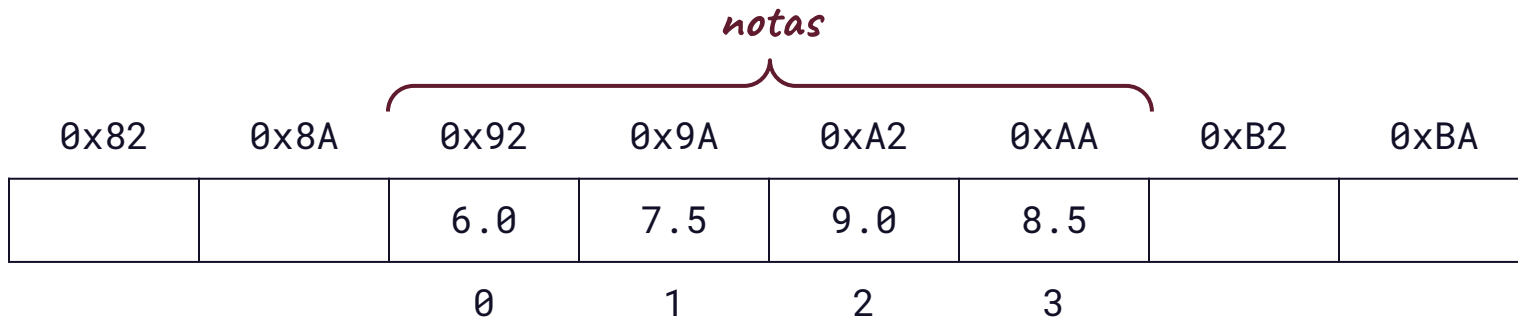
**nome[índice]**

## lendo todos os elementos

```
for(int i = 0; i < 4; i++) {  
    System.out.println(notas[i]);  
}
```



# Acessando os Elementos do Array

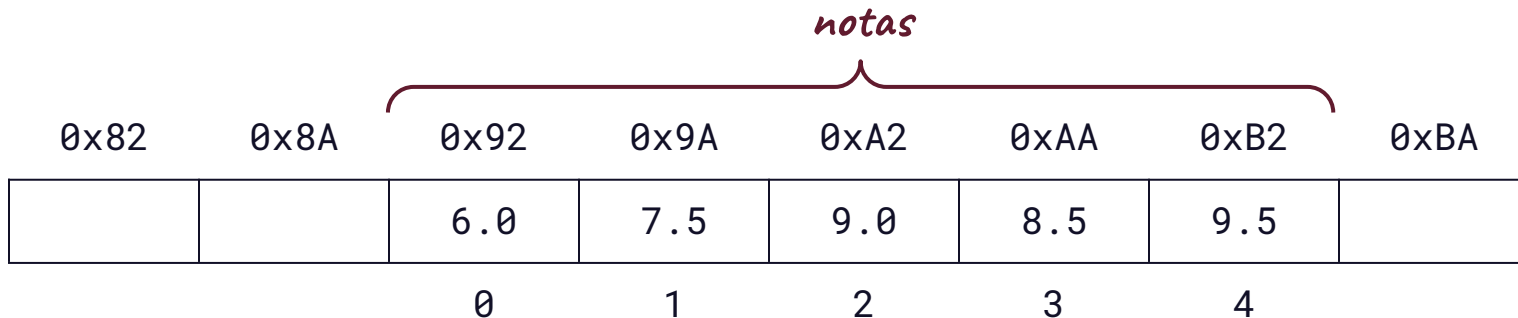


**nome[índice]**

## lendo todos os elementos

```
for(int i = 0; i < notas.length; i++) {  
    System.out.println(notas[i]);  
}
```

# Acessando os Elementos do Array



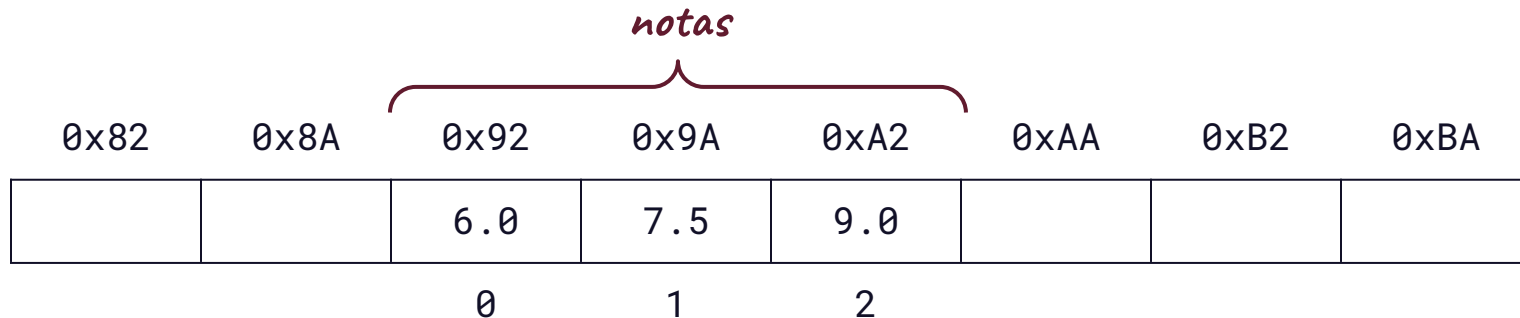
`nome[índice]`

## lendo todos os elementos

```
for(int i = 0; i < notas.length; i++) {  
    System.out.println(notas[i]);  
}
```



# Acessando os Elementos do Array



`nome[índice]`

## lendo todos os elementos

```
for(int i = 0; i < notas.length; i++) {  
    System.out.println(notas[i]);  
}
```



# Programando a Solução

## código Java

```
double[] notas = {6, 7.5, 9, 8};  
double media = 0;
```

```
for(int i = 0; i < 4; i++) {  
    media = media + notas[i];  
}
```

```
media = media/4;
```

```
System.out.println(media);
```

## pseudo-código

entrada: *coleção*

saída: *média*

Início

1. *soma, contador*  $\leftarrow$  0
2. Para cada *número* da *coleção*:
3.     *soma*  $\leftarrow$  *soma* + *número*
4.     *contador*  $\leftarrow$  *contador* + 1
5. *média*  $\leftarrow$  *soma* / *contador*
6. Retorna *média*

Fim



# Programando a Solução

## código Java

```
double[] notas = {6, 7.5, 9, 8};  
double media = 0;  
  
for(int i = 0; i < notas.length; i++) {  
    media = media + notas[i];  
}  
  
media = media/4;  
  
System.out.println(media);
```

## pseudo-código

entrada: *coleção*  
saída: *média*

Início

1. *soma, contador*  $\leftarrow$  0
2. Para cada *número* da *coleção*:
3.     *soma*  $\leftarrow$  *soma* + *número*
4.     *contador*  $\leftarrow$  *contador* + 1
5. *média*  $\leftarrow$  *soma* / *contador*
6. Retorna *média*

Fim



# Programando a Solução

## código Java

```
double[] notas = {6, 7.5, 9, 8};  
double media = 0;  
  
for(int i = 0; i < notas.length; i++) {  
    media = media + notas[i];  
}  
  
media = media/notas.length;  
  
System.out.println(media);
```

## pseudo-código

entrada: *coleção*  
saída: *média*

Início

1. *soma, contador*  $\leftarrow$  0
2. Para cada *número* da *coleção*:
3.     *soma*  $\leftarrow$  *soma* + *número*
4.     *contador*  $\leftarrow$  *contador* + 1
5. *média*  $\leftarrow$  *soma* / *contador*
6. Retorna *média*

Fim



# Vantagens do Uso

- armazenamento eficiente



# Vantagens do Uso



- armazenamento eficiente
- simplicidade



# Vantagens do Uso



- armazenamento eficiente
- simplicidade
- **acesso aleatório**

# Vantagens do Uso



- armazenamento eficiente
- simplicidade
- acesso aleatório
- **fundação para estruturas de dados**

# Vantagens do Uso



- armazenamento eficiente
- simplicidade
- acesso aleatório
- fundação para estruturas de dados

# Exercícios Práticos

**1. Implemente um algoritmo que receba dois arrays, sendo que o primeiro é uma lista de nomes e o segundo é uma lista de notas, e descubra quem é o aluno com a maior nota.**

# Exercícios Práticos

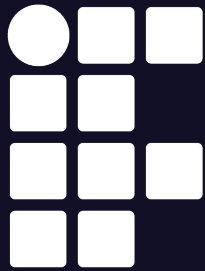
1. Implemente um algoritmo que receba dois arrays, sendo que o primeiro é uma lista de nomes e o segundo é uma lista de notas, e descubra quem é o aluno com a maior nota.
2. Implemente um algoritmo que receba um array de números inteiros, inverta a ordem dos seus elementos e imprima o array resultante.



# Exercícios Práticos

1. Implemente um algoritmo que receba dois arrays, sendo que o primeiro é uma lista de nomes e o segundo é uma lista de notas, e descubra quem é o aluno com a maior nota.
2. Implemente um algoritmo que receba um array de números inteiros, inverta a ordem dos seus elementos e imprima o array resultante.
3. Implemente um algoritmo que una dois arrays de inteiros em um único array e exiba o array resultante.





**INSTITUTO  
FEDERAL**  
Santa Catarina

## **Arrays em Java**

### **Pensamento Computacional e Algoritmos**

Análise e Desenvolvimento de Sistemas

Prof. Adriano Lima

[adriano.lima@ifsc.edu.br](mailto:adriano.lima@ifsc.edu.br)

