



Revisão Algoritmo JAVA

Java

- » linguagem de programação de propósito genérico inicialmente desenvolvida pela Sun, atualmente pela Oracle
- » suporta tipos estáticos, classes, objetos e concorrência
- » compilada para *bytecode* por uma máquina virtual (*"write-once, run anywhere"*)
- » sintaxe baseada no C/C++
- » mais alto nível:
 - > modelo de objetos mais simples que o de C++
 - > libertação automática de memória (*garbage collection*)
 - > **não suporta aritmética de apontadores**
 - > tratamento de erros de execução usando *exceções*.



Plataforma Java

- » Java Standard Edition (JSE):
 - > compilador: javac
 - > interpretador JVM: java
 - > conjunto de bibliotecas padrão
 - > ferramentas (jar, javadoc, etc.)
- » Java Enterprise Edition (JEE):
 - > JSE + componentes extra para aplicações empresariais.



Evolução da linguagem

- » 1996 JDK 1.0 primeira versão
- » 2000 J2SE 1.3 HotspotVM
- » 2005 J2SE 5 tipos genéricos, enumerações, ciclos for implícitos
- » 2011 JSE 7 melhorias na inferência de tipos genéricos
- » 2014 JSE 8 expressões lambda, streams funcionais
- » 2017 JSE 9 avançando na modularização da linguagem
- » 2018 JSE 10 introduzindo o novo sistema de versionamento da linguagem



Princípios da linguagem

- » combina programação imperativa com objetos
- » distingue entre valores básicos e objetos
 - > objetos são usados para estruturar programas
 - > valores básicos não são objetos
- » execução em máquina virtual (facilita portabilidade)
- » gestão automática de memória:
 - > a alocação de objetos é explícita
 - > a libertação é implícita (garbage collection)



Arquivos Java

- » O nome do arquivo em uma classe pública deve ser igual ao nome da classe.

// num arquivo Rectangle.java

```
public class Rectangle {
```

```
    ...
```

```
}
```

- » Pode ter outras classes desde que não sejam públicas.



Método *main*

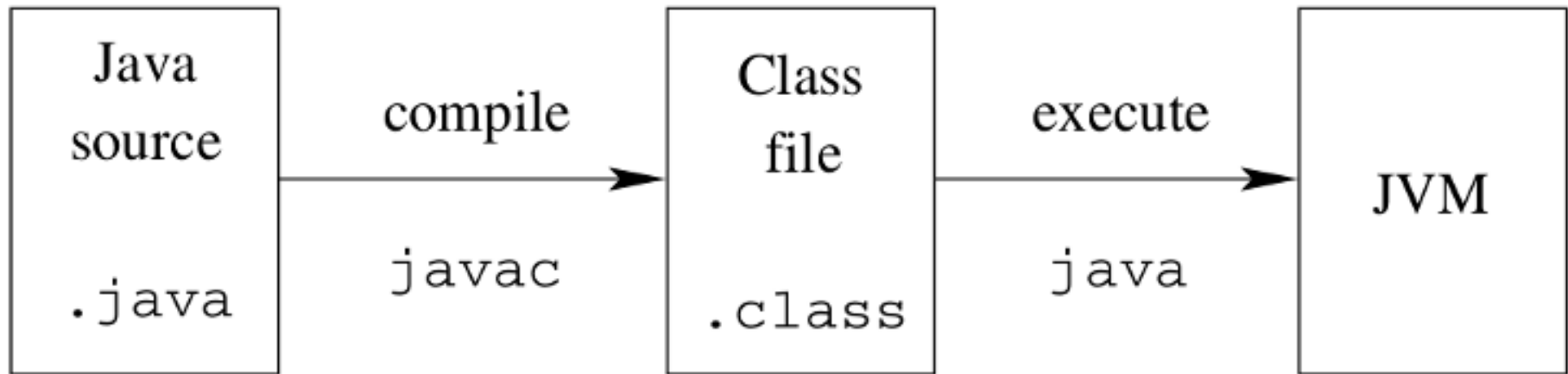
- » programa completo deve ter uma classe com um método estático *main*
- » os argumentos de linha de comando são passados como um *array* de String

// num ficheiro Hello.java

```
class Hello {  
    public static void main(String args[]) {  
        System.out.println("Hello world!");  
    }  
}
```



Compilação e Execução



Vetores

- » Um vetor é uma lista de objetos similares.
- » Um vetor possui características que são fixas:
 - > nome
 - > tipo
 - > tamanho
- » Estas características precisam ser declaradas quando o vetor é definido.
- » O Tamanho do vetor não pode ser alterado durante a execução do programa.



Vetores

- » Vetores são comumente denominados de arrays no Java.
- » Vetores são arrays unidimensionais.
- » Em computação um **Vetor**(Array) ou **Arranjo** é o nome de uma matriz unidimensional considerada a mais simples das estruturas de dados.
- » Constituída por dados do mesmo tipo (homogêneos) e tamanho que são agrupados continuamente na memória e acessados por sua posição (índice -geralmente um número inteiro) dentro do vetor.



meuVetor =

3	6	3	1	6	3	4	1
Índices -> 0	1	2	3	4	5	6	7

A vetor definido acima possui espaço para 8 elementos:

- Os elementos são acessados por seus índices.
- No Java, o vetor inicia pelo índice de valor 0.



Declarando Vetores

```
int meuVetor[];
```

> Declara *meuVetor* como um array de inteiros

```
meuVetor = new int[8];
```

> Reserva um espaço na memória suficiente para armazenar 8 inteiros, que serão acessados como *meuVetor[0]* a *meuVetor[7]*

> As duas setenças acima podem ser combinadas em uma única linha:

```
int meuVetor[] = new int[8];
```



Declarando Vetores

- Declaração e expressão de criação de arrays de 12 elementos int

```
int[] c = new int[ 12 ];
```

- Pode ser realizado em duas etapas como segue:

```
int[] c; // declara a variável de array  
c = new int[ 12 ]; // cria o array
```



Erro comum de programação 7.2

Em uma declaração de array, especificar o número de elementos entre os colchetes da declaração (por exemplo, `int[12] c;`) é um erro de sintaxe.



Observações

- » Quando o tipo do elemento e os colchetes são combinados no início da declaração, todos os identificadores na declaração são variáveis de array.
- » Para legibilidade, declare apenas uma variável por declaração.



Boa prática de programação 7.1

Para legibilidade, declare apenas uma variável por declaração. Mantenha cada declaração em uma linha separada e inclua um comentário que descreva a variável sendo declarada.



Observações



Erro comum de programação 7.3

Declarar múltiplas variáveis de array em uma única declaração pode levar a erros sutis. Considere a declaração `int [] a, b, c;`. Se `a`, `b` e `c` devem ser declarados como variáveis de array, então essa declaração é correta — colocar os colchetes logo depois do tipo indica que todos os identificadores na declaração são variáveis de array. Entretanto, se apenas `a` destina-se a ser uma variável de array, e `b` e `c` variáveis `int` individuais, então essa declaração é incorreta — a declaração `int a[], b, c;` alcançaria o resultado desejado.



Atribuindo Valores

- » Para armazenar valores no vetor, indique o índice referente a posição de armazenamento desejada:

```
meuVetor[0] = 3;
```

```
meuVetor[1] = 6;
```

```
meuVetor[2] = 3;
```

...

- » Pode-se inicializar um vetor em uma única sentença:

```
int meuVetor[] = {3, 6, 3, 1, 6, 3, 4, 1};
```



Iteração Através dos Vetores

» A estrutura de repetição *for* é muito usada para manipular vetores:

```
for (int i = 0; i < meuVetor.length; i++) {  
    meuVetor[i] = 0;  
} // inicializa todos os elementos  
// do vetor com o valor zero.
```



Observações

- » Um índice deve ser um inteiro não negativo.
- » Um programa pode utilizar uma expressão como um índice.
- » Quando um array é criado, cada elemento do array recebe um valor padrão: Zero para os elementos numéricos de tipo primitivo, false para elementos boolean e null para referências?



Erro comum de programação 7.1

Um índice precisa ser um valor int ou um valor de um tipo que possa ser promovido a int — a saber, byte, short ou char, mas não long; caso contrário, ocorrerá um erro de compilação.

Lista de exercício de Vetores

- » 1) Solicite ao usuário 5 números inteiros, guarde-os em um vetor e em seguida imprima-os na tela. Em seguida conte quantos elementos são negativos e informe ao usuário.
- » 2) Leia um vetor de 10 elementos e em seguida ache a posição do elemento m (dado pelo usuário) no vetor. Caso o elemento não exista no vetor informe ao usuário.



A estrutura for aprimorada

- **Instrução for aprimorada**

Itera pelos elementos de um array sem utilizar um contador.


Evita a possibilidade de ultrapassar o limite do array.

Também funciona com as coleções predefinidas da Java API (veja a Seção 7.14).

- **Sintaxe:**

```
for ( parâmetro : nomeDoArray )  
    instrução
```

onde *parâmetro* tem um tipo e um identificador e *arrayName* é o array pelo qual iterar.

- O tipo do parâmetro deve ser consistente com o tipo de elemento no array.
 - A instrução for aprimorada simplifica o código para iterar por um array.
- 

A estrutura for aprimorada

```
1 // Figura 7.12: EnhancedForTest.java
2 // Usando instrução for aprimorada para somar inteiros em um array.
3
4 public class EnhancedForTest
5 {
6     public static void main( String[] args )
7     {
8         int[] array = { 87, 68, 94, 100, 83, 78, 85, 91, 76, 87 };
9         int total = 0;
10
11         // adiciona o valor de cada elemento ao total
12         for ( int number : array )
13             total += number;
14
15         System.out.printf( "Total of array elements: %d\n", total );
16     } // fim do main
17 } // fim da classe EnhancedForTest
```

Para cada elemento em array, atribui o valor do elemento a `number`; então, adiciona `number` a `total`

Total of array elements: 849

Figura 7.12 | Utilizando a instrução for aprimorada para somar inteiros em um array.

A instrução for aprimorada simplifica o código para iterar por um array.

A estrutura for aprimorada

- A instrução **for** aprimorada somente pode ser utilizada para obter elementos do array
- Ela não pode ser utilizada para modificar elementos em um array.
- Para modificar elementos, utilize a instrução for tradicional.
- Pode ser utilizada no lugar da instrução for loop controlado por contador se você não precisar acessar o índice do elemento.

