Linguagem de Programação

Java

Linguagem de Programação JAVA
Conteúdo

- Conteúdo
  - Apresentação
  - Definição das atividades
  - Formas de Avaliação
  - Regras de Horários
  - Tipos de Atividades
  - Apresentação do Plano de Ensino;
  - 4Learn – Ambiente Virtual de Aprendizagem

- Apresentação do Professor
  - Tecnólogo em Processamento de Dados
  - Mestre em Educação
  - INPI (básico e avançado)

- Metodologia de execução das atividades

- Formas de Avaliação
  - duas provas – 4.0
  - C.H.A – 1.0 (lista de exercícios, comprometimento, frequência, Facebook, kkk)
  - Atividades desafio – 1.0

- Apresentação do Plano de Ensino

- 4Learn – Ambiente Virtual de Aprendizagem à Distância
  - www.4learn.pro.br
  - Login e Senha
  - Simulação de utilização

Linguagem de Programação JAVA
definições iniciais

- A linguagem Java
  - Case sensitive
    - umaVariavel É DIFERENTE DE umavariavel (aplicado para nome de classes, objetos, métodos, variáveis e constantes.
    - O uso incorreto de algum nome gerará um erro de compilação
  - Blocos de códigos são colocados entre chaves { } representando o begin e o end de outras linguagens.
  - No **final** de cada **instrução** é obrigatório o uso do **ponto** e **vírgula** ";"
  - A classe deverá ser salva em formato texto em um arquivo com o mesmo nome da classe com extensão .java, ou seja, OlaMundo.java respeitando maiúsculas e minúsculas.
  - Todo programa em Java é representado por uma ou mais classes.
  - Normalmente trabalhamos com apenas uma classe por arquivo.

Linguagem de Programação JAVA

definições iniciais

- Declaração de classes, variáveis, atributos e métodos.

A seguinte regra deve ser respeitada na declaração de identificadores de classes, variáveis, atributos e métodos:

- Devem começar com letras de A - Z, a - z, _ ou $
- Os próximos caracteres podem ser qualquer um dos acima mencionados e também números. Exemplo:

Cliente, PESSOA_FISICA, $MinhaVariavel, contador, Classe1, _variavel

Linguagem de Programação JAVA
definições iniciais

- Declaração de uma classes

A declaração de uma classe é feita utilizando-se a palavra reservada class seguida do nome da classe, como podemos ver no exemplo:

```
class OlaMundo {
    // variáveis
    // atributos
    // métodos
}
```

Linguagem de Programação JAVA
definições iniciais

- Palavras reservadas

As palavras reservadas não podem ser utilizadas como identificadores, porém, nem todas elas são utilizadas na linguagem.

Como utilizaremos o programa JCreator para editar os programas desenvolvidos em Java, este avisará o usuário quando da utilização de alguma palavra reservada, colocando-a em destaque (azul).

Linguagem de Programação JAVA
definições iniciais

- Convenções do código
  - Nome de Classe: O primeiro caracter de todas as palavras que o compõem devem iniciar-se com maiúsculo e os demais caracteres devem ser minúsculos. Exemplo:
    - HelloWord, MeuProgramaEmJava, BancoDeDados
  - Métodos, atributos e variáveis: Primeiro caracter minúsculo; Demais palavras seguem a regra de nome de classes, ou seja, devem ter o primeiro caracter maúsculo. Exemplo:
    - minhaFuncao, minhaVariávelInt
  - Constantes: Todos os caracteres maiúsculos e divisão das palavras utilizando undescore "_". Exemplo:
    - MAIUSCULO, DATA_NASCIMENTO

Linguagem de Programação JAVA
definições iniciais

- Convenções do código
  - Identação: Tabulações (com 4 espaços) devem ser abertas após a chave "{" e retroceder após o fechamento da chave "}"

Linguagem de Programação JAVA
definições iniciais

- Convenções do código
  - Comentários: Servem para realizar comentários durante o desenvolvimento dos programas. Podem ser para uma linha ou para múltiplas linhas.
    - Comentários para uma linha: representados pelas barras paralelas "//"
      - Exemplo: // Esta linha calcula a área
    - Comentários de múltiplas linhas: representados por "/*" e "*/". Inicia-se colocando /* e, após o bloco de comentário, finaliza-se com */.
      - Exemplo:

        /* múltiplas linhas para os comentários, utilizado para textos longos
        */

Linguagem de Programação JAVA

definições iniciais

- Variáveis

Representam um espaço de memória para armazenar um valor. Para cada área de memória associamos um nome (identificador) e o tipo de valor a ser armazenado.

As variáveis são classificadas em:

- Tipo primitivo;
- Tipo reference;
- Arrays.

Linguagem de Programação JAVA
definições iniciais

- Variáveis – Tipos primitivos

Sintaxe:

<tipoVariavel> <nomeVariável> = valor;

Os tipo primitivos podem ser:

- Numérico
- Caracter
- Booleanos (verdadeiro ou falso)

Linguagem de Programação JAVA
definições iniciais

- Variáveis – Tipos primitivos - Inteiros

Tipo: byte
Valor Mínimo: -128
Valor Máximo: 127
Bytes consumidos: 1

Tipo: short
Valor Mínimo: -32.768
Valor Máximo: 32.767
Bytes consumidos: 2

Linguagem de Programação JAVA

definições iniciais

- Variáveis – Tipos primitivos - Inteiros

Tipo: int

Valor Mínimo: -2.147.483.648

Valor Máximo: 2.147.483.647

Bytes consumidos: 4

Tipo: long

Valor Mínimo: -922.337.203.685.475.808

Valor Máximo: 922.337.203.685.475.807

Bytes consumidos: 8

Linguagem de Programação JAVA

definições iniciais

- Variáveis – Tipos primitivos – Ponto flutuante

Tipo: float

Valor Mínimo: $-1.4e^{-45}$

Valor Máximo: $3.4e^{38}$

Bytes consumidos: 4

Tipo: double

Valor Mínimo: $-4.9e^{-324}$

Valor Máximo: $1.7e^{308}$

Bytes consumidos: 8

Linguagem de Programação JAVA
definições iniciais

- Variáveis – Tipos primitivos – Caracter

São utilizados para expressar uma tecla e ocupa 2 bytes na memória ram.

Os literais char são expressas incluindo o caractere desejado entre aspas simples.

Exemplo:

```
char meuChar = 'x';
```

Linguagem de Programação JAVA
definições iniciais

- Variáveis – Tipos primitivos – Caracter

O Java suporta também uma série de sequências de escape usando a barra invertida (\ ), que é chamada de caracter de escape.

A barra invertida indica que um caracter especial deve ser enviado para a saída quando o próximo caracter é combinado com ela, formando uma sequência de escape.

Linguagem de Programação JAVA
definições iniciais

- Variáveis – Tipos primitivos – Caracter

\n – nova linha
\f – nova página
\r – para retorno
\’ – para aspas simples
\t – para tabulação
\” – para aspas duplas
\b – para backspace
\\ - para barra invertida

Linguagem de Programação JAVA

definições iniciais

- Variáveis – Tipos primitivos – Booleanos

  Podem ser representadas apenas por dois valores: true e false.

- Variáveis – Reference

  Armazenam o endereço da memória estendida para um determinado objeto.

  Sintaxe:

  <tipo variável> <nome variável> = new <tipo variável>();

Linguagem de Programação JAVA
definições iniciais

- Variáveis – Reference

Exemplos:

```
String s = new String();
String s2 = “Teste”;
Object meu object = new Object();
```

Linguagem de Programação JAVA

definições iniciais

- Variáveis – a classe String

Geralmente a classe String é uma das primeiras que utilizamos pois representa um texto (um conjunto de caracteres) e sua inicialização pode ser feita semelhante a inicialização de variáveis de tipos primitivos.

- Variáveis locais

Variáveis declaradas dentro de métodos ou blocos de códigos são definidas como locais. Este tipo de variável não possui valor de inicialização padrão, portanto, devemos indicar um valor, caso contrário, receberemos um erro de compilação

Linguagem de Programação JAVA

Operadores em Java

- Exemplo

TesteVariaveisLocais.java

O programa acima tem um erro de compilação. A variável i não foi inicializada;

Linguagem de Programação JAVA

Operadores em Java

- Escopo

  O escopo define em qual parte do programa a variável estará acessível. Até agora utilizamos somente declarações de variáveis dentro de métodos (declaração do main).

- Exemplo:

  TesteEscopo.java

Linguagem de Programação JAVA

Operadores em Java

- Exercício:

  Crie a classe DeclaracaoVariaveis.java e dentro do método main declare, inicialize e utilize as variáveis como definido abaixo:

  - Declare uma variável do tipo String para armazenar o nome de uma pessoa;
  - Declare uma variável do tipo String para armazenar a data de nascimento de uma pessoa;
  - Declare uma variável do tipo String para armazenar o rg de uma pessoa;
  - Declare uma variável do tipo String para armazenar o sexo da pessoa utilizando a seguinte regra: MASCULINO (M) e FEMININO (F)
  - Declare uma variável do tipo double para armazenar o salário da pessoa;
  - Imprima todos os valores de maneira a obter a seguinte saída:

  O Senhor(a) <nome da pessoa> portador(a) do RG número <rg>, nascido em <data nascimento>, do sexo <sexo>, está registrado com o salário de R$ <salario>.

Linguagem de Programação JAVA

Operadores em Java

- Lembre-se:

  Você pode utilizar os caracteres de escape na impressão:

  \n : pular linha
  \t : tab
  \\ : \
  \” : “

Linguagem de Programação JAVA

Operadores em Java

- Exercício de Certificação:

  Qual é a saída do seguinte código quando compilado e executado? Selecione a(s) alternativa(s) correta(s).

```
1. class Questao1 {
2.     public static void main (String args[]) {
3.         int y = 0;
4.         int x = z = 1;
5.         System.out.println(y + “,” + x + “,” + z);
6.     }
7. }
```

Linguagem de Programação JAVA

Operadores em Java

- Alternativas:

a) Imprime 0,1,1

b) Erro durante a compilação na linha 1

c) Imprime 0,0,1

d) Erro durante a compilação na linha 4

e) Erro durante a compilação na linha 5

Linguagem de Programação JAVA

Operadores em Java

- Operadores Unários:

Negação	!
Pré e Pós incremento	+ +
Pré e Pós decremento	- -
Sinal Positivo	+
Sinal Negativo	-
Cast	( )

Linguagem de Programação JAVA

Operadores em Java

- Operadores de negação:

Operador de complemento, na qual é utilizado para inverter o valor de uma expressão booleana ou lógica. Então uma expressão !false resulta em true, enquanto a expressão !true resulta em false.

Exemplo:

TesteOperadorNot.java

Linguagem de Programação JAVA

Operadores em Java

- Operadores de incremento e decremento ++, --

Os operadores “+ +” e “- -” são chamados operadores de incremento e decremento respectivamente. Estes operadores modificam o valor de uma expressão adicionando ou subtraindo 1.

Os operadores de incremento e decremento podem ser:

- **Pós-fixados:** O incremento ou decremento é feito após o uso da variável nas outras expressões que envolvem a variável em questão.
- **Pré-Fixados:** O incremento ou decremento é realizado antes que a variável seja utilizada em qualquer outra operação.

Linguagem de Programação JAVA

Operadores em Java

- Operadores de representação de sinal: + e -

Os operadores unários + e – representam positivo e negativo respectivamente. O operador + não tem outra função a não ser deixar explícito que um número é positivo.

- Operadores de representação de sinal: + e – (unário). Exemplo:

```
int x = -3;
int z = +3;   // z = 3 seria exatamente igual
```

Linguagem de Programação JAVA

Operadores em Java

- Operadores de inversão: ~

Conhecido como Bitwise Inversion Operator (operador de inversão de bits) converte todos os bits 1 em 0 e todos os bits 0 em 1.

```
int x = -3;
int z = +3;   // z = 3 seria exatamente igual
```

Para obtenção do resultado de inversão bit a bit de um número inteiro basta somar um e inverter o sinal.

Linguagem de Programação JAVA

Operadores em Java

- Operadores de inversão: ~

  Exemplo:

  int resultado = ~7 ➔ resultado = 7 + 1 invertendo o sinal = -8

  int resultado = ~ -19 ➔ resultado = -19 + 1 invertendo o sinal = 18

Linguagem de Programação JAVA

Operadores em Java

- Operadores de conversão: cast

O operador de cast é usado para uma conversão explícita de uma expressão.

O cast pode ser aplicado para trocar tipos de valores primitivos e também pode ser aplicado em referência para objeto.

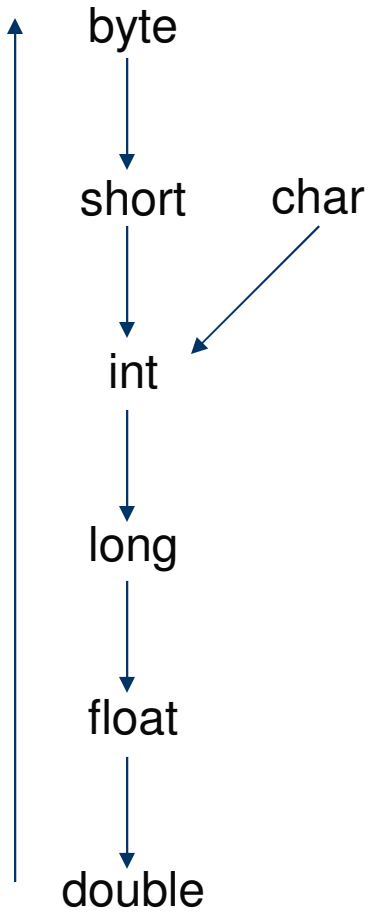
Pode ser: cast explícito e cast automático.

Linguagem de Programação JAVA

Operadores em Java

- Operadores de conversão: cast explícito

Pode ser que um double seja maior que a capacidade máxima de um “byte”. Neste caso estamos convertendo tipos de maior capacidade para tipos de menor capacidade. Temos que fazer o cast explicitamente. Caso contrário teremos um erro de compilação.

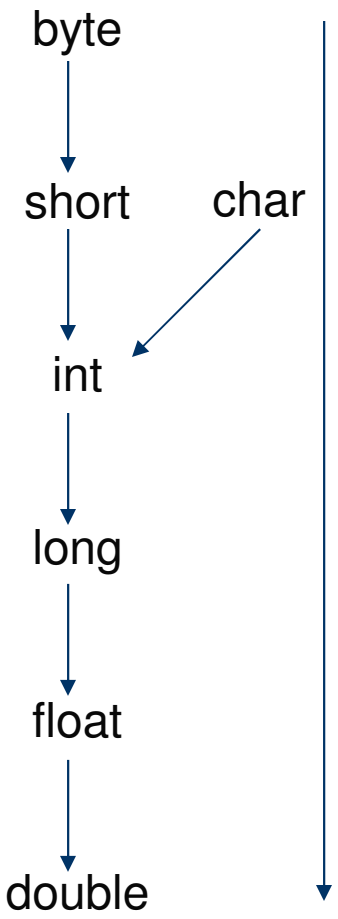


Linguagem de Programação JAVA

Operadores em Java

- Operadores de conversão: cast automático

Não precisamos fazer o cast explícito. Um byte sempre "cabe" em um short, assim como um long sempre "cabe" em um float.



Linguagem de Programação JAVA

Operadores em Java

- Operadores aritméticos: +, -, *, / e %
- Operadores de comparação: <, <=, >, >=, == e !=

Linguagem de Programação JAVA

Operadores em Java

- Operadores Lógicos AND e OR (& e |)

Aplicáveis somente entre operandos booleanos.

& (e "burro") -> avalia todos os operandos, mesmo que o resultado da expressão já seja falso.

&& (e "inteligente") -> avalia a expressão enquanto os seus operandos forem verdadeiros

| (ou "burro") -> avalia todos os operandos, mesmo que o resultado da expressão já seja verdadeiro

|| (ou "inteligente") -> Avalia a expressão enquanto seus operandos forem falsos.

Linguagem de Programação JAVA

Operadores em Java

- Operadores de atribuição: =, +=, -=, *=, /=, %=

Linguagem de Programação JAVA

Operadores em Java

- Operador ternário

Vamos analisar o seguinte exemplo:

```
int a = 2;
int b = 3;
int c = 4;

a = b>c ? b : c;
```

É a mesma coisa que se colocássemos assim:

```
se (b > c) entao
   a = b
senao
   a = c
fimse
```

Linguagem de Programação JAVA

Operadores em Java

- Operador ternário: Exemplo

TesteOperadorTernario.java

Linguagem de Programação JAVA

Operadores em Java

- Operador ternário: Certificação

Qual é a saída do seguinte código quando compilado e executado.

```
Class Questao49 {
    public static void main (String args [ ]) {
        boolean b = false;
        String s = (b = !b) ? (b = !b) ? “Oi” : “oi” : (b = !b) ? “world” : “World”;
        System.out.println(s);
    }
}
```

Linguagem de Programação JAVA

Operadores em Java

- Precedência de operadores

A precedência de um operador especifica quem tem mais prioridade de execução quando há combinações de dois ou mais operadores na mesma expressão, ou seja, independentemente de sua posição dentro da expressão, o operador de maior prioridade será executado primeiro que os demais (caso não haja parênteses). A ordem que uma expressão é avaliada determina o valor do resultado. Exemplo:

10 * 2 – 4 * 2 + 2 / 2 = ???

Para resolver esta expressão corretamente, de acordo com o cálculo que o computador faz, é importante conhecer a seguinte tabela:

Linguagem de Programação JAVA

Operadores em Java

Maior	parênteses mais internos
	++ e – (pós-fixado)
	~, !, ++ e – (pré-fixado)
	*, /, %
	+ e –
	<, >, <=, >=, instanceof
	==, !=
	&
	^
	|
	&&
	||
	? :
Menor	=, +=, -=, *=, /=, %=, ^=

Linguagem de Programação JAVA

Operadores em Java

Além disso, deve se ter em mente que os operadores com a mesma procedência são avaliados da esquerda para a direita, na medida que forem sendo encontrados. Uma boa prática de programação é priorizar a ordem de execução das operações através da inclusão de parêntesis, tornando o código mais legível.

No caso de existirem grupos de parênteses uns dentro dos outros, a prioridade de execução deve ser do grupo mais interno. Por exemplo:

```
(2 + 3 * (4 – (8 / 4 + (5 + 3) ) ) + 1 )
|          |     |         |--1--|  | |      |
|          |     |-------- 2 ------ | |      |
|          |--------- 3 -------------|       |
|----------------- 4 ----------------------|
```

Linguagem de Programação JAVA

Operadores em Java

Resumo:

Operador	Exemplo	Resultado
+ (unário)	+ 3	O valor int 3
+ (binário)	3 + 4	O valor 7
- (unário)	-3	O valor –3
- (binário)	3 – 4	O valor –1
*	2 * 7	14

Linguagem de Programação JAVA

Operadores em Java

Resumo:

Operador	Exemplo	Resultado
+ + (pré-fixado)	x = 3; y = ++ x;	x vale 4 e y vale 4
+ + (pós-fixado)	x = 3; y = x ++;	x vale 4 e y vale 3
- - (prefixado)	x = 3; y = - - x;	x vale 2 e y vale 2
- - (pós-fixado)	x = 3; y = x - -;	x vale 2 e y vale 3
/	11 / 5	2
% (resto)	11 % 5	1

Linguagem de Programação JAVA

Operadores em Java

Resumo:

Operador	Exemplo	Resultado
=	x = 5;	x vale 5
+ =	x = 1; x += 5;	x vale 6
- =	x = 1; x -= 5;	x vale -4
* =	x = 1; x *= 5;	x vale 5
/ =	int x = 1; x /= 5;	x vale 0
% =	x = 1; x %= 5;	x vale 1

Linguagem de Programação JAVA

Operadores em Java

Resumo:

Operador	Exemplo	Resultado
<	5 < 7	true
<=	5 <= 7	true
>	5 > 7	false
>=	5 >= 7	false
= =	5 = = 7	false
!=	5 != 7	True

Linguagem de Programação JAVA

Operadores em Java

Resumo:

Categoria	Operador	Exemplo	Resultado
STRING	+ (concatenação)	“ab” + “cd”	“abcd”
CAST	(tipo) (conversão)	double d = 4.2; byte b = (byte) d;	b vale 4
Ternário	? : (if)	x = 1; y = (x > 0) ? 5 : 10	y vale 5

Linguagem de Programação JAVA

Operadores em Java

- Exercício:

1. Coloque após cada linha o valor que será impresso ou diga se a linha será impressa. (Operadores.java)