

Programação II

Prof. Alcemir Rodrigues Santos

Introdução



Agenda

- Linguagem Java
- Princípios Básicos
- Variáveis e Operadores
- Tipos de Dados
- Escopo de Variáveis
- Conversão de primitivos
- Constantes
- Operadores
- Entrada e Saída



9/17/19

Programação II | Alcemir Santos

2



Linguagem Java

- A linguagem Java foi desenvolvida pela Sun Microsystems em 1995;
- Apesar de relativamente nova, a linguagem obteve uma espetacular aceitação por programadores do mundo inteiro, tendo se difundido como nunca antes ocorreu com uma linguagem de programação;
- Um fator que colaborou com isso, é o fato da linguagem possuir vantagens agregadas tais como:
 - Orientação a objetos
 - Independência de plataforma
 - Multitarefa
 - Robusta
 - Segura
 - Distribuída

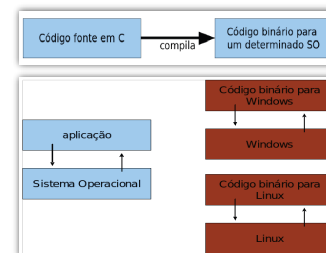
9/17/19

Programação II | Alcemir Santos

3



Principal Vantagem do JAVA



9/17/19

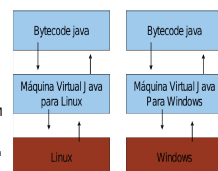
Programação II | Alcemir Santos

4



Principal Vantagem do JAVA

- Java é independente do SO, ou seja, não é preciso se preocupar em qual SO seu sistema está rodando;
- Sua aplicação é gerenciada pela Java Virtual Machine (JVM);
- A JVM é responsável por gerenciar memória, threads, pilha de execução, etc;
- Independência de plataforma através da JVM (Java Virtual Machine):
 - A JVM roda em uma plataforma específica e interpreta um programa Java para código de máquina específico da plataforma em questão.



9/17/19

Programação II | Alcemir Santos

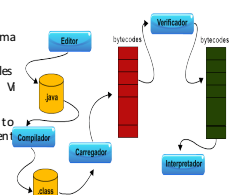
5



Funcionamento de um Programa em Java

Editor

- É a escrita ou desenvolvimento do programa (código-fonte).
- Se faz necessário um editor de texto simples como o Bloco de Notas (Windows) ou o VI (Linux).
- Existem ambientes de desenvolvimento integrado (IDE - Integrated Development Environment) como:
 - NetBeans;
 - Eclipse;
 - JCreator;
 - BlueJ.



9/17/19

Programação II | Alcemir Santos

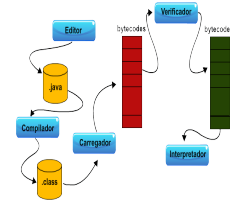
6



Funcionamento de um Programa em Java

Compilador

- O compilador Java converte o código-fonte Java em "bytecodes", que representam as tarefas a serem realizadas durante a fase de execução. Os bytecodes são executados pela Java Virtual Machine (JVM) – uma parte do JDK e a base da plataforma Java;
- A máquina virtual Java (JVM) é um aplicativo de software que simula um computador, mas oculta o sistema operacional e o hardware subjacentes dos programas que interagem com a JVM.



9/17/19

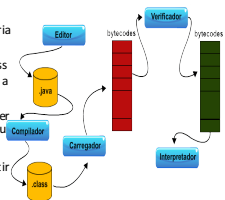
Programação II | Alcaimir Santos

7

Funcionamento de um Programa em Java

Carregador

- Todo programa deve ser colocado na memória antes de poder executar;
- O carregador de classe transfere os arquivos .class contendo os bytecodes do programa para a memória principal;
- O carregador de classe também carrega qualquer arquivo .class fornecido pelo Java que seu programa utiliza;
- Os arquivos .class podem ser carregados a partir de um disco em seu sistema ou em uma rede.



9/17/19

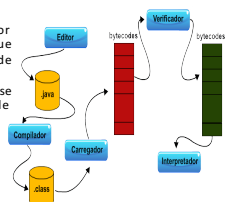
Programação II | Alcaimir Santos

8

Funcionamento de um Programa em Java

Verificador

- Enquanto as classes são carregadas, o verificador de bytecodes examina os códigos para assegurar que eles são válidos e não violam restrições de segurança do Java;
- O Java impõe uma forte segurança para certificar-se de que os programas Java que chegam pela rede não danifiquem os arquivos do sistema.



9/17/19

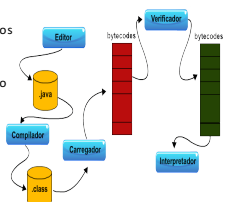
Programação II | Alcaimir Santos

9

Funcionamento de um Programa em Java

Interpretador

- A JVM executa o programa interpretando os bytecodes gerados na fase de compilação;
- Com isso, sequências de ações especificadas pelo programador são, enfim, executadas.



9/17/19

Programação II | Alcaimir Santos

10

Utilizações e Aplicações do Java

- Desenvolvimento de aplicativos corporativos de grande porte.
- Desenvolvimento de aplicativos Web.
- Fornecimento de aplicativos para dispositivos voltados para o consumo popular (celulares, PDAs, smartcards, etc.).
- Muitos outros propósitos.

9/17/19

Programação II | Alcaimir Santos

11

Plataformas JAVA

- Atualmente Java é uma linguagem que pode ser utilizada para desenvolvimento para desktop, web e desenvolvimento para micro dispositivos. Para tanto, existem três plataformas distintas:
 - J2SE (Standard Edition) – Indicada para desenvolvimento Desktop;
 - J2EE (Enterprise Edition) – Indicada para desenvolvimento Web;
 - J2ME (Micro Edition) – Indicada para desenvolvimento para micro dispositivos.
- Java eliminou
 - Variáveis e funções globais;
 - Ponteiros (Gerenciamento automático de memória);
 - Goto e struct;

9/17/19

Programação II | Alcaimir Santos

12

Java é Orientada a Objetos

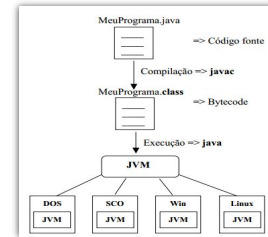
- Objetos e Classes;
- Encapsulamento de dados e operações;
- Herança;
- Polimorfismo;
- Criação e remoção dinâmica de objetos.

9/17/19

Programação II | Alcaimir Santos

13

Programa em Java



9/17/19

Esquema de compilação e execução de um programa em Java.

14

Programa em Java

- O desenvolvimento de aplicações Java sempre é feito através de classes.

```

<modificador de acesso> class <nome da classe>
{
    <Declaração das Variáveis de Instância (Atributos)>
    <Declaração de Métodos>
    public static void main( String args[] )
    {
        //corpo principal do programa
    }
}
  
```

- Os atributos são espaços em memória reservados para armazenar informações durante a execução da classe;
- Os métodos por sua vez definem as operações que podem ser realizadas pela classe.

9/17/19

Programação II | Alcaimir Santos

15

Programa em Java

```

// Modificador de Acesso
public class Exemplo1 // Nome da Classe
{
    // Variável de Instância
    String mensagem = "Meu primeiro programa em Java!";

    // Definição de Métodos
    public void Imprime_Msg()
    {
        //Mostra um texto na tela
        System.out.println(mensagem);
    }

    // Corpo do Programa
    public static void main( String [] args )
    {
        Imprime_Msg();
    }
}
  
```

9/17/19

Programação II | Alcaimir Santos

16

Princípios Básicos da Linguagem Java

- Java é case-sensitive
- As classes, métodos ou blocos de código sempre estarão delimitados por um abrir ({) e fechar (}) de chaves
- Um comando deve sempre ser finalizado por um ponto e vírgula (;)
- Dois tipos de comentário:
//Comentário de uma linha e
/* Comentário de Várias */
- Nomes de variáveis, classes e métodos devem sempre começar por letras, \$ ou _

9/17/19

Programação II | Alcaimir Santos

17

Princípios Básicos da Linguagem Java

❖ abstract,	❖ double	❖ int	❖ static
❖ Boolean	❖ else	❖ interface	❖ super
❖ Break	❖ extends	❖ long	❖ switch
❖ byte	❖ false,	❖ native	❖ synchronized
❖ case	❖ final	❖ new	❖ this
❖ catch	❖ finally	❖ null	❖ throw
❖ char	❖ float	❖ package	❖ transient
❖ class	❖ for	❖ private	❖ true
❖ const	❖ if	❖ protected	❖ try
❖ continue	❖ implements	❖ public	❖ void
❖ default	❖ import	❖ return	❖ volatile
❖ do	❖ instanceof	❖ short	❖ while

PALAVRAS CHAVES

9/17/19

Programação II | Alcaimir Santos

18

Princípios Básicos da Linguagem Java

- **Nome de Classes:** Utilize substantivos ou frases substantivas descritivas para nome de classes. Deixe maiúscula a primeira letra de cada substantivo que compõe o nome, por exemplo: MyFirstClassName;
- **Nome de Métodos:** Use verbos para nome de métodos. Faça a primeira letra do nome minúscula com cada letra inicial interna maiúscula. Por exemplo: `getUserName()`;
- **Nome de Variáveis:** Escolha nomes que indiquem o uso pretendido da variável. Utilize a primeira letra do nome minúscula e a inicial das outras palavras que formam o nome maiúscula. Exemplos: `customerName`, `customerCreditLimit`

9/17/19

Programação II | Alcaimir Santos

19

Variáveis e Operadores

- Uma variável é um nome simbólico para uma “porção de memória” na qual um valor pode ser armazenado. Toda variável tem as seguintes características:
 - Deve ser declarada antes de ser utilizada;
 - Possui um tipo, um identificador e um escopo;
 - Podem ser locais, quando declaradas dentro de métodos, ou atributos, quando declaradas no corpo da classe;
 - Podem ser inicializadas;

9/17/19

Programação II | Alcaimir Santos

20

Variáveis e Operadores

- Como Java é uma linguagem fortemente tipada, **todas as variáveis devem ser declaradas antes de serem usadas**;
- O tipo de uma variável determina o tipo de informação que pode ser armazenada nela;
- Variáveis em Java podem ser declaradas como atributos, no corpo da classe, ou podem ser declaradas localmente em qualquer parte da implementação de um método;
- Variáveis também possuem um escopo, o qual determina onde no programa ela estará visível e poderá ser acessada.

9/17/19

Programação II | Alcaimir Santos

21

Variáveis e Operadores

- A declaração de variáveis no Java segue a seguinte sintaxe:

tipo identificador [= valor] [, identificador [= valor]];



```
String msg = "Aula 1";
int umInteiro, UmInteiro;
float umFloat = 0.5;
char caracter = 'I';
boolean flag = false;
umInteiro = 90;
```

9/17/19

Programação II | Alcaimir Santos

22

Tipos de Dados

- Uma variável em Java pode ser de um dos quatro tipos de dados: classes, interfaces, arrays (referências) e tipos primitivos (tamanho fixo);

Categoria	Tipo	Tamanho
Inteiro	byte	8 bits
Inteiro	short	16 bits
Inteiro	int	32 bits
Inteiro	long	64 bits
Ponto Flutuante	float	32 bits
Ponto Flutuante	double	64 bits
Caracter	char	16 bits
Lógico	boolean	true / false

Tipo	Tamanho	Variação
byte	8 bits	-128 a +127
short	16 bits	-32.768 a +32.767
int	32 bits	-2.147.483.648 a +2.147.483.647
long	64 bits	-9.223.372.036.854.775.808 a +9.223.372.036.854.775.807

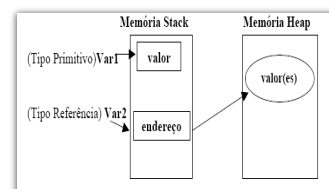
9/17/19

Programação II | Alcaimir Santos

23

Tipos de Dados

- Todo programa em Java possui dois tipos de memória:
 - principal (stack);
 - Auxiliar (heap).



9/17/19

Programação II | Alcaimir Santos

24

Escopo das Variáveis

- O escopo de uma variável é o bloco de código dentro do qual ela é acessível e determina quando a variável é criada e destruída:

```
void metodo ()
{
    int x;
    {
        int y;
        x = 10;
    }
    y = 20;
}
```

→ Início do bloco de x
→ Início do bloco de y
→ OK! A variável x está sendo acessada dentro de seu bloco!
→ Fim do bloco de y
→ ERRO! A variável y está sendo acessada fora de seu bloco!
→ Fim do bloco de x

Poderia ser um while, for ou if

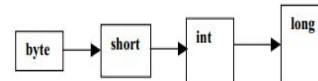
9/17/19

Programação II | Alcaimir Santos

25

Conversão entre tipos primitivos

- Java automaticamente converte um valor de um tipo numérico para outro tipo de maior capacidade de armazenamento. Por exemplo, se tentarmos atribuir um byte para um short, o compilador automaticamente converte o byte em short antes de realizar a atribuição.



9/17/19

Programação II | Alcaimir Santos

26

Conversão entre tipos primitivos

- O contrário não é verdadeiro. Para efetuarmos uma conversão de um tipo maior para outro de menor capacidade é necessário usar um mecanismo chamado type cast, caso contrário um erro será lançado.
- Exemplo:
 - byte é um tipo que permite armazenar valores inteiros no intervalo de 0 a 127
 - byte b1 = 127, b2 = 127, b3;
 - b3 = b1 + b2; Erro, pois b1 + b2 resulta em 254
 - b3 = (byte) (b1 + b2); Está correto, mas irá ocorrer perda de informação!

9/17/19

Programação II | Alcaimir Santos

27

Constantes

- Uma constante pode tanto ser definida como um atributo de classe como uma variável local.
- Uma constante uma vez que foi declarada e atribuído um valor para a mesma, não é permitido que outra atribuição seja efetuada, ocasionando um erro de compilação caso isso ocorra.

```
final <tipo> <identificador> [= valor];
final double MIN_VALOR = 100.0;
```

9/17/19

Programação II | Alcaimir Santos

28

Operadores

- Operador de Atribuição

Operador	Nome	Exemplo
=	Atribuição	int var1 = 5; var2 = 0; var2 = var1 + 10; var1 = var2 + 50;

- Operadores Aritméticos

Operador	Nome	Exemplo	Resultado
+	Adição	23 + 29	52
-	Subtração	20 - 23	6
*	Multiplicação	0.5 * salario	
/	Divisão	100 / 50	2
%	Módulo (resto da divisão)	20 % 3	2

9/17/19

Programação II | Alcaimir Santos

29

Operadores

- Operadores Relacionais

Operador	Nome	Exemplo	Resultado
==	Igual	x == 10	
!=	Diferente	3 != 2	true
<	Menor	10 < 10	false
>	Maior	10 > 6	true
>=	Maior ou igual	3 >= 3	true
<=	Menor ou igual	7 <= 6	false

- Operadores Lógicos

Operador	Nome	Exemplo	Resultado
&&	AND	(0 < 2) && (10 > 5)	true
	OR	(10 > 11) (10 < 12)	true
!	NOT	!(1 == 4)	true
^	XOR	(1 != 0) ^ (2 < 3)	false

9/17/19

Programação II | Alcaimir Santos

30

Operadores

• Atribuição Composta

Operador	Nome	Exemplo (x = 10;)	Resultado
+=	mais igual	x += 10;	20
-=	menos igual	x -= 10;	10
*=	vezes igual	x *= 10;	100
%=	módulo igual	x %= 10;	0
/=	dividido igual	x /= 10;	0

9/17/19

Programação II | Alcemir Santos

31

Operadores

• Incremento e Decremento

- Operador ++ incrementa de um.
 - Ex: int vari = 4; vari++; //vari = 5
 - Ex: int vari = 4; ++vari; //vari = 5
- Operador -- decrementa de um Ex: int vari = 4; vari--; //vari = 3
- Cuidado com a ordem de avaliação das expressões
 - Ex:

```
int num = 10;
if (num++ > 10) {
    system.out.println("maior que dez");
}
```

9/17/19

Programação II | Alcemir Santos

32

Operadores

• Incremento e Decremento

- Operador ++ incrementa de um.
 - Ex: int vari = 4; vari++; //vari = 5
 - Ex: int vari = 4; ++vari; //vari = 5
- Operador -- decrementa de um Ex: int vari = 4; vari--; //vari = 3
- Cuidado com a ordem de avaliação das expressões
 - Ex:

```
int num = 10;
if (++num > 10) {
    system.out.println("maior que dez");
}
```

9/17/19

Programação II | Alcemir Santos

33

Entrada e Saída

• Entrada de dados:

- Scanner sc = new Scanner(System.in);
- int a = sc.nextInt();

• Saída de dados:

- System.out.println("Programação Orientada a Objetos I");
- String str = "Programação Orientada a Objetos I";
- System.out.println(str);

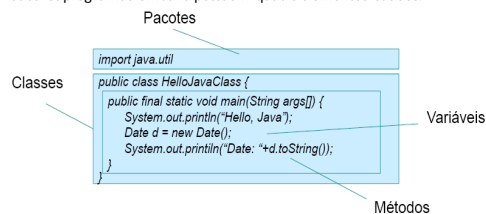
9/17/19

Programação II | Alcemir Santos

34

Elementos básicos

- Todos os programas em Java possuem quatro elementos básicos:



9/17/19

Programação II | Alcemir Santos

35

Exemplo 1

- Programa em Java para ler uma String e mostrar na tela a String lida.

```
1 package Aula1;
2
3
4 import java.util.Scanner;
5
6 public class Exemplo1 {
7     public static void main(String[] args) {
8         String nome;
9         Scanner teclado = new Scanner(System.in);
10        System.out.println("Digite seu nome: ");
11        nome = teclado.nextLine();
12        System.out.println("Seu nome é: " + nome);
13    }
14 }
15
```

```
Saída - POOI (run)
run:
Digite seu nome:
thiago
Seu nome é: thiago
CONSTRUÍDO COM SUCESSO
```

9/17/19

Programação II | Alcemir Santos

36

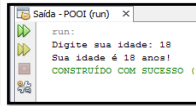
Exemplo 2

- Programa em Java para ler um inteiro e mostrar na tela o valor lido.

```

1 package Aula1;
2
3 import java.util.Scanner;
4
5 public class Exemplo2 {
6     public static void main(String[] args) {
7         int idade;
8         Scanner teclado = new Scanner(System.in);
9         System.out.println("Digite sua idade: ");
10        idade = teclado.nextInt();
11        System.out.println("Sua idade é " + idade + " anos!");
12    }
13 }
14

```



9/17/19

Programação II | Alcemir Santos

37

Exercício

- 1- Faça um algoritmo em Java que sejam digitadas as **três notas** de um aluno e, em seguida, calcule e exiba a **média** dessas notas.

Entrada de dados:

- Primeira nota: **n1**
- Segunda nota: **n2**
- Terceira nota: **n3**

$$\text{Média das notas} = \frac{(n1 + n2 + n3)}{3}$$

Saída de dados:

- Média das notas: **média**



9/17/19

Programação II | Alcemir Santos

38

Exercício

- 2- Faça um programa em Java que dados os coeficientes **a**, **b** e **c** de uma equação de segundo grau ($ax^2 + bx + c = 0$), exibir o **delta** da equação.

Entrada de dados:

- Coeficiente A: **a**
- Coeficiente B: **b**
- Coeficiente C: **c**

$$\Delta = b^2 - 4ac$$

Saída de dados:

- Delta: **d**

9/17/19

Programação II | Alcemir Santos

39

Exercício

- 3- Faça um algoritmo em Java que receba do usuário o **tempo** e a **distância** e, em seguida, calcule e exiba a **velocidade média**.



$$\text{Velocidade Média} = \frac{\text{Distância}}{\text{Tempo}}$$

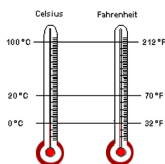
9/17/19

Programação II | Alcemir Santos

40

Exercício

- 4- Faça um algoritmo em Java para realizar a conversão de graus **Fahrenheit** para **Celsius**, onde, o usuário irá informar o valor da temperatura em Fahrenheit e o programa deverá exibir o valor convertido em Celsius.



$$\frac{^{\circ}C}{5} = \frac{^{\circ}F - 32}{9}$$

9/17/19

Programação II | Alcemir Santos

41