

Introdução a linguagem de programação Java

(Operadores, tipos e condicionais)

Prof. Rhafael Freitas da Costa, M. Sc.

Desenvolvimento de Software

Revisão



Como se tornar um “bom” desenvolvedor de software?



- ❑ Praticar muito: não há substituto para a prática quando se trata de desenvolvimento de software. Escreva muito código, faça projetos pessoais, participe de competições de programação e, se possível, trabalhe em projetos de código aberto.



Definir o problema:

- ❑ Verificar quais as formas possíveis para resolver o problema; Definir a melhor forma de solucionar o problema dentro do contexto atual.



Criar o algoritmo:

- ☐ Transformar o algoritmo em programa através de uma linguagem;
- ☐ Analisar se o problema foi resolvido, caso contrário, voltar à etapa inicial e repetir o processo.



Legibilidade do Código

- ☐ O algoritmo deve ser compreendido por qualquer pessoa, mesmo que ela não tenha participado da criação do mesmo;
- ☐ Para que isso seja possível você deve escrever o código seguindo alguns princípios básicos a fim de torná-lo de fácil entendimento;
- ☐ Usar comentários para informar o que cada trecho de código faz;
- ☐ Identar o código aninhando linhas pertencentes a uma mesma rotina ou função.



- ❑ Um programa de computador é dividido em Instruções e Informações:
- ❑ **Instruções:** ordens dadas para a máquina através do programa criado por nós;
- ❑ **Informações:** dados que serão manipulados pelas instruções.



Variáveis e Constantes

- ❑ Variáveis e constantes são elementos fundamentais em programação e matemática, usados para armazenar valores e representar dados;
- ❑ As variáveis podem ser modificadas ao longo do programa;
- ❑ As constantes permanecem com o mesmo valor durante todo o programa.



Variáveis e Constantes

- ☐ Nomenclatura **NÃO** pode conter:
 - ☐ Primeiro caractere numérico;
 - ☐ Espaço vazios;
 - ☐ Caracteres especiais;
 - ☐ Palavras reservadas da linguagem.
- ☐ Cuidar com casesensitive.



Tipos de dados primitivos

Grupo	Tipo Primitivo	Tamanho	Valores válidos
Inteiros	byte	1 byte	-128 até 127
	short	2 bytes	-32768 até 32767
	int	4 bytes	-2.147.483.648 até 2.147.483.647;
	long	8 bytes	-9.223.372.036.854.775.808 até 9.223.372.036.854.775.807
Flutuantes	float	4 bytes	1.40129846432481707E-45 até 3.40282346638528860E+38;
	double	8 bytes	4.94065645841246544E-324 até 1.79769313486231570E+308.
Booleanos	boolean	1 byte	True ou False
Caractere	char	2 bytes	1 caractere da tabela Unicode (Ex: 'a')



Operadores Aritméticos

Operador	Propósito	Exemplo	Resultado
=	“Setar” uma variável	$i = 10$	$i = 10$
+	Soma	$i = 10 + 5$	$i = 15$
-	Subtração	$i = 10 - 5$	$i = 5$
*	Multiplicação	$i = 10 * 5$	$i = 50$
/	Divisão	$i = 10 / 5$	$i = 2$
%	Resto da divisão	$i = 10 \% 5$	$i = 0$



Operadores Aritméticos e Atribuições

Operador	Propósito	Exemplo	Resultado
=	“Setar” uma variável	<code>i = 10</code>	<code>i = 10</code>
<code>+=</code>	Soma	<code>i += 5</code>	<code>i = 15</code>
<code>-=</code>	Subtração	<code>i -= 5</code>	<code>i = 10</code>
<code>*=</code>	Multiplicação	<code>i *= 5</code>	<code>i = 50</code>
<code>/=</code>	Divisão	<code>i /= 5</code>	<code>i = 10</code>
<code>++</code>	Adicionar 1 ao valor	<code>i++ (i = i + 1)</code>	<code>i = 11</code>
<code>--</code>	Subtrair 1 do valor	<code>i-- (i = i - 1)</code>	<code>i = 10</code>



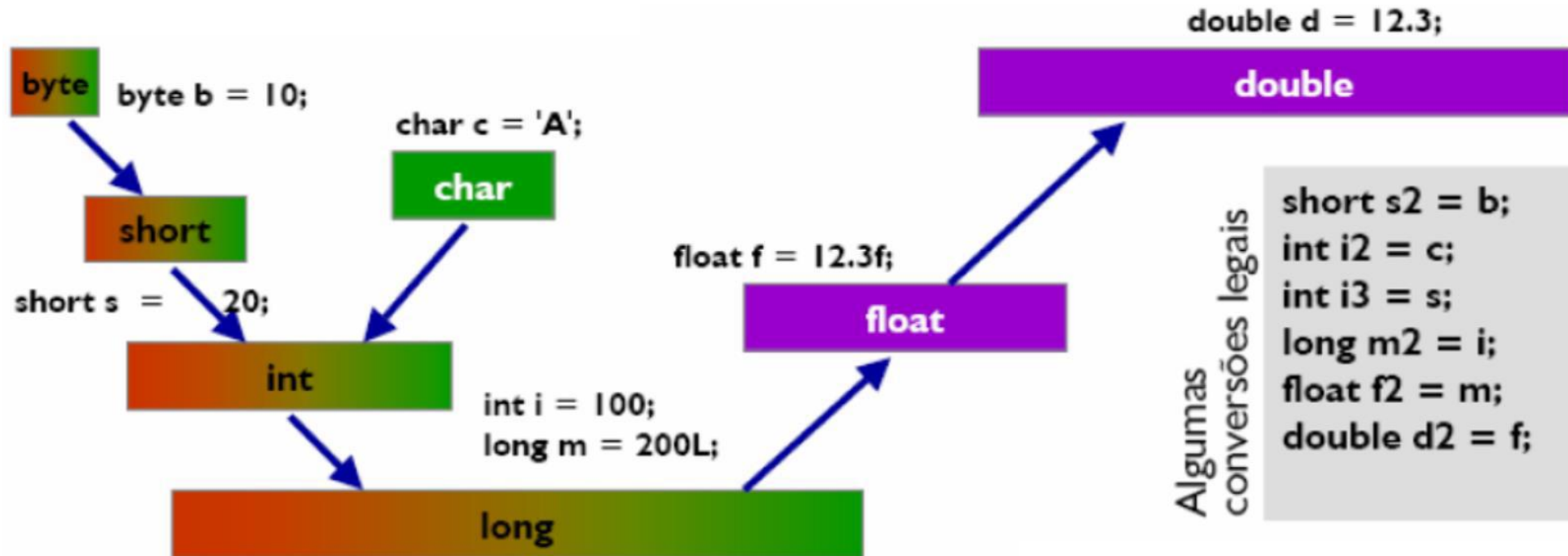
Operadores Comparativos e Lógicos

Operador	Significado
>	Maior que
>=	Maior ou igual a
<	Menor que
<=	Menor ou igual a
==	Igual a
!=	Diferente de

Operador	Significado
&&	AND
	OR
!	NOT



Conversão automática e casting



Wrappers e Parsers

Primitive Data Type	Wrapper Class
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
boolean	Boolean



Conversores, valueOf(), toString()

- ☐ Integer.parseInt(...);
- ☐ Integer.valueOf(...);
- ☐ Double.parseDouble(...);
- ☐ Double.valueOf(...);
- ☐ Float.parseFloat(...);
- ☐ Float.valueOf(...);
- ☐ Conversores, valueOf(), toString()



Strings (sequências de caracteres)

- ❑ **String** é uma “classe” amplamente utilizada na linguagem Java para manipular textos que são armazenados na memória do computador como uma sequência de caracteres.
- ❑ String frase = Só sei que nada sei (Sócrates).



Entrada e Saída



Entrada e Saída

```
1  import java.util.Scanner;
2
3  public class Programa {
4
5      Run | Debug
6      public static void main(String[] args) {
7
8          System.out.println(x:"Vamos iniciar nosso aprendizado em Java!");
9          System.out.print(s:"Digite um valor: ");
10
11          var scanner = new Scanner(System.in);
12          var valor = scanner.nextInt();
13
14          System.out.println("O valor informado foi: " + valor);
15          System.out.println(String.format(format:"O valor informado foi: %s", valor));
16          System.out.printf(format:"O valor informado foi: %s", valor);
17      }
```



Conceitos



Extensões dos arquivos

- ☐ Código fonte: .java
- ☐ Classe compilada: .class
- ☐ Pacote de arquivos para execução: .jar



Convenção

- ❑ Nomes de classes são iniciados por letras maiúsculas
- ❑ Nomes de métodos, atributos e variáveis são iniciados por letras minúsculas
- ❑ Em nomes compostos, cada palavra é iniciada por letra maiúscula, as palavras não são separadas por nenhum símbolo.



- ❑ Estruturas de controles são blocos de códigos que analisam os tipos de dados armazenados nas variáveis. As estruturas de controles são processos básicos de tomada de decisões das máquinas.



Estruturas de controle condicional



If / else

```
10     if (condicao) {  
11         // Bloco de comando  
12     } else {  
13         // Bloco de comando  
14     }  
15
```

```
12     if (condicao1) {  
13         // Bloco de comando  
14     } else if (condicao2) {  
15         // Bloco de comando  
16     } else {  
17         // Bloco de comando  
18     }
```



Switch Case

- ❑ Tipo de variáveis aceitas (INT e CHAR).

```
8      var opcao = "";  
9  
10     switch(opcao) {  
11         case "Brasil":  
12             // bloco_comandos  
13             break;  
14         case "New_York":  
15             // bloco_comandos  
16             break;  
17         default:  
18             // bloco_comandos  
19     }
```



- ☐ [Operadores](#)
- ☐ [DataTypes](#)
- ☐ [Conversão](#)
- ☐ [Strings](#)
- ☐ [If / else](#)
- ☐ [Switch case](#)



Exercício 1

- ☐ Criar um programa para processar as seguintes informações de um aluno: matrícula, nome , nota1 , nota2 e se ele está aprovado ou reprovado.
- ☐ Durante a execução calcular a nota final do aluno utilizando a seguinte regra: $\text{notaFinal} = (\text{nota1} + \text{nota2}) / 2$. Se a nota final for igual ou superior a 6 o aluno está aprovado, caso contrário estará reprovado;
- ☐ O programa deverá imprimir o seguinte relatório:
 - ☐ Matrícula: xxxxx
 - ☐ Nome: xxxxx xxxxx
 - ☐ Aprovado: (x) Sim () Não
 - ☐ Nota final: xxxxx



Exercício 1

- ❑ Escreva um programa que leia o código de um determinado produto e mostre a sua classificação. Utilize a tabela abaixo como referência:

Código	Descrição
1	Alimento não-perecível
2	Alimento perecível
3	Vestuário
4	Limpeza



OBRIGADO!

Desenvolvimento de Software

Professor Rhafael Freitas
da Costa, M. Sc.